UNIVERSITÉ DU QUÉBEC À MONTRÉAL

ÉTUDE DE LA CLASSIFICATION DES BACTÉRIOPHAGES

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR
VAN DUNG NGUYEN

DÉCEMBRE 2008

REMERCIEMENTS

Les premières personnes que je tiens à remercier sont, bien sûr, Vladimir Makarenkov et Pierre Poirier, mes directeurs de thèse. A Vladimir qui m'étonne chaque jour depuis notre rencontre par son style sobre mais terriblement efficace pour m'amener doucement à apprécier puis finalement aimer le domaine de l'analyse phylogénétique. A Pierre pour m'avoir fait découvrir que le philosophe, qu'il incarne si bien, est plus qu'un « pelleteux de nuages », c'est une personne dotée d'une rigueur de raisonnement et d'une finesse d'esprit qui permet d'appréhender des problèmes de grande complexité. Avec l'un et l'autre, j'ai appris à approfondir le sens de dépassement de soi face à l'adversité et aux difficultés durant mon parcours universitaire. À Pierre et à Vladimir, simplement un mot : merci.

Je tiens également à remercier mes compagnons dans les travaux académiques tout comme dans la pratique sportive du soccer, Abdoulaye Baniré Diallo et Alix Boc. Si Abdoulaye est devenu professeur à l'UQAM, Alix est aussi en bonne voie après l'achèvement de ses études doctorales.

Je désirerais aussi remercier les professeurs qui ont lu cette thèse. Je leur suis reconnaissant du temps qu'ils ont pris pour lire cette étude et pour les commentaires qu'ils ont apportés.

Sur un plan plus personnel, je ne peux oublier de remercier mon épouse Kim, mes enfants Vincent et Alexandre, ma mère, les autres membres de ma famille ainsi que tous mes amis qui m'ont soutenu tout le long de mon parcours. Une pensée spéciale pour mon père qui nous a quittés depuis longtemps pour un monde meilleur.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

Les bactériophages (i.e., virus de bactéries) constituent l'un des groupes d'organismes les plus abondants dans la biosphère. Ils jouissent d'une très grande biodiversité. Nos connaissances partielles de ces microorganismes sont sans cesse remises en cause par de nouvelles découvertes et le recensement est loin d'être terminé. Il existe bien des classifications basées sur les critères de morphologie et d'homologie génétique, mais celles-ci ne tiennent pas compte de l'évolution caractéristique des virus qui comprend à la fois la transmission verticale (évolution classique) et horizontale (évolution réticulée) de l'information. De plus, ces classifications ne disent rien à propos des ancêtres communs des espèces. Il y a donc beaucoup de possibilités d'affiner la taxonomie virale existante.

Dans cette étude, nous présentons une nouvelle approche de classification des bactériophages, basée sur des méthodes heuristiques tirées des sciences cognitives de la catégorisation. Cette approche originale vise à reconstruire l'histoire évolutive des organismes viraux, en tenant compte de l'hypothèse d'évolution classique ainsi que l'hypothèse d'évolution réticulée, i.e., les transferts horizontaux de gènes (THG).

En d'autres termes, la classification proposée prend en considération d'une part, l'approche traditionnelle d'analyse phylogénétique qui inclut la reconstruction d'arbres d'espèces par les méthodes de distances et d'inférence bayésienne et la reconstruction de séquences de protéines ancestrales par la méthode Tree-HMM en tenant compte des substitutions, des insertions et des délétions de caractères génétiques [Diallo et al. 2006 ; Felsenstein 1981], et d'autre part, l'approche de détection des transferts horizontaux par la méthode de réconciliation topologique de l'arbre d'espèces et l'arbre de gène [Makarenkov et al. 2008].

Mots-clés : *classification, catégorisation, phylogénie, taxonomie, virus, bactériophages, transferts horizontaux de gènes, reconstruction ancestrale.*

INTRODUCTION

L'étude des virus est un domaine complexe. De récents travaux ont mis en évidence, par exemple, des ressemblances au niveau de la morphologie de certains virus bien que ces derniers ne partagent aucun gène *homologue**,[1], et inversement, des virus avec une grande proportion de gènes partagés sont souvent classés dans des familles morphologiquement distinctes [Lawrence et al. 2002]. De même, d'autres résultats de recherches ont montré que des organismes viraux phylogénétiquement éloignés au niveau des espèces et infectant des hôtes de niches écologiques différentes présentent de très fortes similarités structurales au niveau des protéines [Spinelli et al. 2005, 2006 ; Ricagno et al. 2006].

Ainsi, on peut difficilement imaginer deux organismes plus différents qu'une bactérie du lait et un être humain. Et pourtant, la protéine *Receptor Binding Protein* (RBP), qui permet à certains virus d'infecter les cellules de l'un ou l'autre de ces organismes hôtes, présente d'étonnantes similitudes structurales [Spinelli et al. 2005]. Cette protéine agissant comme une clé pour ouvrir la voie à l'infection virale, a une structure intéressante en soi, mais c'est plutôt la serrure qui est visée. Cette « serrure » est la structure de la membrane cellulaire de l'hôte qui est reconnue par la protéine RBP de chaque virus impliqué. L'objectif est de réussir à inactiver la serrure pour ainsi empêcher les virus d'infecter les cellules [Spinelli et al. 2005]. Il est évident qu'il est très important dans ce cas de comprendre les relations évolutives entre les hôtes cellulaires et leurs parasites viraux.

Une des étapes préalables, et non des moindres, consiste aussi à mieux comprendre l'histoire de l'évolution des différents types de virus existants en se basant notamment sur une taxonomie établie de ces microorganismes.

Si les premiers travaux autour des virus datent de la fin du XIX^{e} siècle, la classification des organismes viraux, elle, est toujours en cours de construction. La difficulté intrinsèque, due aux modes d'évolution à la fois classique (i.e., par héritage vertical d'un

[1] Le signe (*) invite le lecteur à consulter le glossaire en fin du document pour plus de détails. Ce signe n'apparaît qu'à la première occurrence du mot dans la suite du document.

ancêtre commun) et réticulée (caractérisée notamment par les transferts horizontaux de gènes entre espèces distinctes), et à la complexité de l'écosystème des sujets, est telle qu'une classification exhaustive et de consensus reste à proposer. De plus, la découverte permanente de nouvelles espèces vivant dans une très grande biodiversité, vient remettre en cause nos connaissances acquises [Pace 1997 ; Forterre et al. 2002 ; Galus 2006] ainsi que nos méthodes traditionnelles d'analyse phylogénétique qui ne sont pas nécessairement adaptées pour étudier de telles espèces [Liu et al. 2006].

Il y a là des possibilités d'affiner la taxonomie des virus. C'est précisément ce que nous allons essayer de faire à travers la présente étude.

Nous proposons en effet, une approche de classification des bactériophages (un ensemble représentatif des virus) qui vise à classer, avec des méthodes plus appropriées, les espèces en tenant compte à la fois de leur mode d'évolution classique et de leur mode d'évolution réticulée. Cette classification se résume en trois phases : l'inférence phylogénétique [Swofford et al. 1996], la détection des transferts horizontaux de gènes [Makarenkov et al. 2008] et la reconstruction de séquences ancestrales [Blanchette et al. 2007 ; Diallo et al. 2006].

De surprenants résultats ont été obtenus à chacune des phases citées : [1] Un arbre d'espèces a été généré, et avec lui nombreux sont les signaux phylogénétiques qui ont été retrouvés. Plus de la moitié des 22 groupes identifiés par la présente étude sont reconnus par l'organisme de classification de référence NCBI/ICTV. Sur les 10 groupes restant, 7 regroupent chacun plusieurs espèces différentes, cela explique-t-il la difficulté qu'ont les organismes de classification traditionnels comme NCBI/ICTV de les considérér dans leur classification (section 5.1) ? [2] Près de 1500 transferts horizontaux de gènes ont été détectés, les deux tiers concernent les transferts entre les groupes (inter) et à l'intérieur de chacun des groupes (intra) identifiés précédemment (section 5.2), [3] Des séquences ancestrales ont été également générées pour chacun des nœuds internes de l'arbre phylogénétique qui représentent les ancêtres communs les plus proches (ACP). L'accent a été mis sur un cas d'exemple d'ancêtres communs particulièrement intéressant à étudier à savoir les ancêtres possibles des virus *Lactoccocal Lactis* (section 5.3).

Les diverses approches méthodologiques menant à ces résultats seront bien entendu discutées tout le long de ce document. Toutes ces approches recourent à la classification. Nous avons donc naturellement regardé du côté des sciences cognitives à la recherche des modèles[2] de catégorisation utilisés en psychologie cognitive. Ce travail de réflexion accompli, nous avons été davantage en mesure de parourir le domaine des méthodes[2] de classification employées en analyse phylogénétique. Cette démarche, qui visait d'abord à satisfaire les exigences définies par le doctorat en informatique cognitive, et par là même d'amener une dimension cognitive au sujet traité quant à la démarche méthodologique adoptée, s'est avérée fructueuse. Pour preuves, si besoin en est, les trois points suivants, et qui vont être repris en détail dans les chapitres à venir : la bioinformatique comme domaine inter-disciplinaire, les modèles de catégorisation humaine transposables en méthodes de classification machine et un cas concret d'apport mutuel entre les sciences cognitives et les sciences de la vie dont l'analyse phylogénétique.

Sur le plan de la légitimité d'abord, la bioinformatique, mariage entre la biologie et l'informatique, comprend plusieurs champs de recherche dont la phylogénie qui nous intéresse ici. La bioinformatique fait partie intégrante, selon l'organisme IEEE[3], d'un domaine chaperon inter-disciplinaire sous le nom de « *Cognitive Informatics* » lequel regroupe entre autres : *Neural Networks, Machine Learning, Knowledge Representation, Problem Solving, Cognitive Modeling, Artificial Intelligence*, etc.

Pour démontrer ensuite, la transposition entre les disciplines, nous mettrons en perspective, dans les chapitres II et III, d'un côté, les modèles de catégorisation humaine utilisés en psychologie cognitive tels que les modèles basés sur la similarité (section 2.3.1) et l'inférence bayésienne (section 2.3.2), et de l'autre, en autant de méthodes différentes de classification machine employées en analyse phylogénétique comme les méthodes basées sur les distances (section 3.3) et les méthodes de types probabilistes (section 3.4).

---

[2] Dans ce document, le terme *modèle* désigne (sauf quelques exceptions) essentiellement le modèle cognitif de catégorisation, tandis que le terme *méthode* désigne les techniques de classification machine, spécifiquement celles utilisées en bioinfomatique en général et en traitement phylogénétique en particulier.

[3] On peut par exemple, consulter : *http://www2.enel.ucalgary.ca/ICCI2006/ICCI2006_files/ICCI07-CFP-V3.pdf.*

Finalement, à travers un cas concret d'expériences mené durant nos travaux, nous essayerons de montrer l'apport mutuel entre la phylogénie et les sciences cognitives.

Dans un sens, les travaux en psychologie cognitive nous apprennent qu'il existe, lors d'un processus de catégorisation, plusieurs modèles cognitifs alternatifs dont celui par distances et celui par inférence bayésienne (chapitre II). Fort de ce point, nous avons pu opter, face aux différentes méthodes d'analyse phylogénétique (maximum de vraisemblance, parcimonie et estimation bayésienne) utilisées pour confronter les résultats de classification obtenus, pour un choix éclairé en retenant la méthode d'estimation bayésienne comme une alternative valide de la méthode de distances (section 3.6.3.1). Il s'est avéré aussi par la suite, en comparaison avec la méthode de distances que c'est la méthode qui donne les meilleurs résultats (section 4.2.2.6).

Dans le sens inverse : grâce à ses performances, la méthode bayésienne gagne en popularité chez les phylogénéticiens comme en témoignent les récents travaux [Huelsenbeck & Ronquist 2001 ; Larget & Simon 1999 ; Blanquart & Lartillot 2006], sans oublier la présente étude. Avec l'intérêt croissant en phylogénie pour cette méthode de classification, nous informons de retour les sciences cognitives des potentiels de l'approche bayésienne susceptibles de renforcer notre compréhension des processus de catégorisation des objets. En effet, si jusqu'à récemment, les modèles basés sur la similarité comme le modèle des prototypes [Rosch 1973, 1975 ; Rosch & Mervis 1975] et le modèle des exemplaires ont été historiquement [Medin & Schaffer 1978 ; Nosofsky 1986] utilisés et donc largement diffusés dans les expériences cognitives, il faudrait sans doute accorder dorénavant plus de considérations encore aux modèles cognitifs bayésiens [Anderson 1991 ; Rehder 2003].

Avant de clore cette mise en contexte, voici une brève description du plan général de dissertation à venir qui comprend les chapitres suivants :

- Chapitre I – Problématique et objectif de recherche : la biodiversité des virus sera présentée comme la problématique de recherche. Face à un des enjeux de cette problématique, la taxonomie, notre objectif de recherche consiste à avancer une proposition originale de classification des espèces.

- Chapitre II – Catégorisation humaine : avant d'attaquer la thématique de la classification relevant de l'aspect machine, nous l'appréhenderons d'abord sous la perspective psychologique à savoir les processus cognitifs de catégorisation. Suivra ensuite, la description de la modélistion de ces processus. Il sera question en fin de partie, de la façon dont ces modèles seront transposés en méthodes de classification machine.

- Chapitre III – Classification machine : la classification machine sera décrite sous l'angle de l'apprentissage et de l'inférence, puis, plus contextuel, du point de vue des méthodes de distances et des méthodes d'inférence bayésienne. La discussion portera par la suite, sur la façon dont ces méthodes vont être appliquées dans le cadre d'une analyse phylogénétique.

- Chapitre IV – Approche originale de classification des bactériophages : nous aborderons par une mise en contexte à propos des modes d'évolution des bactériophages. Ensuite, il sera question des méthodes les plus pertinentes pour la reconstruction d'arbre d'espèces, pour la détection des transferts horizontaux de gènes et aussi, pour la reconstruction ancestrale.

- Chapitre V – Résultats : les résultats expérimentaux seront révélés à ce stade, notamment au niveau des transferts horizontaux de gènes et la reconstruction de séquences de protéines ancestrales, avec une emphase particulière sur le cas des virus *L. Lactis* (virus des bactéries du lait).

- Conclusion : les principales réalisations seront reprises en guise de conclusion, en particulier, les résultats ayant permis de souligner l'originalité de l'approche de classification des bactériophages. Nous terminerons sur plusieurs perspectives exploratoires stimulantes dont une suggestion, faite aux biologistes, d'analyser de manière approfondie les séquences protéiques ancestrales générées dans cette étude.

CHAPITRE I

—

PROBLÉMATIQUE ET OBJECTIF DE RECHERCHE

Plan du chapitre

Les virus en général et les bactériophages (virus de bactéries) en particulier constituent l'un des groupes d'organismes les plus abondants en nombre d'individus dans la biosphère [Bergh et al. 1989 ; Wommack et Colwell 2000]. Ils jouissent d'une très grande biodiversité. Leur recensement est toujours en cours et les classifications proposées sont nombreuses et diverses. Cependant, la difficulté intrinsèque due aux modes d'évolution caractéristiques des virus par transmission verticale et horizontale, ainsi qu'à la complexité de leur écosystème, est telle qu'une classification exhaustive de consensus reste à établir. Il y a donc là des possibilités d'affiner la taxonomie virale existante.

La première partie de ce chapitre posera la problématique entourant l'analyse de l'histoire évolutive des virus dans un contexte de biodiversité complexe. L'objectif de recherche sera présenté dans la partie suivante, et consistera en une nouvelle approche de classification des bactériophages combinant les méthodes de détection des transferts horizontaux de gènes et de reconstruction de séquences ancestrales.

1.1 PROBLÉMATIQUE

Nous décrivons dans ce chapitre, le contexte complexe dans lequel évoluent les virus, l'approche de classification communément utilisée et le degré de nos connaissances à propos des organismes viraux.

1.1.1 CONTEXTE DE BIOLOGIE VIRALE

Le contexte de biologie virale est fort complexe. La diversité des virus est telle que certains émettent l'hypothèse selon laquelle les virus seraient d'origine *polyphylétique** [Iyer et al. 2001], alors que d'autres les considèrent comme des éléments « anciens » [Gorbalenya et al. 1990 ; Koonin & Ilyina 1992], probablement antérieurs à la divergence des trois domaines du vivant. Chose certaine, il est très difficile d'étudier leur histoire évolutive.

1.1.1.1 Découverte des virus

Les premières études faisant état des virus dataient de la fin du XIX^{e} siècle. Les virus de bactéries ou bactériophages (appelés aussi phages) ont été découverts simultanément par Twort en 1915, sur des colonies de *Micrococcus*, et par d'Herelle en 1917 sur des cultures de *Shigella*. Les virus d'Archéobactéries (nouvellement appelés *Archaea**) ont été découverts beaucoup plus récemment puisque les premiers bactériophages d'Archéobactérie parasitant

Halobacterium salinarium n'ont été isolés qu'au milieu des années 1970 [Torvisk & Dundas 1978]. Mais il faudra attendre les travaux précurseurs de Wolfram Zillig et ses collaborateurs, au début des années 1980, pour que l'isolement de virus d'Archéobactéries halophiles puis hyperthermophiles prenne une véritable ampleur.

Depuis ces travaux précurseurs, ce sont près de 3000 espèces virales [van Regenmortel et al. 2000] qui ont été isolées et reconnues comme valides par le comité international de taxonomie des virus ICTV[4] [Büchen-Osmond 2003]. Ces organismes ont été isolés dans presque tous les environnements. Sur le plan quantitatif, il s'agit d'un composant majeur de la biosphère puisqu'on estime, par exemple, que le nombre de phages infectant des bactéries dans des environnements aquatiques approche les 10 millions de particules par ml, soit au moins 10 fois plus que les organismes cellulaires [Bergh et al. 1989 ; Wommack et Colwell 2000].

1.1.1.2 Diversité des virus

Un virus est traditionnellement défini comme un parasite de la cellule dont il utilise les constituants pour se multiplier. Il est capable de provoquer une maladie. Un virus est composé d'acides nucléiques (*ADN** ou *ARN**) entourés d'une enveloppe appelée la capside.

Derrière cette définition simple se cache une extraordinaire diversité [Rohwer & Edwards 2002] (section 4.1.6.1). D'abord sur la nature du génome qui peut être circulaire ou linéaire, et être composé d'ADN simple ou double brin, ou d'ARN simple ou double brin [sites NCBI et ICTV, les références de liens URL sont en Annexe B] (section 4.1.6.2, tableau 4.1). La taille des génomes est extrêmement variable [Collins et al. 1998 ; La Scola et al. 2003 ; Liu et al. 2006]. De plus, la très grande variabilité de composition génétique des virus résulte de riches stratégies réplicatives des gènes [Eisen 2000 ; Ochman et al. 2000]. La plupart des génomes viraux codent, à différents degrés, des éléments de leur propre machinerie de réplication et de transcription, incluant en particulier des *ADN* ou *ARN polymérases**, et détournent par contre divers composants de la machinerie de leur hôte pour compléter leur cycle.

---

[4] En plus de ICTV, il existe d'autres classifications des virus, se référer entre autres à Jarvis et al. [1991], et Maniloff et Ackermann [1998].

Le séquençage d'un nombre croissant de génomes viraux a permis d'établir certains aspects clés de l'évolution des virus qui montrent que la transmission verticale de l'information (vision traditionnelle de l'évolution) est loin d'être le seul mécanisme évolutif adopté par les virus. Les mécanismes non verticaux d'évolution existent aussi et ils sont principalement de trois ordres : la recombinaison homologue [Nilsson & Haggard-ljungquist 2001], la recombinaison non homologue [Lawrence et al. 2002] et les transferts horizontaux de gènes en provenance de leurs hôtes [Doolittle 1999]. Ces mécanismes sont aussi connus sous le nom d'évolution réticulée (versus l'évolution classique ou l'héritage vertical à partir d'un ancêtre commun). Lire par ailleurs, la section 4.1.2, "*Évolution réticulée et transferts horizontaux de gènes*".

1.1.1.3 Origine des virus

Etant donné la très grande diversité des génomes des virus, il semble très peu probable que tous ces éléments aient une origine unique [Iyer et al. 2001]. Au contraire, un consensus se dégage autour d'une origine multiple des virus (i.e., *polyphylétique*). Mais parallèlement, des analyses comparées de génomes de virus ont parfois suggéré l'existence d'une origine commune entre des virus très divergents [Gorbalenya et al. 1990 ; Koonin & Ilyina 1992].

Chose certaine, des connections évolutives ont souvent été mises en évidence entre les virus et les différents éléments génétiques mobiles des génomes (*plasmides**, *transposons**, *rétrotransposons**, etc.) [Xiong & Eickbush 1990 ; McClure 1991]. Il existe par ailleurs, des éléments *chimériques** tels que par exemple, le bactériophage *N15* qui possède un génome composé à 50% de gènes de type bactériophage et à 50% de gènes dérivant de plasmides linéaires [Ravin et al. 2000]. Bref, dans l'état de connaissances actuelles, il est très ardu de reconstruire l'hisoire évolutive des virus.

Toutefois, certaines hypothèses se dégagent quant à l'origine des virus. L'hypothèse d'un *monde des cellules* qui accrédite l'idée selon laquelle les virus sont issus d'un fragment d'ADN d'origine cellulaire qui s'est échappé, devenu autonome et infectieux [Ravin et al. 2000] ou bien encore, le produit de l'extrême réduction d'un génome de cellules ou de proto-cellules [Bandea 1983]. Cette hypothèse échoue, néanmoins, à expliquer l'origine évolutive d'un grand nombre de gènes n'ayant aucun homologue cellulaire (qu'ils codent ou non pour des protéines de fonction connue).

L'hypothèse alternative d'un *monde des virus* distinct (d'un point de vue originel) avance l'idée d'un ancêtre commun très ancien [Gorbalenya et al. 1990 ; Koonin & Ilyina 1992]. Une convergence de vue se dessine d'ailleurs et vient confirmer aujourd'hui cette idée. Des études génomiques, et surtout structurales, apportent de solides arguments sur une origine ancienne des virus, probablement antérieure à la divergence des trois domaines du vivant : *Archaea* (ou Archées), *Bacteria** (ou Bactéries) et *Eucarya** (ou Eucaryotes).

Ainsi, Spinelli et al. [2005, 2006] et Ricagno et al. [2006], par exemple, ont montré qu'entre des virus d'espèces différentes, il y a de très fortes ressemblances au niveau de la structure tridimensionnelle de la protéine RBP bien qu'aucune similarité au niveau de la séquence primaire ne soit détectable (lire par ailleurs, la section 5.3.2.1., "*Protéine RBP des phages L. Lactis*"). D'autres similarités structurales ont aussi été mises en évidence entre des virus *P3* des *Adénovirus* (virus d'eucaryotes) et des *Hexon* du phage *PRD1* (virus de bactéries) [Benson et al. 1999]. Par contre, leurs gènes ne possèdent pas d'homologue cellulaire connu à ce jour, ce qui rend peu probable leur acquisition indépendante par transferts horizontaux en provenance de leur hôte. Il est donc fort possible que ces structures similaires soient héritées d'un ancêtre commun antérieurement à la divergence des bactéries et des eucaryotes.

En fin de compte, ces différents résultats indiquent que des virus très divergents, infectant des hôtes phylogénétiquement très éloignés, partagent des caractères homologues. Même si l'on ne peut pas exclure des phénomènes de convergence évolutive (i.e., *homoplasie**), il est très probable qu'au moins une partie de ces caractères soit bien héritée d'un ancêtre commun, probablement antérieur à la divergence des trois domaines.

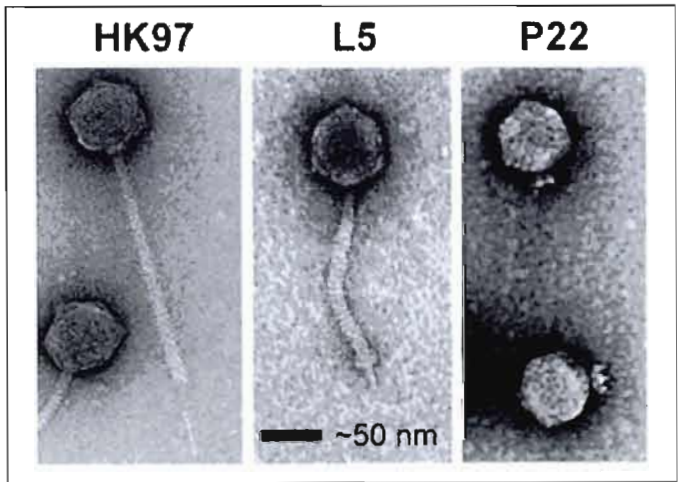
1.1.2 APPROCHE DE CLASSIFICATION EXISTANTE À REVOIR

L'approche de classification existante n'est pas en mesure de rendre compte de la réalité complexe du monde viral à cause de ses critères de discrimination insuffisants et ses méthodes d'analyse phylogénétique inadaptées.

1.1.2.1 Classification existante remise en cause

Historiquement, la grande diversité virale a rendu difficile leur classification. Jusqu'à une date récente, les principaux critères de classification retenus par l'ICTV étaient la morphologie générale, la nature de l'acide nucléique et les hôtes infectés. Pourtant, le séquençage d'un nombre croissant de génomes viraux a très largement remis en cause cette classification.

Ainsi, la morphologie du *virion** est utilisée pour grouper les virus *dsDNA* « à queue » (ou bactériophages) en trois groupes principaux (section 4.1.6.2 et tableau 4.1, et section 5.1.2 et figure 5.2) : *Siphoviridae*, *Myoviridae* et *Podoviridae*.



Hôte bactérien des phages		**Siphoviridae** Escherichia coli	**Siphoviridae** Mycobacterium smegmatis	**Podoviridae** Salmonella enterica
Gènes communs	**HK97**	–	0%	20%
	L5	0%	–	0%
	P22	22%	0%	–

Figure 1.1 : Ressemblance morphologique versus ressemblance génétique
(Tirée de Lawrence et al. 2002)

Pourtant, comme on peut le voir sur la figure 1.1, ces familles regroupent des éléments qui ne partagent que très peu de gènes homologues, sinon aucun, entre eux (les bactéries

HK97 et *L5* pourtant classées dans la famille *Siphoviridae*). Alors que des éléments qui partagent pourtant une fraction substantielle de gènes homologues sont classés dans deux familles différentes (*Podoviridae* et *Siphoviridae* pour les *P22* et *HK97*) [Lawrence et al. 2002]. De plus, les gènes codant pour les protéines associées à des morphologies ressemblantes sont souvent non homologues (e.g., les phages *T4* et *Mu* sont classés au sein des Myoviridae). Cela ne serait-il pas le signe d'une évolution convergente, indiquant qu'il existe des contraintes fonctionnelles favorisant une certaine morphologie de virion ?

1.1.2.2 Méthodes d'analyse phylogénétique inadaptées

La comparaison, par exemple, des génomes des phages *Lambdoïdes* a montré que dans les groupes de gènes codant pour les protéines de structures, on observe des gènes peu conservés, présents d'une manière erratique dans des génomes [Ravin et al. 2000]. Ces gènes sont très souvent de fonction inconnue et d'origine évolutive mystérieuse. Comme discuté précédemment (section 1.1.1.3), les virus sont considérés à la fois comme d'origine polyphylétique probable, mais également comme des éléments « anciens » certains.

Ceci pose le problème, en termes de méthodes d'analyse phylogénétique, de la reconstruction de l'histoire évolutive des virus avec les outils traditionnellement utilisés pour étudier l'évolution des êtres cellulaires. Face à cette biodiversité virale, les outils sont-ils bien adaptés ? Quelle est l'importance du transfert vertical de l'information en comparaison avec le transfert horizontal ? La reconstruction ancestrale peut-elle apporter des éléments de réponse dans la recherche de fonctions protéiques non identifiées ?

1.1.3 CONNAISSANCES PARCELLAIRES

D'un point de vue génomique, les analyses actuelles sont fortement dépendantes de l'échantillonnage utilisé et l'afflux de données issues du séquençage de génomes complets modifie régulièrement notre vision des choses. Le problème est d'autant plus crucial que notre connaissance de la biodiversité microbienne reste très parcellaire, au point que certains auteurs estiment que l'on sait cultiver moins de 1% de la biodiversité microbienne [Pace 1997]. Il ne fait donc aucun doute qu'une meilleure connaissance (au niveau génomique en particulier) de cette extraordinaire diversité pourrait apporter des éclairages nouveaux sur les questions abordées ici.

1.2 OBJECTIF DE RECHERCHE

Notre objectif de recherche consiste à relever le défi posé par la problématique discutée à la section précédente, en apportant des éléments de réponse par le biais d'une proposition originale de classification des bactériophages dsDNA (un ensemble représentatif de virus, voir la section 4.1.3). Cette classification prend en compte à la fois l'évolution classique et un aspect caractéristique de l'évolution réticulée, les transferts horizontaux de gènes.

En d'autres termes, cette classification prend en considération d'une part, l'approche traditionnelle de reconstruction d'arbre phylogénétique d'espèces (comparaison suivant les critères de similarité de gènes et de caractères homologues) (section 4.1.1.1) ainsi que la reconstruction de séquences ancestrales (inférence, à partir des séquences observées, des séquences ancestrales) [Diallo et al. 2006] (section 4.1.5), et d'autre part, l'approche novatrice de détection des transferts horizontaux de gènes (réconciliation topologique entre l'arbre d'espèces et l'arbre de gène[5]) [Makarenkov et al. 2008] (section 4.1.2).

Un exemple de comparaison structurale de protéine sera donné pour étayer nos travaux (section 5.3.2). Cela ne signifie pas pour autant que nous ayons la prétention d'attaquer la problématique d'analyse structurale des molécules, qui relève davantage des études biologiques fondamentales.

Bien que la biodiversité virale soit de nature fort complexe et les connaissances la concernant soient sans cesse renouvelées par des nouvelles découvertes, nous estimons de manière raisonnable que la présente étude basée sur les connaissances actuelles pourrait contribuer sinon à compléter les différentes classifications existantes perfectibles (e.g., celle de l'ICTV), du moins à mieux comprendre l'histoire évolutive des phages.

[5] Le mot *Gène* est au singulier car il s'agit de l'arbre phylogénétique d'un ensemble d'un même gène présent chez différentes espèces.

CHAPITRE II

–

PROCESSUS DE CATÉGORISATION

Plan du chapitre

La catégorisation se révèle être une activité cognitive qui consiste à regrouper des objets ou des événements non identiques dans des catégories, une catégorie cognitive étant un ensemble d'objets *considérés comme équivalents* par l'individu [Mervis & Rosch 1981]. Dès lors, les catégories cognitives facilitent non seulement le stockage et l'organisation des connaissances en mémoire, elles jouent aussi un rôle essentiel dans le processus d'évaluation des objets. En effet, l'organisation catégorielle des connaissances permet de réduire la complexité de l'information à laquelle l'individu est confronté, et ainsi d'améliorer l'efficience du traitement de l'information [Cohen & Basu 1987]. Les catégories sont donc utilisées dans différentes fonctions cognitives importantes dont la classification, la prédiction, l'inférence, la communication, la perception visuelle ou encore le raisonnement complexe [e.g., Markman & Ross 2003 ; Yamauchi & Markman 1998 ; Harnad 1997 ; Holyoak & Thagard 1995 ; Smith & Medin 1981].

L'étude de la catégorisation permet non seulement d'établir quels sont les critères et les types de traitements utilisés par le système cognitif pour catégoriser mais également permet de rendre compte de la manière dont les connaissances sont structurées dans la mémoire. Et à partir de là, la modélisation de la catégorisation s'en trouve facilitée.

Dans ce chapitre, nous présenterons les différents aspects cognitifs ayant traits à la catégorisation pour y mener finalement à sa modélisation. Pour commencer, nous présenterons les principales approches théoriques contemporaines de psychologie cognitive sur la catégorisation, en particulier les approches basées sur la notion de similarité entre les objets ainsi qu'une des approches basées sur l'inférence causale. La discussion portera également sur les organisations catégorielles et l'apprentissage de catégories qui sous-tendent les processus de catégorisation. Nous aborderons ensuite, les différentes modélisations dont le modèle des prototypes, le modèle des exemplaires, le modèle rationnel et le modèle causal. Nous montrerons finalement la façon dont nous comptons transposer les modèles de catégorisation humaine en méthodes de classification machine, lesquelles ont été appliquées dans cette étude d'analyse phylogénétique.

2.1 APPROCHES THÉORIQUES DE LA CATÉGORISATION

Dans cette première partie, nous commencerons par la critique de l'approche théorique classique de catégorisation. La discussion se poursuivra avec la présentation des approches théoriques contemporaines basées sur la similarité dont la théorie des prototypes et la théorie des exemplaires, et les approches basées sur l'inférence causale, dont spécifiquement la théorie des models causaux.

2.1.1 CRITIQUE DE L'APPROCHE CLASSIQUE

Avant les années 1970, le cadre d'analyse des catégories et des processus de catégorisation s'inscrit dans une conception *classique* ou aristotélicienne, s'appuyant notamment sur les travaux pionniers de Bruner et al. [1956]. Une catégorie est une entité discrète, définie par une liste de propriétés *nécessaires et suffisantes* au sein de laquelle tous les objets sont considérés comme étant équivalents quant à leur appartenance à une catégorie. Dans la catégorie Oiseau, par exemple, Moineau, Aigle, Autruche et Pingouin ont le même statut. Les propriétés sont indépendantes l'une de l'autre et considérées comme d'importance équivalente. La détermination de l'appartenance est *analytique* (c.-à-d., comparaison des d'attributs) et est basée sur la distinction vrai/faux. Un nouvel objet est affecté à la catégorie dont il possède tous les attributs. Tous les membres d'une catégorie possèdent nécessairement un même ensemble d'attributs.

Le problème avec l'approche classique est que cela suppose une frontière nette séparant les catégories. Avec une liste de propriétés nécessaires et suffisantes, il est toujours possible de dire que quelque chose appartient ou non à une catégorie. Or, les objets d'une catégorie ne semblent pas toujours partager les mêmes propriétés. Par exemple, la capacité de Voler est un trait caractéristique de beaucoup d'Oiseaux (e.g., le Moineau) mais pas de tous (e.g., l'Autruche). Wittgenstein [1953] propose l'idée de ressemblance de famille ou *air de famille* comme alternative à la notion traditionnelle de catégorie. Suivant cette idée, bien que tous les membres d'une même famille aient tendance à se ressembler, il n'est habituellement pas possible de définir un trait commun à tous les membres de la famille. On peut cependant, envisager un membre typique qui représente la famille. Par le biais de la catégorie des *jeux*, Wittgenstein [1953] préfigure comme l'un des précurseurs des approches contemporaines de catégorisation.

2.1.2 APPROCHES BASÉES SUR LA SIMILARITÉ

Comme alternative à la vision classique, plusieurs approches contemporaines de psychologie cognitive sur la catégorisation ont été proposées, dont les plus influentes sont la théorie des prototypes [Posner & Keele 1968, Rosch 1973, 1975 ; Minda & Smith 2001], la théorie des exemplaires [Medin & Schaffer 1978 ; Nosofsky 1986], la théorie *Decision Bound Theory* – DBT [Ashby & Townsend 1986 ; Ashby & Maddox 1993], la théorie basée sur les règles et exceptions [Nosofsky et al. 1994 ; Palmeri & Nosofsky 1995], et la théorie des modèles causaux [Rehder 2003 ; Rehder & Hastie 2004 ; Waldman et al. 1995].

Bien que la théorie basée sur les règles et la théorie DBT offrent chacune une alternative[6] différente et intéressante sur la catégorisation, nous nous sommes intéressés en particulier, dans le cadre de nos travaux, aux seules théories des prototypes et des exemplaires, ainsi qu'à la théorie des modèles causaux. Les deux premières citées sont des variantes des approches basées sur les mesures de similarité calculées entre le nouvel objet et les catégories. La dernière nommée est basée sur l'approche d'inférence causale qui nécessitent des hypothèses sur les probabilités *a priori* des catégories et la génération de cas à partir des catégories.

2.1.2.1 Théorie des prototypes

L'approche basée sur les prototypes a été proposée par Rosch [1973, 1975]. Cette approche s'inscrit dans le cadre d'analyse de catégories *naturelles* (e.g., Chien, Chat, Oiseau, etc.). Tous les membres d'une catégorie naturelle n'ont pas le même statut et les membres les plus représentatifs, appelés *prototypes* ou membres le plus central d'une catégorie (tel un point de référence cognitif), jouent un rôle privilégié dans la structure de catégorie (section 2.2.1.1).

Rosch [1973, 1975], et Rosch et Mervis [1975] prennent en compte l'organisation des catégories et leur fonctionnalité. La thèse défendue est que les catégories sont structurées par

---

[6] D'après la théorie basée sur les règles et exceptions, l'individu tente de trouver des règles qui permettent de placer tous (ou la plupart) des exemples dans la bonne catégorie. S'il y a des exceptions à la règle, alors ces exceptions peuvent être sauvegardées séparément [Nosofsky et al. 1994].

Avec l'approche théorique DBT, l'individu représente les catégories comme des régions d'espaces de propriétés, et les décisions de catégorisation sont prises en déterminant la région dans l'espace de propriétés dans laquelle certains exemples ont plus de probabilité d'y être (étant donné un modèle de bruit de perception) [Ashby & Maddox 1993]. Par ailleurs, Ashby et Maddox [1993] ont investi en détail les liens formels entre les modèles de catégorisation basés sur les exemplaires et les modèles DBT dans la théorie de la reconnaissance générale. Globalement, ils ont constaté qu'il y a de fortes équivalences entre ces deux types de modèles.

des effets prototypiques qui déterminent des espaces catégoriels hétérogènes, caractérisés par des cas centraux typiques et des limites floues : beaucoup de catégories naturelles sont structurées intérieurement en un prototype de la catégorie avec des membres non prototypes qui tendent vers un ordre allant des meilleurs aux plus faibles exemples. Ainsi, la catégorie se définit par rapport à un prototype, soit le meilleur représentant de la catégorie. Les autres membres de la catégorie se repèrent sur un gradient de typicalité, selon leur similitude avec le prototype. Par exemple, dans la catégorie Oiseau, Moineau est plus typique que Pingouin ou Autruche. La typicalité peut être définie comme l'une des dimensions décrivant l'espace catégoriel [Cordier & Dubois 1981]. Pour décider si un objet est membre d'une catégorie, on le compare (non plus avec l'ensemble des propriétés partagées par tous les objets mais) avec le prototype de catégorie [Posner & Keele 1968]. L'appartenance à une catégorie est maintenant définie en termes de distance (degré de similarité) et non plus en termes de propriétés. Notons aussi, que sur la base du degré de similarité, les cas marginaux peuvent être pris en compte (e.g., Chaise à trois pieds...).

Le prototype condense en fonction du principe d'*économie cognitive*, l'ensemble des propriétés de la plupart des objets. Cette condensation de la représentation de catégories sous forme de prototypes réduit les coûts de traitement, s'effectue de manière globale ou *holistique* (c.-à-d., processus global de similarité et de typicalité), et permet des inférences sur des valeurs par défaut. Aussi, les prototypes correspondent aux exemplaires les plus fréquemment cités, les plus rapidement identifiés, les plus disponibles pour effectuer des tâches comme par exemple, la résolution de problèmes [Dubois 1983] ou la classification de nouveaux objets. En d'autres termes, l'organisation de nos catégories suit le principe d'exploitation de la *structure du monde perçu* [Pacherie 2004]. Certaines combinaisons d'attributs ont tendance à être rencontrées plus fréquemment dans le monde que d'autres [Pacherie 2004]. Ainsi, les créatures qui ont des ailes ont aussi tendance à avoir des plumes plutôt que de la fourrure.

Il reste que, pour certains critiques, la théorie des prototypes s'est avérée insuffisante. Comme l'ont notés entre autres Medin et Schaffer [1978], et Murphy et Medin [1985], les principaux défauts de la théorie des prototypes sont sans doute liés à l'insuffisance de l'idée que les catégories sont représentées par des listes de traits et que la catégorisation est fonction

du degré de similitude avec le prototype catégoriel : qu'est-ce qui détermine la pertinence d'un trait et comment définir la similitude ? Pour les lecteurs intéressés, se référer par exemple, à la discussion développée dans [Pacherie 2004]. Malgré les insuffisances soulignées ici, la théorie des prototypes reste pertinente, comme le défendent notamment Minda et Smith [2001, 2002] dans leurs récents travaux.

2.1.2.2 Théorie des exemplaires

A partir de l'approche théorique précédente, Medin et Schaffer [1978] ont proposé la théorie des exemplaires. Ces derniers soutiennent que les catégories sont représentées par des exemples spécifiques et concrets. Une représentation mentale de catégories code donc les exemples *in extenso* qui composent la catégorie, telle une collection d'exemples [Medin & Schaffer 1978 ; Nosofsky 1986, Nosofsky & Zaki 1998]. Autrement dit, tous les exemples sont mémorisés, c'est-à-dire, stockés en mémoire. Les informations sur la catégorie n'ont pas un statut différent de celui des exemples, car les catégories sont représentées par les objets qui sont déjà en mémoire. Notre représentation de l'Oiseau est ainsi composée d'un ou quelques exemples d'Oiseaux que nous avons rencontrés auparavant. Ainsi, les représentations typiques de l'Oiseau sont celles qui se sont produites le plus fréquemment. Les tenants de l'approche avancent l'idée que la catégorisation est faite principalement en comparant de nouveaux exemples à chaque exemple déjà rencontrés [Nosofsky 1986]. En d'autres termes, la détermination de l'appartenance d'un nouvel objet s'effectue en comparant avec le plus proche voisin plutôt qu'un prototype. Si tous les exemples d'une catégorie sont codés, tous les exemples n'ont pas besoin de partager les mêmes propriétés. Plus les stimuli sont les *meilleurs* exemples d'une catégorie, plus ils sont étroitement liés aux exemples de la catégorie [Dopkins & Gleason 1997]. L'effet prototypique se manifeste par le biais de la comparaison entre le stimulus et l'exemple le plus similaire (i.e., la similarité de chaque exemple est calculée), faisant office de *prototype*. Les membres (ou les exemples) les plus typiques tendent en moyenne à être les plus actifs [Dopkins & Gleason 1997].

2.1.2.3 Prototypes versus exemplaires

Si l'approche des exemplaires peut préserver les informations (puisque tous les exemples sont mémorisés) sur les corrélations entre les différentes propriétés d'une catégorie [Medin et al. 1982], l'évidence suggère qu'après une longue expérience de l'individu avec l'objet, celui-ci utilise plutôt l'approche des prototypes pour le catégoriser [Homa et al. 1981]. Cependant, l'individu continue à être influencé par les cas qu'il a vu récemment : le Dermatologue, par exemple, peut faire un diagnostique d'autant plus juste qu'il a vu récemment des cas semblables.

Que les catégories soient décrites suivant la théorie des prototypes ou la théorie des exemplaires, l'une des propriétés récurrentes essentielles [e.g., Rosch & Mervis 1975 ; Deschamps 1977] consiste d'une part, en une accentuation des différences entre les objets appartenant à des catégories différentes (soit la minimisation des ressemblances inter-catégorielles), et d'autre part, en une accentuation des ressemblances entre objets appartenant à une même catégorie (soit la maximisation des ressemblances intra-catégorielles) : l'individu fait comme si les ressemblances ou les différences étaient plus marquées qu'elles ne le sont dans la réalité.

2.1.3 UNE APPROCHE D'INFÉRENCE CAUSALE – THÉORIE DES MODÈLES CAUSAUX

Les approches d'inférence causale proposent d'aborder les catégories sous un autre angle que celui de la similarité vu précédemment. En effet, de récentes études suggèrent qu'au moins certaines catégories sont définies ou décrites par une structure causale sous-jacente [Ahn et al. 2002 ; Hadjichristidis et al. 2004 ; Rehder, 2003 ; Rehder & Burnett 2005 ; Rehder & Hastie, 2004]. Il existe plusieurs approches théoriques pour expliquer la catégorisation par inférence causale, dont les deux principales sont l'approche associationniste [Gluck & Bower 1988 ; McClelland & Rogers 2003] et l'approche computationnelle [Anderson 1990, 1991 ; Ashby & Alfonso-Reese 1995 ; Rosseel 2002 ; Rehder 2003 ; Rehder & Burnett 2005].

Des deux approches, nous retiendrons la seconde pour discussion. Cette dernière est elle-même subdivisée en deux courants : l'inférence par typicalité [Anderson 1990, 1991] et l'inférence par structure causale [Rehder 2003 ; Rehder & Burnett 2005]. Avec la recherche de typicalité des exemples, le premier courant est ancré en fait quelque part entre la théorie des prototypes et la théorie des exemplaires [Markman & Roos 2003]. Nous reviendrons plus loin (section 2.2.2.2) sur ce point. Dans cette section, nous discuterons en particulier de l'approche d'inférence causale définie sous le terme de théorie des modèles causaux.

Selon la théorie des modèles causaux, la connaissance de l'individu des différentes catégories n'inclut pas seulement une représentation des propriétés de la catégorie mais également une représentation explicite des mécanismes causaux que l'individu croit exister entre ces propriétés [Rehder 1999, Waldman et al. 1995]. De plus, il utilise l'approche causale pour déterminer l'appartenance d'une nouvelle catégorie d'objets. Supposons par exemple, que l'individu croit que l'ADN d'Oiseau cause l'apparition des Ailes, laquelle cause à son tour la possibilité de Voler qui cause finalement la construction de Nids dans les arbres. Il devrait donc croire aussi (toute chose étant égale par ailleurs) que les propriétés résultantes les plus directement causées par l'ADN d'Oiseau (e.g., avoir des Ailes) ont plus de chances d'être générées que celles qui sont les plus indirectement causées (e.g., construction de Nids dans l'arbre). Par conséquent, les propriétés causées directement devraient être vues comme des occurrences les plus fréquentes parmi les membres de catégories (et donc devraient peser plus fortement dans la détermination de l'appartenance de catégorie).

Par ailleurs, d'après la théorie des modèles causaux, les combinaisons de propriétés sont les preuves importantes pour la détermination de l'appartenance de catégorie dans la mesure où elles sont conjointement consistantes ou inconsistantes avec la connaissance causale de la catégorie. Un animal, par exemple, qui ne Vole pas mais qui construit des Nids dans les arbres pourrait être considéré moins plausible comme un Oiseau (comment le Nid peut-il se retrouver dans l'arbre ?) qu'un animal qui ne Vole pas et qui construit des Nids au niveau du sol (e.g., l'Autruche) même si le premier a plus de propriétés qui sont typiques des Oiseaux.

Un facteur important qui contrôle les prédictions de patrons (*patterns*) de corrélation inter propriétés, est l'asymétrie des connaissances causales : le fait, par exemple, d'avoir des Ailes n'a pas pour cause d'avoir l'ADN d'Oiseau (e.g., la Chauve-souris qui a des Ailes mais qui n'est génétiquement pas un Oiseau) (section 2.2.1.2). C'est une caractéristique qui distingue des autres modèles d'inférence causale (e.g., modèle rationnel d'Anderson [1991], voir la section 2.3.2.1) qui représente les relations de manière symétrique.

2.2 PROCESSUS DE CATÉGORISATION

D'abord, nous montrerons dans cette seconde partie, deux différents types de structures organisationnelles de catégories, ensuite, la façon dont s'effectuent les processus d'apprentissage de catégories.

2.2.1 ORGANISATIONS CATÉGORIELLES

Au moins deux formes d'organisations de catégories sont possibles selon le point de vue théorique auquel on se réfère. La théorie des prototypes par exemple, a introduit la notion de catégories hiérarchisées, laquelle hiérarchie est organisée par une structure interne. Tandis que la théorie des modèles causaux décrit, elle, les catégories sous forme de structure causale.

2.2.1.1 Structures de catégories naturelles

Il existe différents types de catégories, des plus précises aux plus abstraites : la catégorisation est ainsi un concept multidimensionnel, les différents niveaux de catégorie étant hiérarchisés par une structure interne [Dubois, 1991]. L'organisation hiérarchique des catégories fait référence au niveau d'inclusion d'une catégorie, au degré de généralité de celle-ci : Meuble est une catégorie plus inclusive que Chaise, Lampe ou Table (figure 2.1).

Rosch [1973, 1975] met en évidence le fait que les catégories sont généralement organisées selon trois niveaux d'inclusion : le niveau surordonné, le niveau de base et le niveau subordonné (figure 2.1). Elle démontre également que le niveau de base est le niveau cognitivement le plus important, notamment : c'est le niveau le plus élevé auquel les membres d'une catégorie ont une forme générale semblable ; c'est aussi à ce niveau qu'une unique image (ou représentation) mentale peut résumer la catégorie ; c'est le niveau auquel des actions motrices semblables sont utilisées par les sujets dans leurs interactions avec les

membres de la catégorie ; ou bien encore, c'est à ce niveau que les gens identifient le plus rapidement l'appartenance catégorielle.

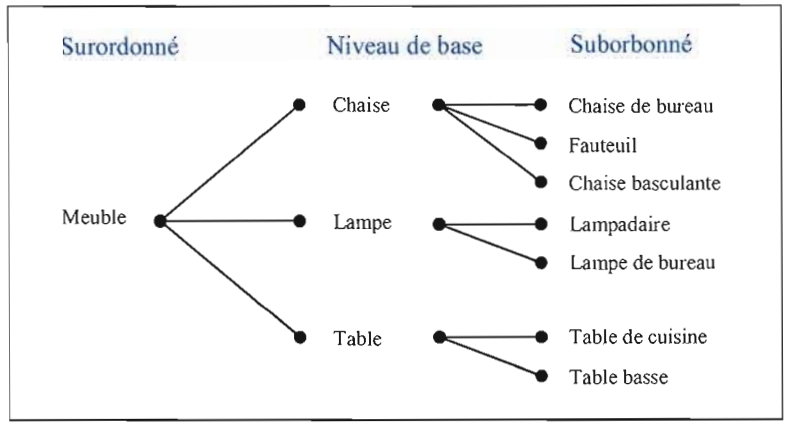


Figure 2.1 : Structure hiérarchique des catégories naturelles

Le niveau de base est ainsi fondamental sous quatre aspects : [1] La perception : c'est la forme perçue générale, la représentation mentale unique et l'identification rapide ; [2] La fonction : c'est le programme moteur général ; [3] La communication : ce sont les mots les plus courts et les plus couramment utilisés, ce sont aussi les premiers appris ; [4] L'organisation des connaissances : la plupart des attributs des membres d'une catégorie sont stockés à ce niveau. Le fait que les connaissances soient organisées principalement à ce niveau est établi de la manière suivante : quand on demande aux sujets de citer les attributs d'une catégorie, ils citent très peu d'attributs de catégories surordonnées, la plupart citent les attributs des catégories de niveau de base, et il n'y a presque aucun les attributs des catégories de niveau subordonné. Ainsi, lorsqu'on leur demande « *sur quoi êtes-vous assis ?* », la plupart des sujets répondent en utilisant le terme Chaise de préférence à Chaise de Cuisine ou bien encore à un terme surordonné tel que Meuble.

Néanmoins, il y a des différences selon les individus suivant leur degré de connaissance d'un domaine. Alors que le niveau de base de catégorisation des instruments de musique pour le non musicien est en gros : Piano, Guitare, Flute, Saxophone, les musiciens sont quant à eux, capables de faire des distinctions beaucoup plus précises d'instruments

individuels. Autrement dit, il semble que le niveau de base s'abaisse (i.e., devient plus précis dans la terminologie) avec la connaissance que l'on a d'un domaine [Pacherie 2004].

2.2.1.2 Structures de catégories causales

L'organisation causale des catégories suggère que certains types de catégories soient définis par une structure causale : les objets sont membres d'une même catégorie si leurs propriétés observées sont générées par la même structure causale sous-jacente [Rehder, 2003; Rehder & Hastie, 2004].

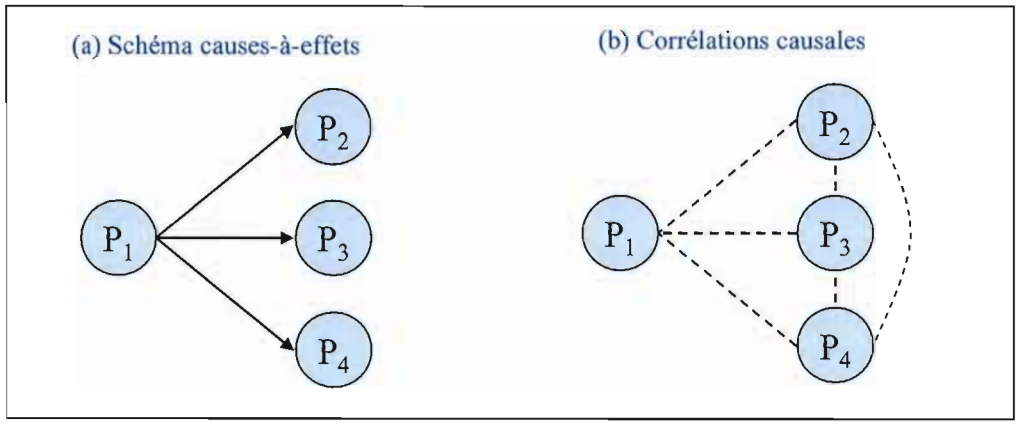


Figure 2.2 : Structure causale des catégories
(Adaptée de Rehder & Hastie 2001)

Comme illustration [Rehder & Hastie 2001], considérons la structure causale présentée à la figure 2.2a : dans le schéma causes-à-effets, la structure est déterminée par l'organisation de la propriété de catégorie P_1 à l'origine des trois autres propriétés résultantes P_2, P_3 et P_4. Le sens des flèches signifie que la cause précède leurs effets, indiquant une relation asymétrique, tel par exemple, le début d'un Feu comme l'élément précédent la production de Fumée et l'activation de l'Alarme. Par ailleurs, la structure causes-à-effets implique que les trois attributs d'effets sont corrélés parce qu'ils sont issus de la même cause. Ceci est illustré à la figure 2.2b avec des lignes en pointillé entre les propriétés P_1, P_2, P_3 et P_4 : trois Symptômes, par exemple, causés par une Maladie sont susceptibles d'être corrélés à travers une population de patients dans laquelle la maladie est présente chez une sous population.

Formellement, les structures causales sont représentées par le biais de réseaux bayésiens [Waldman et al. 1995, Rehder 2003, Rehder & Hastie 2004]. Les réseaux bayésiens se composent de nœuds qui représentent les variables et les arcs orientés pouvant être interprétés comme la représentation des relations causales directes entre les variables. Pour une description complète de réseaux bayésiens, se référer à Glymour [1998] et Pearl [2000].

2.2.2 APPRENTISSAGE DE CATÉGORIES

Un objectif majeur de la psychologie cognitive est de comprendre comment les catégories sont apprises et utilisées. La recherche sur la catégorisation a exploré à la fois la structure des catégories des gens et la faculté de ces derniers à acquérir de nouvelles catégories.

Il semblerait que la façon dont s'effectue l'apprentissage de catégories peut influencer la façon dont les catégories sont représentées [Yamauchi & Markman 1998 ; Markman & Ross 2003, Anderson et al. 2002 ; Chin-Parker & Ross 2004]. En effet, différents types d'apprentissage introduisent différents buts aux participants apprenants. Ces buts, à leur tour, peuvent affecter la représentation des catégories (soit la structure mentale). L'apprentissage par classification, par exemple, est une tâche supervisée[7] (l'apprenant peut saisir l'étiquette de catégorie) avec un but clair : apprendre à déterminer l'appartenance de catégorie des objets. La représentation des catégories résultante reflète ce but en incluant l'information la plus *diagnostique* (ou pertinente) utilisée pour déterminer l'appartenance de catégorie [Markman & Ross 2003 ; Chin-Parker & Ross 2004].

Cependant, une autre tâche d'apprentissage supervisé de catégories, l'apprentissage par inférence notamment, avec un autre but, peut mener à des différences dans la représentation [Markman & Ross 2003 ; Minda & Ross 2004].

Dans le contexte d'apprentissage de catégories, les deux fonctions fondamentales de catégories sont la classification et l'inférence [Smith 1994]. La classification et l'inférence

---

[7] Il y a un intérêt croissant pour l'apprentissage non supervisé. L'apprentissage non supervisé ne fournit pas de rétroactions (*feedback*) permettant de guider l'apprenant sur les catégories auxquelles les objets appartiennent [Clapper & Bower 1994 ; Love 2002]. Bien qu'il y a un certain nombre de travaux sur ce thème [e.g., Fried & Holyoak 1984 ; Love 2002], à ce jour, il n'y a pas suffisamment de recherche [Cf. Markman & Ross 2003] pour en tirer une idée claire telle que peut l'être avec l'apprentissage supervisé.

jouent un rôle critique dans la formation de catégories naturelles [Yamauchi & Markman 1998]. Par exemple, il est accepté que la structure *air de famille* des catégories du niveau de base émerge dans le processus compris entre la spécificité et la généralité, associé à la prédiction de propriétés (i.e., l'inférence) et la classification d'objets [Rosch et al. 1976].

De plus, la classification et l'inférence sont fonctionnellement liées et peuvent être traitées de manière semblable si l'étiquette[8] de catégorie et les propriétés de catégories sont compatibles [Anderson 1991]. Par conséquent, le fait de mieux comprendre le mécanisme impliquant ces deux fonctions et leurs impacts sur la formation de catégories, on arriverait à mieux comprendre la relation entre l'apprentissage de catégories et la formation de catégories [Yamauchi & Markman 1998].

2.2.2.1 Apprentissage par classification

Avec l'apprentissage par classification, les sujets focalisent sur l'information de propriétés utile pour diviser les exemples en plusieurs groupes [Ahn & Medin 1992]. Les théories des prototypes et des exemplaires mettent effectivement l'emphase sur les catégories comme quelque chose que les participants utilisent pour classer un nouvel objet : les participants classent les exemples un à un, avec des rétroactions (*feedback*) à chaque essai.

Avec ces deux types de théories, on souligne par ailleurs, l'importance dans le contraste entre les catégories pour prendre une décision. En d'autres termes, afin d'être utile lors de la prise de décision de classification, deux prototypes ou deux regroupements (*clusters*) d'exemples devraient être distincts l'un de l'autre. Les participants apprennent à s'occuper des propriétés qui maximisent cette distinction [Kruschke 1992 ; Nosofsky 1987].

Markman et Ross [2003], et Chin-Parker et Ross [2004] par exemple, ont souligné ce caractère distinctif par le fait que certaines propriétés d'une catégorie sont plus diagnostiques que d'autres. Les propriétés qui sont effectivement diagnostiques (par exemple, avoir le cou gonflé et être venimeux sont des propriétés diagnostiques du Cobra, alors qu'être long et mince ne le sont pas) sont particulièrement importantes pour classer de nouveaux exemples.

---

[8] Le terme *étiquette de catégorie* se réfère au symbole qui désigne le groupe particulier de stimuli et le terme *propriété de catégorie* se réfère au symbole qui désigne la caractéristique d'un stimulus particulier.

Les propriétés particulières d'une catégorie qui sont diagnostiques dépendent des autres catégories qui sont apprises. Par rapport aux catégories naturelles notamment, Markman et Ross [2003] suggèrent que la tâche de classification soit particulièrement concentrée sur les informations relatives aux *entre-catégories*, et ces informations devraient être acquises par les apprenants de classification.

2.2.2.2 Apprentissage par inférence

En revanche, avec l'apprentissage par inférence, Rips [1975], Rosch et al. [1976], et Markman et Ross [2003] entre autres, avancent que les sujets ont tendance à se concentrer sur les relations entre les exemples à l'intérieur d'une catégorie naturelle (e.g., l'air de famille parmi les exemples à l'intérieur d'une catégorie ou l'information de typicalité à propos des exemples dans une catégorie). Rehder et Burnett [2005], mais aussi Waldman et al. [1995], ou Markman et Ross [2003] soutiennent, en plus de l'approche classique précédente, l'idée novatrice selon laquelle les sujets qui connaissent les relations (ou structures) causales spécifiques liant les propriétés sont en mesure d'inférer de *propriétés à propriétés* (i.e., inférence de propriétés à partir d'autres propriétés) où la présence ou l'absence d'une propriété non observée est inférée à partir de la présence ou l'absence de propriétés spécifiques avec lesquelles elles sont reliées [Medin 1983].

En guise d'exemple, imaginons qu'on voit Voler un Oiseau non familier et souhaitons faire une inférence sur cet animal. Une première façon d'inférer, inférence par typicalité, est basée sur le cas typique de la catégorie d'Oiseau ayant le Vol comme propriété associée. Suivant cette approche, un Oiseau, comme le Moineau par exemple, qui a fortement cette caractéristique typique de la catégorie, serait jugé plus probable (i.e., en termes de pourcentage) que ne le serait un Oiseau moins typique comme l'Autruche. Une autre façon d'inférer, inférence par raisonnement causal, est de raisonner sur les causes à l'origine du Vol des Oiseaux : de larges Ailes adaptées à la taille du corps, de Forme Aérodynamique, etc. Par contre, les propriétés telles qu'un Bec de forme caractéristique par exemple, peuvent être considérées moins pertinentes pour l'inférence (bien que ces propriétés font que l'Oiseau soit plus typique dans sa catégorie). Selon cette approche, l'inférence des propriétés relève d'un raisonnement causal.

Clairement, l'apprentissage par inférence exige des sujets de connaître comment les propriétés des membres de catégories sont reliées. Ces relations internes peuvent être exprimées à la fois par le biais de l'inférence de propriétés typiques, et de l'inférence de propriétés causales. Ces deux types d'inférences sont appelés dans les deux cas, *inférence bayésienne*. L'inférence bayésienne qui utilise le paradigme de typicalité est issue du courant d'*analyse rationnelle* [Anderson 1991 ; Griffiths et al. 2008], alors que l'inférence bayésienne qui utilise le paradigme de structure causale hérite des travaux sur la théorie des modèles causaux [Waldman et al. 1995 ; Rehder, 2003 ; Rehder & Burnett 2005]. Nous y reviendrons à la section 2.3.2 plus loin, pour voir comment ces deux approches sont effectivement modélisées.

2.2.2.3 Classification versus inférence

La classification et l'inférence sont des tâches d'apprentissage supervisées. Elles procèdent avec les mêmes ensembles d'informations disponibles mais diffèrent sur la façon dont les informations sont traitées. Elles diffèrent aussi sur la façon dont les rétroactions son fournies : l'appartenance de catégorie dans un cas, et les propriétés manquantes (ou non observées) dans l'autre cas. Markman et Ross [2003] donnent une large revue de cas d'exemples ; Rehder et Burnett [2005] le complètent avec l'inférence bayésienne via les modèles causaux.

Dans les conditions d'inférence, les participants apprennent les valeurs de propriétés prototypiques [Yamauchi & Markman 1998] et les valeurs de propriétés causales [Rehder et Burnett 2005], alors que dans les conditions de classification, ils apprennent plutôt les informations sur les exemples [Yamauchi & Markman 1998]. Anderson et al. [2002] ont montré que les apprenants en classification ont de meilleurs résultats dans les tâches de classification d'exemples avec une optique globale, par contre, les apprenants en inférence se montrent plus performants dans les tâches portant sur les simples propriétés. Les participants qui se trouvent dans les conditions d'apprentissage par classification ont une moins bonne connaissance sur les propriétés prototypiques ou les propriétés causales [Rehder et Burnett 2005] que ceux dans les conditions d'apprentissage par inférence.

En finale, on retiendra, que ce soit en classification ou en inférence, qu'il est suggéré que la tâche d'apprentissage de catégories peut avoir un effet sur la façon dont les gens

acquièrent une catégorie [Minda & Ross 2004]. La classification qui a constitué pendant longtemps la base dans les études sur la catégorisation, et pouvant imposer ses propres besoins spécifiques au niveau des traitements, n'est plus l'unique moyen d'apprendre une catégorie. La classification favorise l'apprentissage des propriétés diagnostiques, l'inférence favorise, quant à elle, l'apprentissage de propriétés prototypiques [Chin-Parker & Ross 2004] et de propriétés causales [Rehder & Burnett 2005].

2.3 MODÈLES FORMELS DE CATÉGORISATION

Nous avons vu jusqu'à présent, les différents modèles de catégorisation du point de vue psychologique et expérimental. Dans cette troisième partie, la présentation va être consacrée aux modèles formels de catégorisation basés sur la similarité et aux modèles formels de catégorisation basés sur l'inférence.

2.3.1 MODÈLES BASÉS SUR LA SIMILARITÉ

La plupart des modèles formels de catégorisation développés depuis les trente dernières années prennent pour hypothèse que les catégories soient définies selon un *air de famille*. Les deux modèles les plus influents sont le modèle des prototypes et le modèle des exemplaires, qui postulent que l'individu assigne les stimuli aux catégories en se basant sur les mesures de similarité d'un air de famille.

2.3.1.1 Fonctions d'évaluation de la similarité

Les mesures de similarité sont formulées comme suit : étant donné un ensemble de $N-1$ stimuli avec des propriétés $X_{N-1}=(x_1,x_2,...,x_{N-1})$ et d'étiquettes de catégorie $Y_{N-1}=(y_1,y_2,...,y_{N-1})$, la probabilité que ce stimulus N avec les propriétés x_N soit assigné à la catégorie j est exprimée par :

$$P(y_N=j|x_N,X_{N-1},Y_{N-1})=\frac{\eta_{N,j}\beta_j}{\sum_y \eta_{N,j}\beta_y} \tag{2.1}$$

où $\eta_{N,y}$ est la similarité du stimulus x_N par rapport à la catégorie y, et β_j est interprété comme le paramètre de biais de la réponse de la catégorie y [Nosofsky 1986, p. 40]. Ainsi,

la décision de classification est une fonction des différentes similarités de catégorie, et implique l'utilisation directe du *modèle de choix de similarité* (équation 2.1) [Luce 1963 ; Shepard 1957].

Dans la formulation initiale de Shepard [1957], les paramètres de similarité sont considérés comme une interprétation explicite en termes de distances dans un espace psychologique. Il prend pour hypothèse que :

$$\eta_{N,j} = f(d_{N,j}) \tag{2.2}$$

où $f(\cdot)$ est une certaine fonction monotone décroissante, et les $d_{N,j}$ sont les distances métriques. Les questions qui se posent [Nosofsky 1986] sont de savoir [1] quelle fonction de distance utilisée pour calculer les relations de distance inter stimulus dans l'espace psychologique, et [2] quelle fonction f utilisée pour relier la similarité du stimulus à la distance psychologique. Premièrement, la fonction de distance peut avoir la forme de Minkowski de métrique r, pour laquelle la distance entre les points x_N et x_j est exprimée par :

$$d_{N,j} = \left[\sum_y \left| x_{N,y} - x_{j,y} \right|^r \right]^{1/r} \tag{2.3}$$

où $r \geq 1$, N est le nombre de dimensions des stimuli, et $x_{N,y}$ est la valeur psychologique du stimulus N à la dimension y. Quand $r = 2$, on obtient la distance euclidienne entre les stimuli, et quand $r = 1$, on obtient la distance de *Manhattan* (ou *city-block*) entre les stimuli. Ensuite, deux types de fonctions f couramment considérées sont la fonction exponentielle décroissante :

$$\eta_{N,j} = e^{-d_{N,j}} \tag{2.4}$$

et la fonction gaussienne :

$$\eta_{N,j} = e^{-d_{N,j}{}^2} \tag{2.5}$$

Le choix de ces deux fonctions résulte des travaux théoriques autant qu'empiriques [Nosofsky 1986].

2.3.1.2 Modèle des exemplaires

La grande différence entre les modèles des exemplaires et le modèle des prototypes réside dans la façon dont la similarité $\eta_{N,y}$ d'un stimulus par rapport à une catégorie est calculée.

Pour généraliser le modèle des exemplaires initial [Medin & Schaffer 1978], Nosofsky [1986] a proposé le modèle *Generalized Context Model* (GCM), formulé comme suit :

$$P\left(y_N = j \middle| x_N, X_{N-1}, Y_{N-1}\right) = \frac{\sum_j \eta_{N,j}\beta_j}{\sum_{y=1}^{m}\left(\sum_y \eta_{N,j}\beta_y\right)} \tag{2.6}$$

où m est le nombre d'exemples. Nosofsky [1986, p. 40] a démontré que le modèle des exemplaires (équation 2.6) a une forte ressemblance structurale avec le modèle de choix de Luce-Shepard (équation 2.1). Dans le modèle des exemplaires, tous les exemples de la catégorie sont stockés. La similarité du stimulus N par rapport à la catégorie j est calculée en multipliant la similarité du stimulus avec tous les exemples sauvegardés. Ce qui donne :

$$\eta_{N,j} = \prod_{i=1}^{N} s_{N,i} \tag{2.7}$$

où $s_{N,i}$ est une mesure symétrique de la similarité. Nosofsky [1986, p. 42] a démontré que la règle de similarité *multiplicative interdimensionnelle* entre deux stimuli x_N et x_i peut s'écrire comme suit :

$$\eta_{N,j} = \prod_{i=1}^{N} d_{N,j} \tag{2.8}$$

2.3.1.3 Modèle des prototypes

Par opposition, le modèle des prototypes [e.g., Minda & Smith 2001] exprimé comme suit :

$$P(y_N = j | x_N, X_{N-1}, Y_{N-1}) = \frac{\eta_{N,p_j}\beta_j}{\sum_{y=1}^{m}(\eta_{N,p_j}\beta_y)} \tag{2.9}$$

représente une catégorie j en termes d'exemple prototypique. Suivant cette formulation, la similarité du stimulus N par rapport à la catégorie j est définie par :

$$\eta_{N,j} = s_{N,p_j} \tag{2.10}$$

où p_j est l'exemple prototypique de la catégorie, et s_{N,p_j} est une mesure de similarité entre le stimulus N et le prototype p_j. Une façon de définir le prototype est de le considérer comme le centroïde de tous les exemples de la catégorie dans un certain espace psychologique[9] [Griffiths et al. 2008], c'est-à-dire :

$$p_j = \frac{1}{N_j}\sum_{i|y_i=j} x_i \tag{2.11}$$

où N_j est le nombre d'exemples de la catégories (i.e., le nombre de stimuli pour lesquels $y_i = j$).

2.3.2 MODÈLES BASÉS SUR L'INFÉRENCE

Comme discuté plus en avant, l'apprentissage par inférence suggère deux approches théoriques pour la prédiction des propriétés causales. Le premier est supporté par le modèle théorique appelé modèle rationnel de catégorisation qui utilise le paradigme d'inférence par typicalité [Anderson 1990, 1991], le second par le modèle rationnel causal qui utilise le paradigme d'inférence par structure causale [Rehder 2003].

[9] D'un homme moyen, bien entendu.

2.3.2.1 Modèle rationnel

Sur le plan cognitif, le modèle rationnel [Anderson 1990, 1991] est une alternative[10] aux modèles basés sur la similarité. En effet, le modèle des exemplaires et le modèle des prototypes discutés dans les sections précédentes tentent d'expliquer le comportement de catégorisation en termes de processus cognitifs sur fond de mesures de similarité et de choix [Luce 1963 ; Shepard 1957]. Alors que le modèle d'inférence par typicalité tel que celui d'Anderson (section 2.2.2.2 plus haut), cherche plutôt une explication dans la résolution de problèmes computationnels qui sous-tend la catégorisation.

Selon l'approche méthodologique préconisée par Anderson [1990], le modèle rationnel de catégorisation traduit le comportement humain comme une solution adaptative à un problème computationnel posé par l'environnement, plutôt que de mettre l'accent sur l'implication des processus cognitifs. Anderson [1990], Ashby et Alfonso-Reese [1995], et Rosseel [2002] défendent l'idée de base selon laquelle une des techniques de prédiction ou d'identification d'étiquette de catégorie (ou autres propriétés) non observables, consiste à utiliser des propriétés qui peuvent être observées. Ce problème de prédiction a une interprétation naturelle sous la forme d'inférence bayésienne.

Suivant le modèle rationnel, le théorème de Bayes (section 3.4.1, l'équation 3.5) permet de calculer la probabilité que l'objet N appartienne à la catégorie j étant donné les propriétés et les étiquettes de catégorie des objets $N-1$:

$$P\left(y_N = j \middle| x_N, X_{N-1}, Y_{N-1}\right) = \frac{P(x_N | y_N = j, X_{N-1}, Y_{N-1})\, P\left(y_N = j \middle| Y_{N-1}\right)}{\sum_y P(x_N | y_N = y, X_{N-1}, Y_{N-1})\, P\left(y_N = y \middle| Y_{N-1}\right)} \tag{2.12}$$

où nous supposons que la probabilité *a priori*, $P\left(y_N = j \middle| Y_{N-1}\right)$, d'un objet en provenance d'une catégorie particulière est indépendante des propriétés des objets précédents. Dans l'équation (2.12), la probabilité *a posteriori*, $P\left(y_N = j \middle| x_N, X_{N-1}, Y_{N-1}\right)$ de la catégorie j est reliée à la probabilité d'échantillon (soit la probabilité conditionnelle ou encore la

[10] Notons que certains auteurs [e.g. Ashby & Alfonse-Reese 1995 ; Rosseel 2002] travaillent à démontrer l'équivalence entre les approches basées sur la similarité (i.e., prototypes et exemplaires) et l'approche d'analyse rationnelle d'Anderson [1991]. Nous n'aborderons pas cette thématique ici car elle va bien au-delà de la présente discussion.

vraisemblance, $P(x_N|y_N = j, X_{N-1}, Y_{N-1}))$ d'un objet avec les propriétés x_N de cette catégorie, et la probabilité *a priori* de choisir cette catégorie.

L'apprentissage de catégorie dans un modèle rationnel est alors une question de détermination de la probabilité *a priori* et de la vraisemblance. La probabilité *a priori* est supposée fixe et caractérise les objets issus de la même catégorie. Cette probabilité ne dépend pas du nombre d'objets vus jusqu'à présent [Anderson 1991]. La vraisemblance de différentes dimensions, étant donné l'appartenance de catégorie, est supposée indépendante des probabilités des autres dimensions. Pour plus de détails, se référer à Anderson [1991, p.411-414].

2.3.2.2 Modèle causal

Le modèle causal [Waldman et al. 1995 ; Rehder 2003], contrairement au modèle d'inférence précédent, utilise les réseaux bayésiens comme support à la modélisation.

Les réseaux bayésiens représentent le fait qu'une variable d'effet est influencée de manière causale par ses parents immédiats (techniquement, la distribution de probabilité de la variable d'effet est conditionnellement indépendante de toute non filiation lorsque l'état des variables parents est connu). Pour qu'un réseau bayésien puisse être fonctionnel, des hypothèses doivent être émises, et spécifient la façon dont l'individu conçoit les relations causales entre les variables binaires (seuls les attributs binaires sont considérés ici, i.e., 1=présence et 0=absence). En particulier, il est supposé que l'individu voit les propriétés comme étant liées par des mécanismes causaux probabilistes. Par exemple, avec un modèle à deux variables binaires (figure 2.3a), il est supposé que quand une propriété de cause est présente (i.e., $C = 1$), il autorise une opération d'un mécanisme causal qui, avec une certaine probabilité, provoque la présence de la propriété d'effet (i.e., $E = 1$). Quand la propriété de cause C est absente, il est supposé qu'il n'y a aucune influence causale sur la propriété d'effet E.

L'autre aspect important du modèle causal concerne la prise de décision de classification. La décision s'effectue à travers l'évaluation de la probabilité qu'un exemple est susceptible d'être généré par le modèle [Rehder 2003]. Le modèle causal de la figure 2.3a a trois paramètres clés, c, m, b, de probabilité variant de 0 à 1. Suivant la théorie des

probabilités élémentaires, ces paramètres spécifient la vraisemblance qu'a le modèle de générer toute combinaison de deux propriétés C (cause) et E (effet). Le paramètre c représente la probabilité que la propriété C sera présente. Le paramètre m représente la probabilité que le mécanisme probabiliste reliant C à E sera effectivement opérationnel (i.e., provoquera la présence de E) quand C est présente. Le paramètre b représente la probabilité que E sera présent même quand elle n'est pas provoquée par C. Le paramètre b peut être interprété comme la probabilité que E soit provoquée par des causes antérieures (*background*) non spécifiées autres que C.

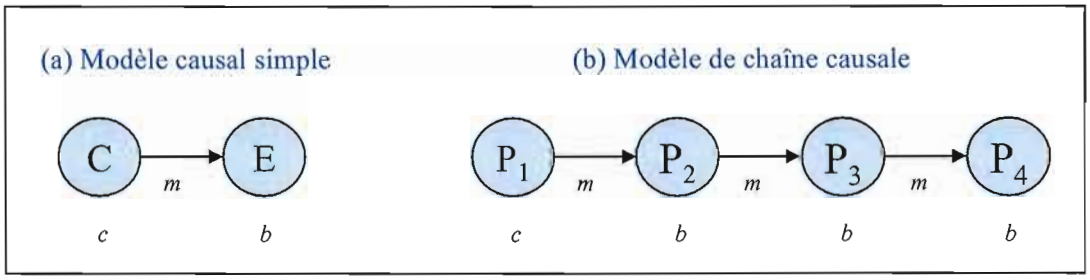


Figure 2.3 : Modèles causaux
(Adapté de Rehder 2003)

Le modèle à deux variables peut être étendu naturellement à un nombre illimité de variables connectées de manière causale suivant un patron qui ne boucle pas sur lui-même (soit une forme quelconque de graphe acyclique orienté). La figure 2.3b montre un exemple d'un modèle de chaîne à quatre variables binaires qui s'influencent séquentiellement de causes à effets : $P_1 \rightarrow P_2$, $P_2 \rightarrow P_3$ et $P_3 \rightarrow P_4$. Dans ce modèle de chaîne, on suppose que les trois mécanismes causaux interconnectant les quatre propriétés s'opèrent de manière indépendante, et ayant chacun une probabilité m. De même, on suppose que les causes antérieures de P_2, P_3 et P_4 sont aussi indépendantes, et ayant chacune une probabilité b.

La façon dont les vraisemblances générées par le modèle causal sont traduites en décision de classification dépend de la nature de la tâche de décision présentée aux participants humains. Par exemple, quand il y a deux catégories candidates auxquelles un objet est susceptible d'y appartenir, les vraisemblances de chaque catégorie du modèle causal

peuvent être combinées selon l'axiome de choix de Luce [1963] pour prédire les probabilités de choix. Ainsi, la probabilité qu'un exemple E soit classé dans la catégorie A au lieu de la catégorie B est exprimée par l'équation $P(A|E) = L_A(E)/[L_A(E) + L_B(E)]$ où L_A et L_B sont les vraisemblances que E soit généré respectivement par A et B du modèle causal. Cependant, dans les études expérimentales [e.g. Rehder 2003 ; Rehder & Burnett 2005], les participants apprennent d'abord à reconnaître chaque nouvelle catégorie. Ensuite, on leur demande d'évaluer l'appartenance de catégorie à partir d'un ensemble d'exemples. L'hypothèse de travail est que l'évaluation d'un exemple E est égale à sa vraisemblance par rapport à une catégorie du modèle causal, et multipliée par une constante K, soit [Rehder 2003] :

$$Evalutation(E) = KL(E;c,m,b) \tag{2.13}$$

où $L(E;c,m,b)$ est la vraisemblance de l'exemple E, et est une fonction de c, m et b ; K est une constante multiplicatrice qui permet de porter les vraisemblances dans l'intervalle 0 à 100.

Exemple (E)	L (E; c, m, b)	L (E; .50, .80, .20)
$P(00)$	$(1 - c)(1 - b)$	.40
$P(01)$	$(1 - c)(b)$	.10
$P(10)$	$(c)[(1 - m)(1 - b)]$	.08
$P(11)$	$(c)(m - b - mb)$	.42

Tableau 2.1 : Équations de vraisemblance d'un modèle causal à deux variables
(Adapté de Rehder 2003)

Le tableau 2.1 montre les vraisemblances qu'un modèle causal à deux variables binaires (figure 2.3a) génère pour les quatre combinaisons possibles de C et E en fonction des paramètres c, m et b. Par exemple, dans la première combinaison du tableau ci-dessus, la probabilité que C et E soient toutes les deux absentes, notée $P(00)$, est égale à la probabilité que C soit absente (c.-à-d., $(1 - c)$) fois la probabilité que E n'est pas provoquée par des causes antérieures (c.-à-d., $(1 - b)$). À noter que le paramètre m n'est pas impliqué dans cette vraisemblance parce qu'on suppose que le mécanisme causal relatif à C et à E n'opère

potentiellement pas quand C est présent. $P(01)$, $P(10)$ et $P(11)$ et leurs équations de vraisemblance $L(.)$ respectives sont les trois autres combinaisons du modèle causal.

Formellement, supposons que la propriété d'effet de catégorie x_N est causée par la propriété de cause x_{N-1}, et qu'en particulier, x_{N-1} produit x_N par le biais d'un certain mécanisme causal qui opère avec la probabilité m quand x_{N-1} est présent (et n'a aucun effet sur x_N quand x_{N-1} est absent). Avec ces hypothèses, Rehder [2003] spécifie la probabilité de x_N par rapport aux membres de la catégorie j par la vraisemblance :

$$P(x_N|y_N = j, X_{N-1}, Y_{N-1}) = m\, P(x_{N-1}|y_N = j, X_{N-1}, Y_{N-1})(1-b)+b \tag{2.14}$$

où b récapitule la probabilité que x_N soit provoquée par des causes alternatives antérieures. Autrement dit, la probabilité de x_N dans certain cas d'appartenance, est la probabilité qu'elle soit provoquée par le mécanisme causal (qui est la probabilité que x_{N-1} est présent, fois la probabilité que le mécanisme causal puisse opérer, $m\, P(x_{N-1}|y_N = j, X_{N-1}, Y_{N-1})$, ou provoqué par des causes antérieures b). L'équation (2.14) indique que la probabilité de x_N croît avec l'augmentation de la probabilité de sa cause x_{N-1} (soit $P(x_{N-1}|y_N = j, X_{N-1}, Y_{N-1})$), et celle de la force du mécanisme causal reliant x_{N-1} à x_N (soit m).

Enfin, les corrélations inter propriétés (i.e., la force de covariance entre les causes et les effets) sont la probabilité d'effet en présence de la cause moins la probabilité d'effet en l'absence de la cause. Par conséquent, le contraste probabiliste [e.g., Cheng & Novick 1990] entre une propriété de cause A et une propriété d'effet B est exprimé comme suit [Rehder 2003] :

$$\Delta_{BA} = P(B|A) - P(B|\sim A) \tag{2.15}$$

où $\sim A$ signifie l'absence de la cause A. En simplifiant, on obtient :

$$\Delta_{i,i-1} = m(1-b) \tag{2.16}$$

pour i = 2, 3, 4 (i.e., un modèle de chaîne à 4 variables, par exemple). Clairement, avec un modèle de chaîne, le contraste entre les propriétés directement connectées par des relations causales croît avec la force du mécanisme m et la décroissance de la probabilité antérieure b tel que montré à l'équation (2.16). Pour plus de détails, se référer à Rehder [2003].

2.4 DES MODÈLES DE CATÉGORISATION EN MÉTHODES DE CLASSIFICATION MACHINE

Nous ferons d'abord, une synthèse sur la thématique de catégorisation en science cognitive. Ensuite, nous présenterons la façon dont nous allons transposer les différents modèles de catégorisation humaine en méthodes de classification machine avec pour cadre le traitement de classification des bactériophages.

2.4.1 SYNTHÈSE DES APPROCHES ET MODÈLES COGNITIFS

Dans cette synthèse, nous retiendrons deux points : l'interdépendance des capacités cognitives issues des prédictions causales et des prédictions par similarité, et l'importance des considérations empiriques dans la modélisation.

Premier point : après avoir passé en revue les quelques-unes des nombreuses facettes de la catégorisation, force est de constater que les processus cognitifs sous-jacents sont complexes : [1] Quand un sujet est exposé pour la première fois à un nouvel objet, il se réfère d'abord à d'autres exemples connus (modèle des exemplaires), mais plus il apprend sur la catégorie de cet objet, plus il se fie à un objet typique de la catégorie (modèle des prototypes) ; [2] Si la tâche de classification favorise l'apprentissage des propriétés diagnostiques, l'apprentissage de propriétés prototypiques et causales, est favorisé, lui, par la tâche d'inférence ; [3] Si l'air de famille entre objets permet de souligner l'importance du rôle de la similarité dans la modélisation de la catégorisation (e.g., modèles basés sur la similarité), elle n'est toutefois pas le seul critère adopté pour la détermination de l'appartenance à une catégorie (e.g., modèles basés sur l'inférence).

Ces observations suggèrent sinon de réelles interdépendances (partie *Perspectives* plus loin, lire le paragraphe sur la théorie unifiée de catégorisation) du moins en partie entre les capacités cognitives issues des prédictions causales et des prédictions par air de famille. Vu

sous cet angle, l'approche par similarité et l'approche par inférence causale se complètent plus qu'elles se concurrencent avec pour objectif commun de comprendre un peu mieux les processus de catégorisation.

Deuxième point : certes, les différentes modélisations discutées sont basées avant tout sur des considérations théoriques, mais les aspects empiriques sont tous aussi importants. Pour preuves : [1] Nosofsky [1986] rapporte qu'effectivement le choix des deux fonctions de similarité, exponentielle décroissante et gaussienne, est basé sur des propositions théoriques mais aussi sur des expériences empiriques (section 2.3.1.1) ; [2] Anderson [1991] suggère de confronter le modèle d'analyse rationnelle de catégorisation avec les résultats empiriques obtenus en sortie du modèle. En effet, pour ce dernier, le but n'est pas de savoir si l'esprit humain fonctionne comme une quelconque formule mathématique bayésienne, mais d'avoir plutôt, peu importe la façon dont le cerveau fonctionne, les sorties d'un système le plus optimal possible. Et, les mathématiques ne servent qu'à déterminer (itérativement et empiriquement) cette optimisation.

Cette dernière remarque constitue une bonne transition nous permettant de passer du paradigme de catégorisation humaine à une utilisation automatique de la classification machine.

2.4.2 TRANSPOSITION AUX MÉTHODES DE CLASSIFICATION MACHINE

Le principal objet de la présente étude est de proposer une classification des bactériophages (microorganismes viraux biologiques). Notre proposition vise à compléter la taxonomie existante. La classification proposée repose sur l'idée originale de combiner la détection des transferts horizontaux de gènes et la reconstruction de séquences ancestrales. L'analyse phylogénétique pour établir cette classification est composée de phases de reconstruction d'arbre phylogénétique, de détection des transferts horizontaux de gènes et de reconstruction de séquences ancestrales (voir schéma général de la figure 3.5 à la section 3.6.1, page 63).

Notre propos ici est de montrer à travers la classification des bactériophages, la transposition de l'approche de catégorisation humaine dans l'approche de classification machine.

La mise en concurrence, dans la première phase de reconstruction d'arbre, de deux méthodes de classification, classification hiérarchique (section 3.3.1) versus estimation bayésienne (section 3.4.3), pour comparer les meilleurs résultats obtenus par l'une ou par l'autre des méthodes, équivaut à mettre de manière sous-jacente en compétition deux modèles de catégorisation, modèle des exemplaires (section 2.3.1.2) versus modèle rationnel (section 2.3.2.1).

Lorsqu'on utilise en complémentarité, dans les phases de détection des transferts horizontaux de gènes et de reconstruction de séquences ancestrales, deux types d'approches machines, méthodes de distances (section 3.3) et méthodes probabilistes (section 3.4), pour classer les espèces, on fait aussi de manière sous-jacente participer deux approches cognitives, mesure de similarité psychologique (section 2.3.1.1) et modèle causal (section 2.3.2.2), pour catégoriser les objets.

Nous exposerons au chapitre suivant, les différentes méthodes de classification machine mentionnées.

CHAPITRE III

—

DE LA CLASSIFICATION À LA PHYLOGÉNIE

Plan du chapitre

Si les processus cognitifs de la catégorisation ne sont pas toujours évidents à expliciter, les tâches d'appariement de données d'entrée et de sortie de la classification machine, en revanche, peuvent être très bien expliquées, aussi complexes soient-elles, par des fonctions mathématiques [Love et al. 2004].

Par le titre : *De la classification à la phylogénie*, nous cherchons dans le premier temps à comprendre les différentes méthodes proposées avec le paradigme de classification machine (ou plus simplement classification), et à retenir ensuite celles susceptibles de nous aider à éclairer sur les différents aspects de classification d'objets dans le cadre d'analyse phylogénétique.

Dans ce chapitre, nous commencerons par la distinction entre la notion d'apprentissage machine et la notion d'inférence de données, en soulignant en particulier, l'inférence appliquée à la phylogénie, suivra ensuite la discussion sur les méthodes de distances et les méthodes probabilistes ainsi que la façon dont ces méthodes ont été employées dans notre analyse phylogénétique. Avant de clore le chapitre, nous noterons certaines analogies des méthodes présentées avec les modèles de catégorisation développés au chapitre précédent, et finalement nous énumérerons quelques principaux défis méthodologiques actuels en phylogénie moléculaire, et ceux qui ont été relevés par notre présente étude.

3.1 APPRENTISSAGE

3.1.1 APPRENTISSAGE MACHINE

Nous avons vu au chapitre précédent comment fonctionnent les différents processus psychologiques de catégorisation dont l'apprentissage de catégories (section 2.2.2), nous abordons à présent les processus artificiels d'apprentissage (i.e., basés sur les théories des mathématiques et des statistiques) qui conduisent aux concepts d'apprentissage machine. Contrairement à l'apprentissage humain où une description précise n'est pas toujours évidente à expliciter, la complexité d'un problème d'apprentissage machine est comparable à la complexité de la fonction qui associe les entrées aux sorties [Love et al. 2004].

Le concept d'apprentissage machine met en avant les stratégies permettant aux machines d'apprendre à partir des expériences. En pratique, cela implique la création de programmes informatiques qui optimisent un critère de performance en faisant une analyse

de données. Mitchell [1997, 2006] et Alpaydin [2004] présentent une bonne introduction dans le domaine. Les statistiques constituent une des nombreuses sources d'inspiration pour l'apprentissage machine : le fait d'estimer la fonction de distribution inconnue est considéré comme de l'apprentissage sur un corpus d'échantillons d'un problème donné [Anderson 1958 ; Vapnik 1995].

3.1.2 APPRENTISSAGE SUPERVISÉ ET APPRENTISSAGE NON SUPERVISÉ

Considérons une machine qui reçoit certaines séquences d'entrée $x_1, x_2, x_3, ...$, où x_t est l'entrée sensorielle à l'instant t. Ces séquences d'entrée que nous appellerons par la suite *données*, pourraient correspondre par exemple, à l'image sur la rétine, au pixel dans une caméra, etc. mais aussi de manière moins évidente, à des mots dans une page Web, à la liste d'articles dans un caddie de supermarché, ou encore à des séquences biologiques dans les banques de données publiques, etc. Selon Duda et al. [2001], durant la conception d'un modèle (ou classifieur), toute méthode qui intègre de l'information à partir des données correspond à un apprentissage au sens le plus large (figure 3.1).

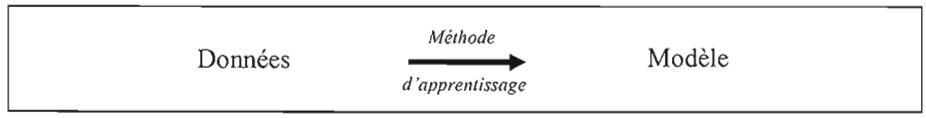


Figure 3.1 : Illustration de méthode d'apprentissage liant les données au modèle

Les algorithmes d'apprentissage sont classés principalement en fonction de deux[11] modes : l'apprentissage supervisé et l'apprentissage non supervisé.

En apprentissage supervisé, un *enseignant* fournit une étiquette de catégorie (données de valeur discrète) ou un coût (données de valeur continue) pour chaque patron dans un ensemble d'apprentissage, et cherche à réduire la somme des coûts pour ces patrons [Duda et

[11] Il existe en fait, dans la littérature d'autres variantes de l'apprentissage : l'apprentissage semi-supervisé (qui combine les exemples étiquetés et non étiquetés pour générer une fonction appropriée) [e.g., Zhu 2006], l'apprentissage par transduction (semblable à l'apprentissage supervisé, il ne construit pas explicitement une fonction mais il essaie plutôt de prédire les nouvelles sorties basées sur l'apprentissage des entrées, des sorties et teste les entrées disponibles au moment de l'apprentissage) [Kasabov et al. 2004], ou bien le méta apprentissage (*learning to learn*) qui apprend de ses propres biais inductifs basés sur ses expériences antérieures [e.g., Baxter 2000].

al. 2001]. L'apprentissage supervisé est donc une technique d'apprentissage machine avec pour but de créer une fonction à partir des données apprises. Ces données consistent en paires d'objets d'entrée $x_1, x_2, x_3, ...,$ et des sorties désirées $y_1, y_2, y_3, ...,$. La tâche de l'*apprenant* supervisé est de prédire la valeur de la fonction pour n'importe quel objet valide après avoir vu un certaine nombre d'exemples lors de l'apprentissage (i.e., appariement de paires d'entrées et sorties cibles) [Ghahramani 1999]. Par conséquent, l'apprenant doit généraliser, à partir de ce qu'il a appris des données, aux nouvelles situations rencontrées suivant l'approche de biais inductif (par opposition à l'apprentissage sans biais qui apprend par cœur, sans généralisation, et ne peut classer de nouvelles instances), soit l'ensemble des hypothèses que l'apprenant utilise pour prédire les sorties compte tenu des nouvelles entrées jamais rencontrées encore.

Pour ce qui concerne l'apprentissage non supervisé (*clustering* ou regroupement), il n'y a pas explicitement d'enseignant. Le système organise des regroupements (*clusters*) naturels de patrons d'entrée. L'aspect naturel est toujours défini explicitement ou implicitement dans le système de regroupement lui-même. Etant donné un ensemble particulier de motifs (*pattern*), chaque type d'algorithme de regroupement conduit à un regroupement particulier [Duda et al. 2001]. Dans ce type d'apprentissage non supervisé, on ne peut observer que les propriétés des objets, et non les mesures de sorties. Par conséquent, notre tâche consiste à décrire comment les données sont organisées ou regroupées [Hastie et al. 2001]. Autrement dit, l'objectif de la machine apprenante est de construire des représentations à partir des $x_1, x_2, x_3, ...,$ qui peuvent être servies comme base de raisonnement, de prise de décision, de prédiction, de communications, etc. [Ghahramani 2004].

La distinction entre les modes d'apprentissage supervisé et non supervisé devait être cependant plus nuancée dans le cadre de nos travaux où il était plus question de méthodes de distances et de méthodes probabilistes. En effet, si notre utilisation des méthodes de distances

était globalement non supervisée[12], deux cas d'utilisation suivants montrent, en revanche, que la dichotomie supervisée et non supervisée ne peut être généralisée.

Duda et al. [2001] ont rangé les méthodes d'estimation bayésienne, tantôt comme non supervisées tantôt comme supervisées, selon les hypothèses faites sur les paramètres du modèle à apprendre : *« L'apprentissage non supervisé de paramètres d'une densité de mélange est similaire à l'apprentissage supervisé de paramètres d'une densité de composantes »* [Duda et al. 2001, p. 533-534]. Autre cas : Zhu [2007, p. 22] a montré que l'approche Tree-HMM, une estimation du maximum de vraisemblance qui associe le modèle HMM et une topologie d'arbre (i.e., Tree), est de type apprentissage semi-supervisé.

Par conséquent, pour éviter la redondance conceptuelle entre les concepts d'apprentissage et d'inférence de données, plutôt que de distinguer les apprentissages supervisés et non supervisés, nous avons opté pour la distinction[13] entre méthodes de distances et méthodes probabilistes. Ces deux familles de méthodes sous-tendent en effet par essence l'apprentissage machine à travers le paradigme d'inférence de données en général [Baldi & Brunak 2001 ; Mitchell 1997, 2006] et d'inférence de données en phylogénie en particulier [Durbin et al. 2006]. En effet, l'inférence phylogénétique peut être définie comme le processus qui permet l'évaluation de l'histoire de l'évolution grâce à l'analyse d'un ensemble choisi de données [Swofford et al. 1996]. Nous aborderons, dans la section suivante, le paradigme d'inférence phylogénétique.

3.2 INFÉRENCE PHYLOGÉNÉTIQUE

Dans cette section, nous commencerons avec les définitions sur la similarité, la dissimilarité et la distance, suivies de celles sur l'arbre phylogénétique et la condition des quatre points. Ces définitions sont importantes par la suite pour comprendre l'approche d'inférence de distances et l'approche d'inférence probabiliste qui constituent les deux principales approches d'inférence phylogénétique.

---

[12] Excepté dans les cas spécifiques d'utilisation où par exemple, le nombre de regroupements k est donné explicitement à l'avance par l'expérimentateur dans les regroupements par partition de *k-Means* [Ewens & Grant 2005, p.472-473] (voir aussi la section 3.3.2). Mais cette méthode n'a pas été utilisée dans cette étude.

[13] Nous laissons ici volontairement de côté les méthodes de parcimonie puisque celles-ci n'ont pas été utilisées dans nos travaux.

3.2.1 SIMILARITÉ, DISSIMILARITÉ ET DISTANCE

Une similarité ou dissimilarité est toute application à valeurs numériques qui permet de mesurer le lien entre les individus d'un même ensemble ou entre les variables. Pour une similarité le lien est d'autant plus fort que sa valeur est grande.

Un indice de similarité (ou plus simplement une similarité) sur un ensemble Ω est une application s de $\Omega * \Omega$ dans R^+ qui vérifie les deux conditions suivantes :

$$\begin{aligned} &\text{c1 (symétrie) :} && \forall (a,b) \in \Omega * \Omega && s(a,b) = s(b,a) \\ &\text{c2 :} && \forall (a,b) \in \Omega * \Omega \ \ avec\ a \neq b && s(a,a) = s(b,b) > s(a,b) \end{aligned} \tag{3.1}$$

Un indice de dissimilarité (ou plus simplement une dissimilarité) est une application d qui satisfait à la condition c1 et à c2' qui suit :

$$\text{c2' :} \quad \forall a \in \Omega \qquad d(a,a) = 0 \tag{3.2}$$

Une distance est un indice de dissimilarité qui vérifie en plus les deux propriétés suivantes :

$$\begin{aligned} &\text{d1 :} && d(a,b) = 0 && ssi\ \ a = b \\ &\text{d2 (inégalité triangulaire) :} && d(a,b) \leq d(a,c) + d(c,b) && \forall a,b,c \in \Omega \end{aligned} \tag{3.3}$$

Une distance d est qualifiée *métrique* si elle satisfait les propriétés d1 et d2 (inégalité triangulaire) définies ci-dessus [Leclerc 1996 ; Duda et al. 2001 ; Ewens et Grant 2005]. L'espace Ω muni d'une distance d est appelé *espace métrique*. Parmi les principales métriques proposées, on a la distance euclidienne, la distance de Manhattan, la distance de Mahalanobis et la distance de Minskowski [voir par exemple pour plus de détails, Duda et al. 2001].

3.2.2 ARBRE PHYLOGÉNÉTIQUE ET CONDITION DES QUATRE POINTS

Un arbre phylogénétique est une représentation graphique de l'histoire de l'évolution d'un groupe de taxa (section 4.1.1) réalisée à partir de l'étude d'un ou de plusieurs caractères. Un arbre phylogénétique montre les relations de parentés entre des entités supposées avoir un ancêtre commun.

Un arbre est défini comme un graphe acyclique connexe. Dans un arbre, chaque paire de sommets est reliée par un unique chemin, et le nombre de sommets (i.e. nœuds) dépasse toujours de 1 le nombre de branches. Un arbre est binaire si chaque sommet a soit un ou trois voisins. Un arbre binaire est enraciné si un nœud r a été sélectionné et appelé *racine** (figure 3.2, c'est un arbre binaire sauf au niveau de la racine). Dans un arbre phylogénétique, la racine représente l'ancêtre commun de tous les autres sommets. Les nœuds intermédiaires représentent les plus proches ancêtres communs des espèces (i.e., taxa ou feuilles) en aval (descendantes).

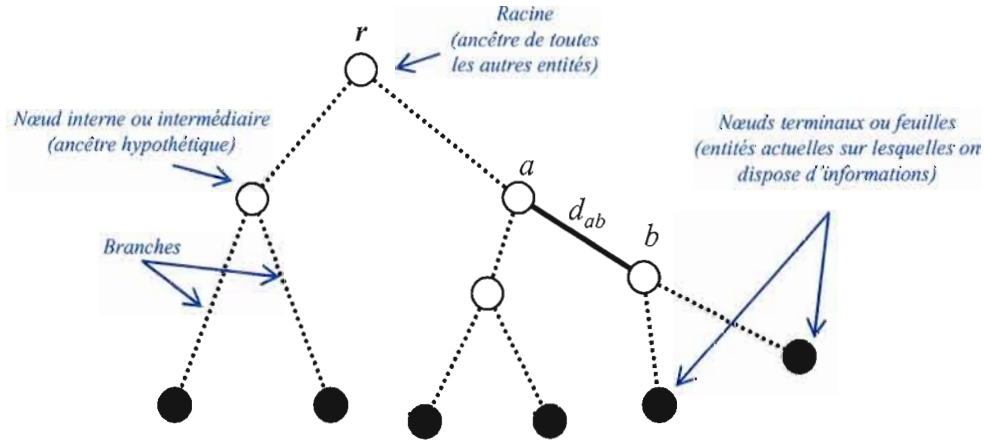


Figure 3.2 : Un arbre phylogénétique

Deux paramètres importants des arbres phylogénétiques (enracinés ou non enracinés) sont la topologie et la longueur des branches. La topologie se réfère à la ramification de l'arbre associée au temps de divergence. La longueur de branche est souvent utilisée pour représenter en quelque sorte la distance temporelle d_{ab}, entre les événements d'évolution a et b. Les nœuds internes ou intermédiaires sont représentés par des symboles a et b. Les

symboles que l'on peut observer sont au niveau des feuilles. La probabilité d'évolution du sommet a au sommet b est notée : $P(d_{ab})$.

En classification, on s'intéresse à la représentation d'une distance d sur un ensemble d'éléments X par un arbre additif (i.e., arbre phylogénétique ou X-arbre). Dans un tel arbre, X est l'ensemble des feuilles, les arêtes ou branches sont valuées[14], et la longueur des chemins entre les feuilles approxime la distance d [Barthélemy & Guénoche 1988 ; Leclerc & Makarenkov 1998 ; Guénoche et al. 2004]. Suivant les domaines d'application, les nœuds correspondent à des catégories (psychologie cognitive), à des ancêtres communs (évolution moléculaire, filiation de textes) ou tout simplement à des classes d'objets. Il est bien connu que cette représentation est exacte si est seulement si d est une distance d'arbre, c'est-à-dire qu'elle vérifie la *condition des quatre points* [Zaretskii 1965 ; Buneman 1971] :

$$\begin{aligned} &\forall a,b,c,d \in \Omega \\ &d(a,b)+d(c,d) \leq \max\left\{ d(a,c)+d(b,d)\ ,\ d(a,d)+d(b,c) \right\} \end{aligned} \tag{3.4}$$

En d'autres termes, parmi les trois sommes $d(a,b)+d(c,d)$, $d(a,c)+d(b,d)$ et $d(a,d)+d(b,c)$, les deux plus grandes sont égales.

3.2.3 INFÉRENCE DE DISTANCES

Une distance réelle entre un groupe d'espèces de l'ensemble X est rarement une distance d'arbre. Si d est une distance sur un ensemble d'espèces X, d' est une distance sur X et il existe une constante C telle que : $d(a,b) = C\,d'(a,b)$ pour tous $a,b \in X$. Ewens et Grant [2005] ont montré qu'il existe une relation entre les mesures de distances d'arbre et les mesures de distances de type dérivé d'arbre, car un arbre reconstruit à partir de d' a la même topologie que celui reconstruit à partir de d.

L'apprentissage par inférence peut être schématisé de différentes façons, la figure 3.3 en est une illustration. La figure 3.3a illustre une méthode d'inférence de distances (e.g., classification par regroupement hiérarchique, voir la section 3.3.1) qui utilise les

[14] par des distances évolutives, c.-à-d. des taux de mutations.

données (terme synonyme de *séquences* par la suite, qui désigne les séquences de protéines – *aminoacides** ou de vecteurs binaires de données morphologiques) pour inférer l'arbre (terme synonyme de *modèle* par la suite) phylogénétique correspondant.

3.2.4 INFÉRENCE PROBABILISTE

L'inférence probabiliste comprend principalement deux approches appelées *estimation du maximum de vraisemblance* et *estimation bayésienne* [Durbin et al. 2006]. Avec la première, on pose traditionnellement la question : *« Supposons que cette hypothèse (arbre) soit vraie, quelle est la probabilité des données (séquences) observées ? »*, alors qu'avec la seconde, *« Quelle est la probabilité que cette hypothèse (arbre)* [parmi une collection d'autres hypothèses ou arbres] *soit vraie sachant les données (séquences) observées soient vraies.»* [Ewens & Grant 2005].

Dans un cas, le meilleur arbre est celui qui maximise la vraisemblance. La stratégie consiste à effectuer des recherches sur les topologies d'arbre, et pour chaque topologie T, de trouver les longueurs de branches τ qui maximisent la vraisemblance $P(D|T,\tau)$, où D représente les données de séquences. La topologie et les longueurs de branches qui donnent le maximum global de cette vraisemblance désignent l'arbre désiré. La figure 3.3b illustre l'approche d'estimation du maximum de vraisemblance (section 3.4.2) qui utilise les séquences pour estimer la vraisemblance de l'arbre phylogénétique correspondant.

Dans l'autre cas, si on connaissait la probabilité *a priori* $P(T,\tau)$, on pourrait utiliser le théorème de Bayes pour calculer la probabilité *a posteriori* $P(T,\tau|D)$, laquelle nous donne justement l'information dont on a besoin pour l'estimation, c'est-à-dire la probabilité de chaque arbre phylogénétique inféré sachant les séquences observées. La figure 3.3c illustre l'approche d'estimation bayésienne (section 3.4.3) qui évalue les arbres phylogénétiques inférés en tenant compte des séquences observables disponibles.

Pour mettre en perspective la discussion à venir sur la classification des bactériophages (chapitre IV), nous abordons à présent les méthodes de distances et les méthodes probabilistes sous l'angle théorique d'abord, puis ensuite, dans le cadre d'utilisation spécifique de l'inférence phylogénétique.

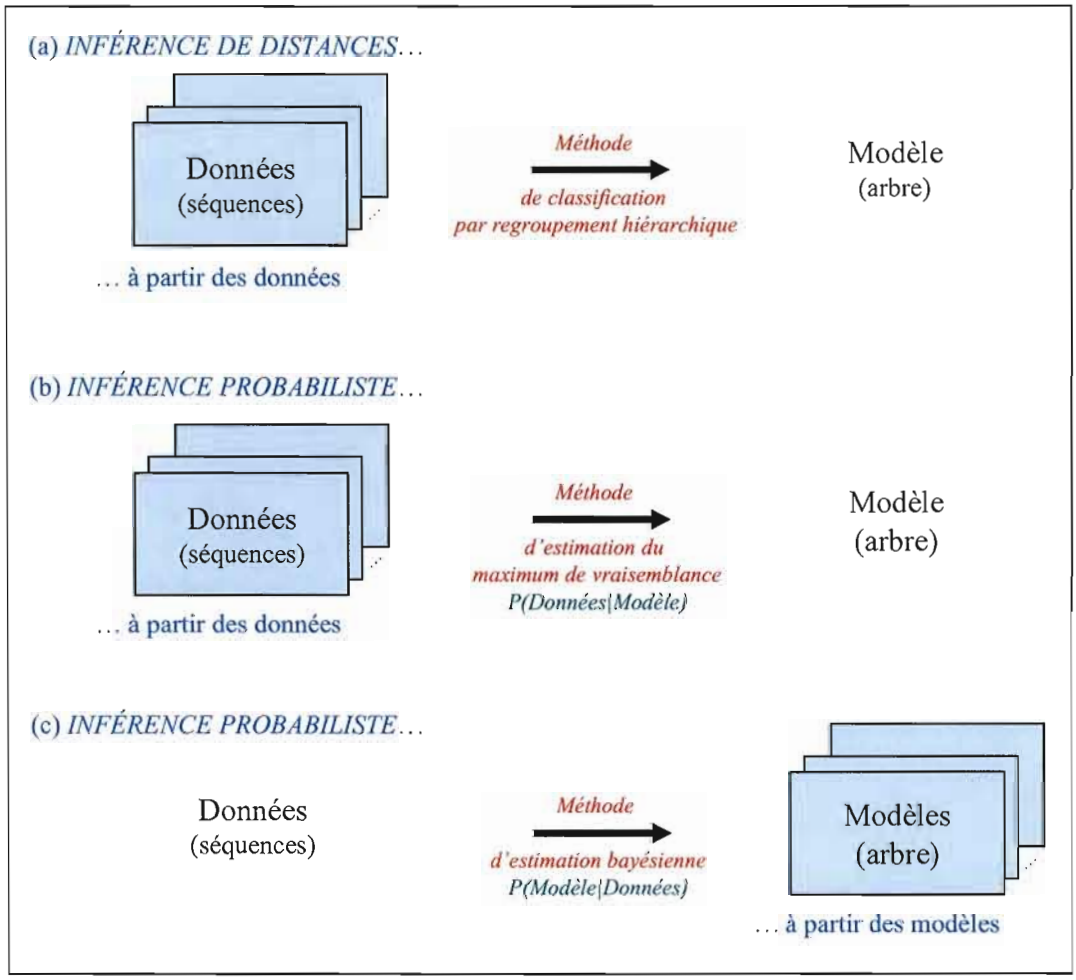


Figure 3.3 : Illustration de méthodes d'inférence de distances (a) et d'inférence probabiliste (b et c)

3.3 MÉTHODES DE DISTANCES

Lorsque l'on étudie d'importants échantillons de données comme c'est souvent le cas en analyse phylogénétique, il est souvent nécessaire de disposer de procédés de regroupements permettant de répartir les données en catégories afin de mieux comprendre les phénomènes sous-jacents.

C'est précisément le rôle de la classification par regroupement de calculer les groupes (*clusters*) de données. La classification par regroupement vise, en particulier, à nommer les classes, à résumer les données. Les données appartenant à une même classe reçoivent le

même nom et leurs propriétés communes sont celles de la classe. Il est difficile de donner une définition concise de la classification par regroupement [Hartigan 1975, Jain & Dubes 1988]. On se contentera d'indiquer que la classification par regroupement est une technique de partitionnement ou de regroupement d'objets basée sur les mesures de distances et de similarités [Berkhin 2002 ; Jain & Dubes 1988]. Il existe deux grandes familles de méthodes : le regroupement par hiérarchie et le regroupement par partition [Asselin de Beauville & Kettaf 2005].

3.3.1 CLASSIFICATION PAR REGROUPEMENT HIÉRARCHIQUE

L'objectif d'un regroupement hiérarchique est de représenter l'ensemble Ω des objets à classer par un ensemble de parties hiérarchiquement emboîtées. La méthode consiste à effectuer une suite de regroupements en agrégeant à chaque étape les objets ou les groupes d'objets les plus proches [Ward 1963].

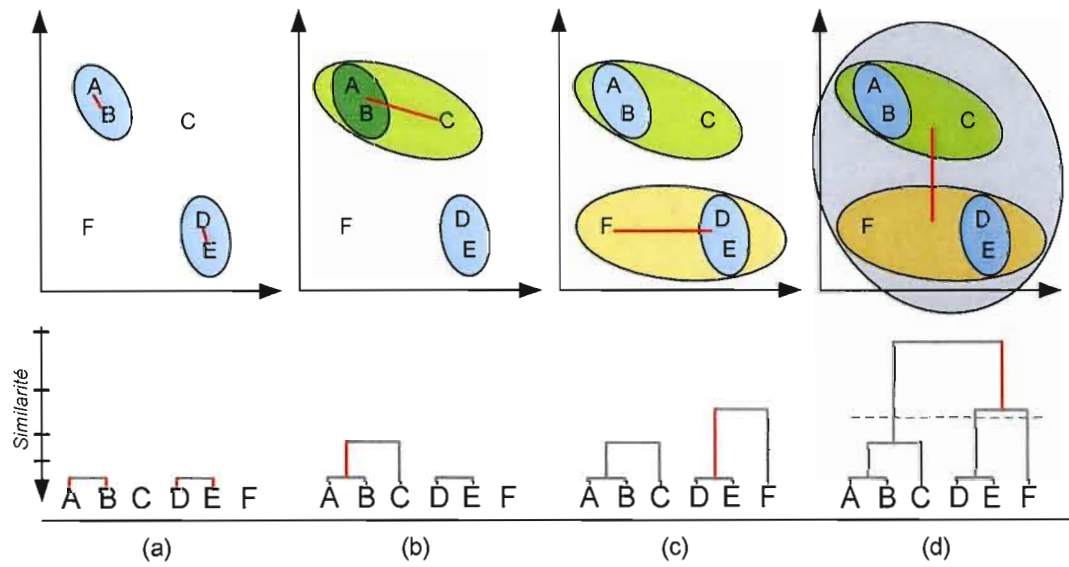


Figure 3.4 : Exemple de regroupement hiérarchique

Il existe deux types de méthodes hiérarchiques : les méthodes agglomératives (*bottom-up*) et les méthodes descendantes (*top-down*) [Jain & Dubes 1988]. On s'intéresse ici au regroupement hiérarchique ascendant. Son principe algorithmique est comme suit (figure 3.4,

de l'itération (a) à l'itération (d)) : d'abord, chaque objet est considéré comme représentant d'une classe ; l'étape qui suit consiste à trouver la paire de classes de plus grande similarité et à les agréger en utilisant une similarité entre un groupe et un objet et/ou entre deux groupes (voir les lignes rouges de l'exemple à la figure 3.4). Ce processus est répété jusqu'à ce que tous les objets soient classés dans une même classe (itération (d)).

Le résultat est représenté par un dendogramme, c'est-à-dire un diagramme qui enregistre les séquences de fusion.

3.3.2 CLASSIFICATION PAR REGROUPEMENT PAR PARTITION

L'approche la plus fréquemment utilisée par les méthodes de regroupement par partition, est celle qui optimise des critères dépendants des distances entre objets et centres de classes (inerties), ou représentant unique de classes, en utilisant une procédure itérative de type recherche du gradient. L'algorithme de partitionnement typique est *K*-Means [Hartigan 1975 ; Hartigan & Wong 1979] (où le représentant est le centroïde, soit la moyenne pondérée des points de la classe) et sa variante floue, *Fuzzy K*-Means [Zadef 1965]. L'idée de base de l'algorithme est de commencer par une partition initiale, puis à l'étape t, disposant d'une solution P^t, on recherche dans l'espace des partitions possibles une partition P^{t+1} meilleure (ou au moins égale) que P^t au sens du critère que l'on s'est fixé. On réitère ce processus jusqu'à sa stabilisation.

Une autre méthode de classification par partition est la famille des méthodes de partitionnement probabiliste [Jain & Dubes 1988 ; Berkhin 2002 ; Asselin de Beauville & Kettaf 2005]. Ces méthodes utilisent l'hypothèse sur la distribution des objets à classifier. Par exemple, on peut considérer que les objets de chacune des classes suivent une loi normale. Le problème qui se pose alors est de déterminer quels sont les paramètres des lois (moyenne, variance) et à quelle classe les objets ont le plus de chances d'appartenir. C'est la démarche notamment de l'algorithme paramétrique espérance-maximisation (EM) appliqué aux modèles de mélange de distributions [Dempster et al. 1977].

Du fait que ces méthodes n'ont pas été utilisées dans nos travaux, elles ne seront pas développées davantage par la suite. Cette brève présentation a toutefois son intérêt pour souligner l'analogie entre ces méthodes de classification machine et le modèle des prototypes

de catégorisation discuté à la section 2.3.1.3. Lire par ailleurs, la section 3.5, "*Analogies entre méthodes de classification et modèles cognitifs*".

3.4 MÉTHODES PROBABILISTES

Les méthodes probabilistes proposent un grand spectre de modèles, allant d'une simple distribution à une grammaire stochastique complexe en incluant plusieurs distributions de probabilité implicites. Lorsqu'un type de modèle est choisi, les paramètres du modèle doivent être inférés à partir des données.

Pour inférer les données, plusieurs stratégies peuvent être employées (se référer pour plus de détails, à Ripley [1996] et MacKay [1992]) dont l'estimation du maximum de vraisemblance et l'estimation bayésienne. Elles ont pour but de classer les modèles en fonction de leur vraisemblance $P(Données|Modèle)$ pour la première, et de leur probabilité *a posteriori* $P(Modèle|Données)$ pour la seconde. Définissons au préalable, le cadre probabiliste bayésien qui sert de lien entre lesdites estimations probabilistes.

3.4.1 CADRE PROBABILISTE BAYÉSIEN

En statistique inférentielle, le théorème de Bayes est utilisé pour mettre à jour ou actualiser les estimations d'une probabilité ou d'un paramètre quelconque, à partir des observations et des lois de probabilité de ces observations [Durbin et al. 2006]. Le théorème de Bayes énonce des probabilités conditionnelles : étant donné deux évènements A et B, il est possible de déterminer la probabilité de A *sachant* B, si l'on connaît les probabilités de A, de B et de B sachant A :

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{3.5}$$

Chaque terme de (équation 3.5) a une dénomination usuelle. Ainsi, le terme $P(A)$ est la probabilité *a priori* de A. Elle est *antérieure* au sens qu'elle précède toute information sur B. Le terme $P(A|B)$ est appelé la probabilité *a posteriori* de A sachant B. Elle est *postérieure*, au sens qu'elle dépend directement de B. Le terme $P(B|A)$, pour un B connu, est appelé la *fonction de vraisemblance* de A. Et, le terme $P(B)$ est appelé la probabilité marginale de B.

Du point de vue de l'apprentissage et de l'inférence, l'approche bayésienne introduit une description intuitive simple : si X implique Y, et Y est vrai, alors X est très plausible. L'approche bayésienne permet, en effet, d'assigner un degré de plausibilité à n'importe quelle proposition, hypothèse ou modèle. Cette approche est en ceci remarquable qu'elle peut être prouvée au sens strictement mathématique, que c'est la seule façon consistante de raisonner en présence de l'incertitude [Baldi & Brunak 2001]. Et en ce sens, les axiomes de Cox-Jaynes [Jaynes 1994] suggèrent que les degrés de plausibilité satisfont les règles de probabilités. Par conséquent, les calculs de probabilité sont tout ce dont la machine a besoin pour inférer, sélectionner et comparer les modèles.

3.4.2 ESTIMATION DU MAXIMUM DE VRAISEMBLANCE TREE-HMM

L'estimation du maximum de vraisemblance (ML) est une méthode statistique courante utilisée pour inférer les paramètres de la distribution de probabilité d'un échantillon donné. Supposons que l'on souhaite inférer les paramètres $\theta = \{\theta_i\}$ d'un modèle M, à partir d'un ensemble de données D. La stratégie la plus évidente est de maximiser la probabilité $P(D|\theta,M)$ à travers tous les paramètres θ possibles de Θ. C'est ce qu'on appelle le critère du maximum de vraisemblance. Formellement, on écrit :

$$\theta^{ML} = \arg\max_{\theta \in \Theta} P(D|\theta, M) \tag{3.6}$$

De manière générale, lorsqu'on traite $P(\alpha|\beta)$ comme une fonction de α, on se réfère à une probabilité, et comme une fonction de β, on se réfère à une vraisemblance. Notons qu'une vraisemblance n'est pas une distribution de probabilité ni une densité, mais simplement une fonction de la variable β. Lorsqu'on se réfère à une probabilité, on effectue une estimation de la densité de distribution de la probabilité a posteriori ou estimation bayésienne. Lorsqu'on se réfère à une vraisemblance, on effectue une estimation de la fonction de vraisemblance ou estimation ML [Durbin et al. 2006, p. 312].

Les méthodes ML sont très gourmandes en temps de calculs machines. Par exemple, l'explosion combinatoire des topologies d'arbres à explorer limite ces méthodes à estimer un petit nombre de topologies. Certains auteurs, tel que Felsenstein [1981], proposent des

stratégies d'heuristique (e.g., algorithme de *Pruning* de Felsenstein [1981]) afin d'améliorer les performances des méthodes ML.

Ls technique Tree-HMM combine un modèle probabiliste HMM (*Hidden Markov Models*) et une topologie d'arbre (Tree). Tree-HMM autorise deux processus stochastiques (ou chaînes de Markov, voir Annexe A.1) : un dans le temps et un autre dans l'espace. L'architecture du modèle est donnée *de facto* par la topologie de l'arbre proposé a priori. À l'instar d'autres modèles HMM standard (Annexe A.2), le modèle Tree-HMM est défini, outre par une topologie d'arbre (Tree), par trois composantes HMM : les états initiaux, les probabilités d'émission et les probabilités de transition. Les Tree-HMM ont été introduites par Felsenstein et Churchill [1996] et Yang [1996] dont le but est d'améliorer les modèles phylogénétiques en autorisant les variations de vitesse de substitution propre à chacun des sites d'une séquence biologique. Suivant le même principe, mais de manière complémentaire (insertion/délétion au lieu de substitution), Diallo et al. [2006] ont développé une méthode de reconstruction *ancestrale* basée sur les scénarios d'insertions et de délétions de nucléotides (*Indel*).

Etant donné un *alignement** de séquences multiple A de longueur de région d'Indel L, sur un arbre phylogénétique T, un chemin π dans le modèle Tree-HMM et une séquence d'états $\pi = \pi_0, \pi_1, \ldots, \pi_L, \pi_{L+1}$, on a :

$$P(\pi, A) = P(\pi_0, A_0) \prod_{i=1}^{L+1} e(A[i] | \pi_i)\, a(\pi_i | \pi_{i-1}) \tag{3.7}$$

où $e(.)$ est la probabilité d'émission et $a(.)$ est la probabilité de transition. Diallo et al. [2006] ont démontré que pour n'importe quelle reconstruction $\hat{A}$ de A, on a $L(\hat{A}) = P(\pi, A)$, où $L(\hat{A})$ est la vraisemblance de $\hat{A}$, et π le chemin correspondant à $\hat{A}$. Donc, maximiser $P(\pi, A)$ revient à maximiser $L(\hat{A})$.

3.4.3 ESTIMATION BAYÉSIENNE PAR ÉCHANTILLONNAGE MCMC–MH

L'estimation bayésienne est une méthode statistique courante utilisée pour inférer la distribution de la probabilité a posteriori. Reprenons l'exemple des paramètres $\theta = \{\theta_i\}$ de la

section précédente et supposons qu'il existe une distribution de probabilité à travers ces paramètres. La stratégie consiste maintenant à maximiser la probabilité $P(\theta|D,M)$ en conditionnant θ à tous les modèles M possibles. Dans ce contexte, l'équation (3.5) du théorème de Bayes peut être exprimée à nouveau comme suit [Durbin et al. 2006] :

$$P(\theta|D,M) = \frac{P(D|\theta,M)\,P(\theta|M)}{P(D|M)} \tag{3.8}$$

où $P(\theta|D,M)$ est la probabilité a posteriori des paramètres θ sachant les données D et le modèle M, $P(D|\theta,M)$ est la vraisemblance de θ pour un D donné, $P(\theta|M)$ est la probabilité a priori et $P(D|M)$ est la probabilité marginale.

La probabilité a priori est la connaissance préétablie de l'expérimentateur sur les paramètres θ et le modèle M. Cette liberté de choisir une probabilité a priori fait des statistiques bayésiennes un sujet de controverse par moment. Mais Huelsenbeck et Ronquist [2001], Larget et Simon [1999] et Durbin et al. [2006], notamment, soutiennent qu'elle représente un cadre très commode, du moins en biologie, pour intégrer les connaissances préalables dans une estimation bayésienne. Des travaux en cours, par exemple Larget et al. [2005] ou Blanquart et Lartillot [2006] y défendent la même position.

L'inférence de la probabilité a posteriori peut se faire suivant deux approches. La première consiste en la maximisation de la probabilité a posteriori (MAP). De manière analogue aux méthodes ML (équation 3.6), on cherche l'unique instance θ du modèle maximisant la probabilité $P(\theta|D,M)$. A cet égard, la méthode ML serait un cas particulier de l'approche MAP, pour lequel on considère toutes les réalisations du modèle comme a priori équiprobables [Kuhner et al. 1995]. Cependant, on est plus intéressé en analyse phylogénétique par la distribution elle-même, soit la densité de probabilité a posteriori en tant que fonction définie sur l'ensemble des réalisations du modèle [Blanquart & Lartillot 2006]. Or, le calcul des probabilités a posteriori des arbres phylogénétiques étant analytiquement impossible [Ronquist & Huelsenbeck 2003]. Il est donc nécessaire de recourir à des méthodes numériques permettant d'échantillonner la distribution de probabilité a posteriori. Ces méthodes d'échantillonnage constituent la seconde approche.

Les méthodes bayésiennes d'échantillonnage présentent elle-même plusieurs techniques différentes, à savoir : la transformation à partir d'une distribution uniforme [Feller 1971], l'échantillonnage à partir d'une Dirichlet[15] par rejet [Law & Kelton 1991], l'échantillonnage de Gibbs [Geman & Geman 1984] et l'échantillonnage par l'algorithme de Metropolis-Hastings (MH) [Metropolis et al. 1953 ; Hastings 1970]. La technique MH associée aux chaînes de Markov Monte Carlo est particulièrement utilisée pour l'estimation de la distribution de probabilité a posteriori des arbres phylogénétiques [Mau & Newton 1997 ; Yang & Rannala 1997; Larget & Simon 1999]. Nous allons détailler cette technique dans la section qui suit.

3.4.3.1 Echantillonnage MCMC

La technique des chaînes de Markov Monte Carlo (MCMC) qui inclut les méthodes de Monte Carlo de marche au hasard (*random walk*) visent à échantillonner des distributions de probabilités inconnues. L'idée sous-jacente aux MCMC est qu'une chaîne de Markov (Annexe A.1), prenant la forme d'une marche guidée à travers l'espace multidimensionnel des paramètres, peut être utilisée pour estimer une distribution de probabilité en échantillonnant les valeurs de ces paramètres de façon périodique. L'approximation de la distribution sera d'autant plus exacte que le nombre de pas effectués par la chaîne de Markov sera élevé [Lewis 2001]. L'algorithme généralement utilisé en analyse phylogénétique est le Metropolis-Hastings (MH) basé sur les travaux de Metropolis et al. [1953] et Hastings [1970].

3.4.3.2 Algorithme de Metropolis-Hastings

Posons θ, le vecteur de N paramètres x_n d'un modèle, $\theta = \{x_1, x_1, ..., x_N\}$ et $(\theta, d\theta') = \{(x_1, dx_1'), (x_2, dx_2'), ..., (x_N, dx_N')\}$, l'ensemble des mouvements MCMC applicables à chacun des paramètres x_n d'une réalisation θ du vecteur de paramètres. Le mouvement (x_n, dx_n') permet la transformation de θ en une réalisation θ', et plus particulièrement de la valeur de son paramètre x_n en une nouvelle valeur x_n'. Le noyau

[15] En statistiques, une distribution de Dirichlet, noté Dir(α), est une famille de distributions de probabilités continues multivariée paramétrées par le vecteur a de réels positifs.

stochastique MCMC $(\theta, d\theta')$ peut être vu dans le cas général comme une application $F(\theta) = \theta'$.

L'algorithme de Metropolis-Hastings est dit « de rejet ». Il permet d'accepter, ou de rejeter le nouvel état θ', conditionnellement à la probabilité a posteriori de cet état. La décision d'acceptation/réjection est formellement exprimée comme suit :

$$P_{accepter}(\theta'|\theta) = \min\left(1, \underbrace{\frac{P(\theta'|D,M)}{P(\theta|D,M)}}_{M(\theta,d\theta')} \times \underbrace{\frac{Q(\theta',d\theta)}{Q(\theta,d\theta')}}_{H(\theta,d\theta')}\right) \tag{3.9}$$

Le premier terme $M(\theta, d\theta')$ est le rapport de Metropolis, c'est-à-dire le ratio des probabilités a posteriori du *nouveau* mouvement sur le mouvement *courant* $(\theta, d\theta')$:

$$\begin{aligned} M(\theta, d\theta') &= \frac{P(D|\theta',M)/P(D|M)}{P(D|\theta,M)/P(D|M)} \times \frac{P(\theta'|M)}{P(\theta|M)} \\ &= \underbrace{\frac{P(D|\theta',M)}{P(D|\theta,M)}}_{\substack{Rapport \\ vraisemblances}} \times \underbrace{\frac{P(\theta'|M)}{P(\theta|M)}}_{\substack{Rapport \\ priors}} \end{aligned} \tag{3.10}$$

En appliquant à la nouvelle formulation le théorème de Bayes (équation 3.8), le rapport des probabilités a posteriori (équation 3.10) est égal au rapport des vraisemblances multiplié par le rapport des probabilités a priori. La probabilité marginale $P(D|M)$ est une constante et se simplifie dans le rapport de Metropolis. Le second terme est le rapport de Hastings :

$$H(\theta, d\theta') = \frac{Q(\theta', d\theta)}{Q(\theta, d\theta')} \tag{3.11}$$

Il s'agit d'une correction apportée lorsque les mouvements $(\theta, d\theta')$ ont des probabilités asymétriques d'être réalisés. Le rapport de Hastings se lit comme la probabilité $Q(\theta, d\theta')$ de faire le mouvement retour $(\theta', d\theta)$ compensant exactement la modification aller $(\theta, d\theta')$, sur la probabilité d'avoir effectué la modification aller.

Enfin, la probabilité d'acceptation du mouvement MCMC $(\theta, d\theta')$ est égale à $\min(1, M(\theta, d\theta') \times H(\theta, d\theta'))$, ce qui implique que si la modification d'un paramètre a permis d'améliorer la probabilité a posteriori du modèle, le rapport de Metropolis-Hastings est supérieur à 1 et le nouvel état θ' est accepté avec une probabilité 1. Au contraire, si le mouvement MCMC produit un nouvel état θ' de moins bonne probabilité a posteriori, le nouvel état est accepté avec une probabilité égale au rapport de Metropolis-Hastings.

3.5 ANALOGIES ENTRE MÉTHODES DE CLASSIFICATION ET MODÈLES COGNITIFS

Les méthodes de classification venant d'être exposées et comparées aux modèles cognitifs de catégorisation du chapitre II, présentent certaines analogies évidentes (tableau 3.1).

D'abord, le concept de distances dans l'espace métrique (section 3.2.1) est équivalent au concept de distances dans l'espace psychologique dont nous faisions mention à la section 2.3.1.1. Il est question en effet des même types de mesures (euclidienne, Manhattan, etc.) utilisées pour calculer à la fois les distances entre les objets dans l'espace métrique et les distances entre les stimuli dans l'espace psychologique (tableau 3.1a).

À remarquer également qu'il y a d'autres points de ressemblance entre les méthodes de distances et les modèles de catégorisation basés sur la similarité. Premièrement, entre le regroupement hiérarchique (section 3.3.1) et le modèle des exemplaires (section 2.3.1.2), la tâche qui consiste, dans le premier cas, à classer tous les objets de Ω (ensemble d'objets), s'apparente à la tâche qui consiste, dans le seconde cas, à mémoriser tous les exemplaires (ensemble des stimuli) (tableau 3.1b). Ensuite, comme mentionné par ailleurs, entre la méthode de regroupement par partition (section 3.3.2) et le modèle des prototypes (section 2.3.1.3), la même notion de centroïde est considérée respectivement comme le représentant de la classe et le prototype de la catégorie (tableau 3.1c).

Soulignons que ceci est vrai lorsqu'on utilise le concept de distances. Mais lorsqu'on le considère suivant l'approche probabiliste (tableau $3.1d_1$), le regroupement par partition prend la forme alternative de l'algorithme paramétrique espérance-maximisation (EM) (section 3.3.2). De manière similaire, avec l'approche probabiliste, la description du modèle des

prototypes rejoint celle du modèle d'inférence bayésienne (tel que le modèle rationnel, voir ci-dessous) puisqu'il utilise l'approche d'inférence de propriétés typiques (section 2.2.2.2).

	Modèles cognitifs de catégorisation (chapitre II)	**Méthodes de classification machine (chapitre III)**	
Approche de distances	**Espace psychologique** (*distances entre stimuli*)	**Espace métrique** (*distances entre objets*)	(a)
	Modèle des exemplaires (*mémorisation de tous les exemplaires*)	**Classification par hiérarchie** (*classement de tous les objets*)	(b)
	Modèle des prototypes (*centroïde = prototype de la catégorie*)	**Classification par partition** (*centroïde = représentant de la classe*)	(c)
Approche probabiliste	**Modèle rationnel** (*inférence par typicalité*)	**Classification par partition – algorithme EM** (*approche probabiliste*)	(d_1)
	Modèle rationnel (*prédiction à partir des propriétés observables*)	**Estimation bayésienne** (*échantillonnage de propriétés a posteriori*)	(d_2)
	Modèle causal (*structure de chaînes causales*)	**Estimation du maximum de vraisemblance (ML) suivant le modèle Tree-HMM** (*structure de chaînes de Markov*)	(e)

Tableau 3.1 : Modèles cognitifs versus méthodes de classification

Note : l'approche de distances et l'approche probabiliste (mentionnées dans le tableau ci-dessus) constituent, avec l'approche de parcimonie, trois ensembles de méthodes utilisées en analyse phylogénétique. Nous nous sommes surtout intéressés dansle cadre de notre étude aux deux premières approches (lire par ailleurs la section 3.6.3.1).

De même, les méthodes d'inférence probabiliste sont semblables à plus d'un titre aux modèles basés sur l'inférence, à savoir : [1] La méthode d'estimation bayésienne (section 3.4.3) et le modèle rationnel (section 2.3.2.1) : les deux utilisent le même théorème de Bayes. La méthode de classification bayésienne l'utilise pour échantillonner les probabilités a posteriori d'un modèle sachant les données observées disponibles. Alors que le modèle cognitif rationnel l'utilise pour prédire ou identifier les étiquettes de catégorie (ou autres propriétés) non observables à partir des propriétés qui peuvent être observées (tableau 3.1d_2) ; [2] La méthode d'estimation du maximum de vraisemblance (ML) Tree-HMM (section 3.4.2) et le modèle causal (section 2.3.2.2) : si Tree-HMM est structuré par des chaînes de Markov où le passage des états est conditionné par les probabilités de transition et d'émission, le modèle causal est, quant à lui, structuré par des chaînes causales où l'état d'effet est conditionné par les probabilités *c, m* et *b* de cause à effets (tableau 3.1e).

Par ailleurs, notons également le fait que certains auteurs utilisent des termes (intentionnellement) flous tels que l'*apprenant* ou l'*enseignant* [Duda et al. 2001 ; Ghahramani 1999] (section 3.1.2) pour désigner tantôt une machine tantôt une personne.

Finalement, au vu des exposés dans les deux chapitres, l'approche de catégorisation humaine et l'approche de classification machine présentent à biens des égards une similitude certaine.

3.6 APPLICATIONS EN ANALYSE PHYLOGÉNETIQUE

Nous poursuivons dans cette partie, la description de la façon dont les méthodes de distances et les méthodes probabilistes ont été appliquées dans nos travaux de classification des bactériophages (chapitres IV et V).

3.6.1 VUE GÉNÉRALE DES MÉTHODES UTILISÉES

Pour une plus grande clarté, nous proposons d'aborder le sujet avec une vision d'ensemble d'utilisation des différentes méthodes au cours des nombreuses étapes d'analyse phylogénétique.

	Méthodes probabilistes		**Méthodes de distances**	
	Nom	***Fonction***	***Nom***	***Fonction***
Reconstruction d'arbre			Calculs des dissimilarités (traitement *ad hoc*)	Permet de calculer les dissimilarités inter-génomiques
	Estimation bayésienne (**MrBayes**)	Permet de reconstruire l'arbre phylogénétique	Classification hiérarchique (**NJ**)	Permet de reconstruire l'arbre phylogénétique
Détection THG			Optimisation de distances R&F (**HGT-Detection**)	Permet de détecter des THG
Reconstruction ancestrale	Estimation ML suivant le modèle Tree-HMM (**Ancestor**)	Permet de reconstruire les séquences de protéines ancestrales		

Tableau 3.2 : Méthodes utilisées dans l'analyse phylogénétique

Les méthodes de distances et les méthodes probabilistes sont rangées en colonne suivant leur type (tableau 3.2). On trouve des méthodes suivies entre parenthèse des noms de programme (terme générique par la suite, pour désigner logiciels, outils et autres implémentations *ad hoc*). Pour chaque méthode, on associe une brève description de la fonction de traitement phylogénétique. Ces méthodes sont par ailleurs, rangées en ligne en fonction des trois phases d'analyse phylogénétique : la reconstruction d'arbre, la détection de THG et la reconstruction de séquences ancestrales.

En détaillant les trois phases d'analyse phylogénétique, on décrit diverses opérations de traitement qui nécessitent pour chacune des programmes spécifiques. La figure 3.5 présente un schéma général d'utilisation des méthodes.

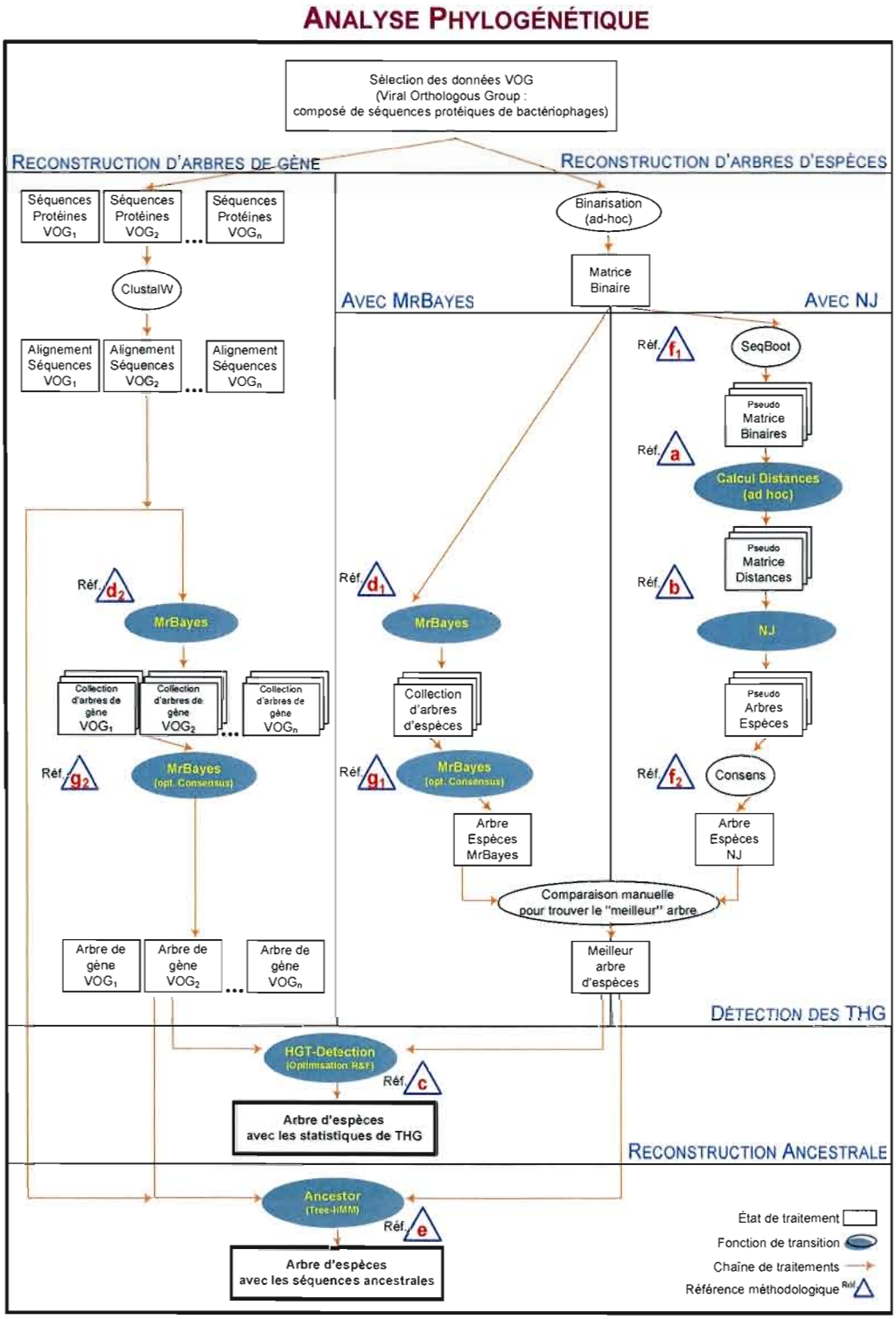


Figure 3.5 : Schéma général montrant les différentes opérations de traitements phylogénétiques

La figure 3.5 montre les principaux programmes : MrBayes, NJ, HGT-Detection, Ancestor, traitements *ad hoc* (en ovale plein) et les références méthodologiques associées (de (a) à (g) en triangle). Le but de l'analyse est de trouver l'arbre phylogénétique d'espèces avec les statistiques de transferts horizontaux de gènes et l'arbre phylogénétique d'espèces avec les séquences de protéines ancestrales reconstruites (en gras sur la figure 3.5).

Les sections subséquentes détaillent l'emploi des différentes méthodes qui vont se rapporter successivement au schéma général via les références indiquées sur la figure 3.5 (en lettre rouge entourée de triangle bleu : a, b, c, …, g_1, g_2).

3.6.2 APPLICATIONS DES MÉTHODES DE DISTANCES

Le concept de distances a été utilisé pour calculer les dissimilarités inter-génomiques entre les espèces. Avec les dissimilarités inter-génomiques comme intrant, le programme *Neighbor-Joining* (NJ) [Saitou & Nei 1987 ; Studier et Keppler 1988] basé sur la méthode de classification hiérarchique ascendante a été utilisé pour reconstruire l'arbre phylogénétique des bactériophages. Ensuite, le principe d'optimisation par les distances topologiques de Robinson et Foulds implémenté dans le programme HGT-Detection/T-Rex a été utilisé pour détecter les transferts horizontaux de gènes.

3.6.2.1 Coefficients de corrélation

Pour mettre en œuvre une classification d'un ensemble de séquences (ou données), il est important de considérer les propriétés, les traits caractéristiques et les variables décrivant ces séquences afin de déterminer les distances qui les séparent. La figure 3.6 illustre un ensemble de n observations sur un ensemble de k variables dans X. La comparaison des variables consiste à mesurer, deux-à-deux, la similarité/dissimilarité entre les variables dans X.

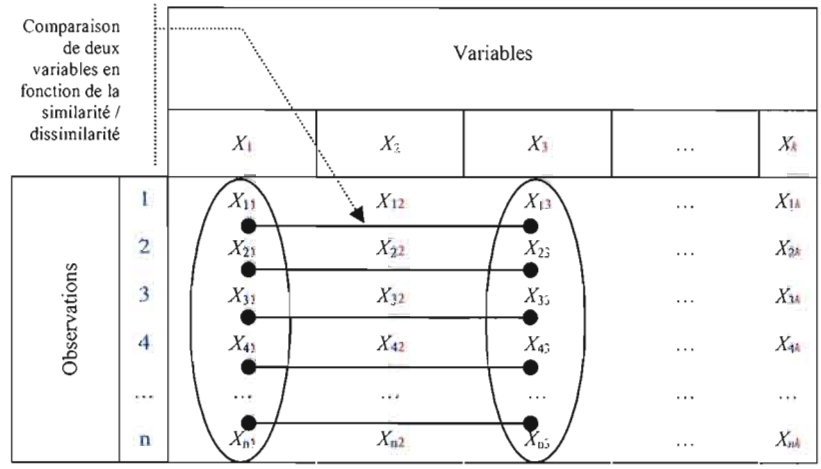


Figure 3.6 : Comparaison de variables en fonction de la similarité/dissimilarité des observations

Les similarités les plus classiques sont calculées à partir de la covariance entre vecteurs X. La valeur absolue de la corrélation est un indice de similarité. Plus la ressemblance entre séquences est grande, plus l'indice de similarité est petit (inversement, plus l'indice de similarité est élevé). Il est facile de transformer une similarité en dissimilarité par complémentation à 1 [Leclerc 1996 ; Asselin de Beauville & Kettaf 2005].

Parmi les principaux coefficients de corrélation, celui de Jaccard (appelé aussi Tanimoto) est sans doute l'un des plus communément utilisée en biologie [Duda et al. 2001 ; Mirkin & Koonin 2003 ; Glazko et al. 2005]. Le coefficient de corrélation de Jaccard est formulé comme suit :

$$J_{ij} = \frac{X_{ij}}{X_{ii} + X_{jj} - X_{ij}} \tag{3.12}$$

où X_{ij}, X_{ii} et X_{jj} sont respectivement les produits scalaires de $X_i X_j$, $X_i X_i$ et $X_j X_j$.

Dans nos travaux, on s'était intéressé en particulier aux variables de type vecteur binaire. Chaque variable X représente une espèce de bactériophage étudiée. Soient deux vecteurs binaires $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iM})$ et $X_j = (x_{j1}, x_{j2}, x_{j3}, \ldots, x_{jM})$, où M est le

nombre de groupes de données appelés VOG*, chaque composante des vecteurs X_i et X_j indique la présence ou l'absence d'un gène dans les VOG (section 4.2.2.1). Les corrélations ont été calculées suivant le principe de similarité entre deux vecteurs binaires lequel est basé sur les quatre nombres suivants :

a_{11} : *nombre de caractéristiques communes aux deux vecteurs,* b_{10} : *nombre de caractéristiques possédées par le vecteur i et non par le vecteur j,* c_{01} : *nombre de caractéristiques possédées par le vecteur j et non par le vecteur i,* d_{00} : *nombre de caractéristiques que ne possèdent ni le vecteur i ni le vecteur j.*	X_i : 1 0 1 0 1 1 1 0 1 X_j : 1 1 0 0 1 1 0 1 0 a_{11} c_{01} b_{10} d_{00} a_{11} a_{11} b_{10} c_{01} b_{10}

Avec des vecteurs binaires, le coefficient de corrélation de Jaccard (équation 3.12) est transformé comme suit :

$$a_{11}/(a_{11}+b_{10}+c_{01}) \tag{3.13}$$

3.6.2.2 Dissimilarités inter-génomiques

Nous montrerons, au prochain chapitre, un exemple de classification de séquences de protéines de bactériophages. Les distances entre les espèces de bactériophages ont été mesurées grâce aux dissimilarités inter-génomiques. Le calcul de ces dissimilarités a été effectué en fonction d'un coefficient de corrélation entre vecteurs binaires représentant les espèces. Outre le coefficient de corrélation de Jaccard (3.12) mentionné, la littérature [Mirkin & Koonin 2003 ; Dutilh et al. 2004 ; Glazko et al. 2005] suggère également d'autres coefficients tels que les coefficients de corrélation de Maryland Bridge et la Moyenne Pondérée :

Coefficient de Maryland Bridge : $$MB_{ij} = \frac{1}{2}\left(\frac{X_{ij}}{X_{ii}} + \frac{X_{ij}}{X_{jj}}\right) \tag{3.14}$$

Coefficient de la Moyenne Pondérée : $$WA_{ij} = \frac{\sqrt{X^2{}_{ii} + X^2{}_{jj}}}{\sqrt{2}X_{ii}X_{jj}} X_{ij} \tag{3.15}$$

Tous les coefficients de corrélation mentionnés ont été successivement évalués dans nos travaux.

La valeur J_{ij} (équation 3.12) est comprise entre 0 et 1, et égale au nombre de bits à *Un* dans les deux vecteurs binaires, divisé par le nombre de bits à *Un* dans l'un ou l'autre des vecteurs [Glazko et al. 2005]. Le coefficient de Maryland Bridge représente la proportion moyenne du recouvrement dans les deux vecteurs comparés. La valeur MB_{ij} (équation 3.14) est le nombre de gènes partagés, normalisé par la moyenne harmonique de la taille du génome [Mirkin & Koonin 2003]. Autre variante, la valeur de la Moyenne Pondérée WA_{ij} (équation 3.15) est le nombre de gènes partagés, normalisé par la moyenne pondérée de la taille du vecteur [Dutilh et al. 2004].

Une matrice de dissimilarités inter-génomiques a été calculée par le programme *ad hoc* que nous avons implémenté, à partir d'une matrice binaire de présence / absence de gènes (voir la figure 3.5a de la section 3.6.1, page 63).

3.6.2.3 Reconstruction phylogénétique avec NJ

Une fois les dissimilarités inter-génomiques calculées, elles servent aux méthodes de classification pour reconstruire un arbre phylogénétique. Dans un arbre phylogénétique, seules les distances additives sont considérées. Deux algorithmes d'inférence phylogénétique basés sur les méthodes de classification hiérarchique ascendante sont communément utilisés : *Unweighted Pair-Group Method using Arithmetic averages* (UPGMA) [Sokal & Michener 1958] et *Neighbor-Joining* (NJ) proposé par Saitou et Nei [1987] et modifié par Studier et Keppler [1988].

Basé sur la plus simple des méthodes de distances, UPGMA reconstruit l'arbre à partir d'une matrice de distances d'évolution. UPGMA utilise la technique de lien moyen (*average linkage*) pour calculer les distances moyennes inter-groupes afin d'agréger (agglomérer) les groupes entre eux, et produire un arbre final enraciné et ultramétrique. Bien qu'il soit simple

et performant, l'hypothèse d'*horloge moléculaire**,[16] utilisée par cet algorithme n'est en général pas valide, voire rejetée [Lepage et al. 2006], et par conséquent cela limite son intérêt en pratique. NJ présente une alternative bien plus populaire parmi les phylogénéticiens. Car il vise à corriger les faiblesses de la méthode précédente en éliminant l'hypothèse d'horloge moléculaire. C'est une méthode qui utilise un algorithme d'approximation pour reconstruire l'arbre en recherchant la paire de feuilles voisines i et j qui minimise la longueur de l'arbre, pour finalement, les joindre ensemble.

Le calcul des plus proches voisins peut être résumé ainsi : étant donné un arbre T avec des longueurs de branches l_i, on peut le reconstruire à partir des distances deux à deux entre ses feuilles $\{d_{ij}\}$. La recherche des feuilles i et j les plus proches voisines consiste à calculer la distance minimale d_{ij}, moins les distances moyennes de toutes les autres feuilles $(r_i + r_j)$ [Durbin et al. 2006] :

$$D_{ij} = d_{ij} - (r_i + r_j) \;\; ; \qquad r_i = \frac{1}{|L| - 2} \sum_{k \in L} d_{ik} \tag{3.16}$$

où k est le nœud parent (interne) des deux feuilles i et j et $|L|$ est la taille de l'ensemble L des feuilles actives (à l'étape de l'itération de l'algorithme). Durbin et al. [2006, p. 190] ont prouvé que la paire de feuilles i et j pour laquelle D_{ij} est minimal sont les voisines les plus proches.

L'algorithme NJ suit le procédé agglomératif, introduit par Sattah et Tversky [1977], qui, à chaque étape, sélectionne une paire de taxons à agglomérer. Lors de cette agglomération, le nouveau nœud créé remplace les deux nœuds sélectionnés et la matrice de distances est réduite en remplaçant les distances aux deux nœuds agglomérés par celles au nouveau nœud créé. Contrairement à UPGMA, NJ produit un arbre final non enraciné et additif, et autorise des taux de substitution différents sur les branches. La faible complexité polynomiale de NJ, $O(n^3)$, où n est le nombre d'espèces, lui permet de traiter de très grands

[16] L'hypothèse d'évolution qui satisfait l'ultramétricité de distances consiste à poser que les taux de mutation ou les vitesses de changements sont identiques sur toutes les branches de l'arbre et donc que la distance est proportionnelle au temps d'évolution.

jeux de données. De nombreuses simulations informatiques [Kuhner & Felsenstein 1994 ; Nei 1991 ; Ota & Li 2000] ont montré que NJ est une méthode de reconstruction phylogénétique relativement fiable. En outre, NJ est statistiquement consistent sous de nombreux modèles d'évolution [Gascuel 1997 ; Atteson 1999]. Il existe d'autres méthodes de distances dont *Bio Neighbor-Joining* (BioNJ) [Gascuel 1997], FITCH [Felsenstein 1997] et *Method of Weighted least-squares* (MW) [Makarenkov & Leclerc 1999].

Dans cette étude, l'arbre phylogénétique d'espèces des bactériophages a été reconstruit avec NJ implémenté dans le package PHYLIP [Felsenstein, 2004], à partir de la matrice de dissimilarités inter-génomiques (voir la figure 3.5b de la section 3.6.1, page 63).

3.6.2.4 Principe d'optimisation de RF dans la détection de THG

Après avoir reconstruit l'arbre phylogénétique des bactériophages et les arbres de chaque gène de protéine (ou simplement gène, par la suite) particulier, nous avons procédé à la détection des transferts horizontaux de gènes (THG) grâce aux techniques d'optimisation proposées dans le programme *HGT-Detection* [Makarenkov et al. 2008]. Ce programme propose des critères d'optimisation dont celui basé sur la distance topologique de Robinson et Foulds (RF) [Robinson & Foulds 1981]. En appliquant l'optimisation RF sur l'arbre d'espèces et l'arbre de gène, Makarenkov et al. [2008] ont démontré qu'il est possible de détecter des THG. Le principe d'optimisation RF dans la détection de THG sera développé plus en détail au chapitre suivant (section 4.3).

Des statistiques de transferts THG inter et intra-groupes ont été calculées par confrontations de distances topologiques entre l'arbre d'espèces et les différents arbres de gène grâce au programme HGT-Detection (voir la figure 3.5c de la section 3.6.1, page 63).

3.6.3 APPLICATIONS DES MÉTHODES PROBABILISTES

Comme alternative à la méthode de distances NJ, le programme MrBayes a été utilisé pour reconstruire l'arbre phylogénétique d'espèces. Le principe du modèle Tree-HMM implémenté dans le programme Ancestor a été utilisé pour reconstruire les séquences de protéines ancestrales.

3.6.3.1 Reconstruction phylogénétique alternative avec MrBayes

Lors d'une reconstruction phylogénétique, il est courant d'utiliser plusieurs familles de méthodes concurrentes afin de vérifier les résultats qu'elles donnent (i.e., l'arbre juste). Les méthodes du maximum de parcimonie[17] [Farris 1970 ; Fitch 1971], les méthodes du maximum de vraisemblance [Felsenstein 1981] et les méthodes bayésiennes [Huelsenbeck & Ronquist 2001] sont concurrentes aux méthodes de distances, telles que NJ discutée plus tôt. Sans occulter les deux premières méthodes citées qui pourraient être appelées à servir dans des développements subséquents, la méthode bayésienne a été retenue dans les présents travaux pour les avantages qu'elle semble apportée (voir la suite du document).

Bien qu'étant l'une des approches probabilistes parmi les plus anciennes, l'approche bayésienne, comparée aux autres méthodes, n'a été que récemment appliquée au problème de la reconstruction phylogénétique [Li 1996 ; Mau 1996 ; Rannala & Yang, 1996].

Soit $T = \{\tau, v, \lambda\}$ un arbre (τ), une combinaison de longueurs de branches (v) et un ensemble de paramètres (λ) d'un modèle d'évolution donné et D un ensemble des données. La distribution stationnaire est la probabilité a posteriori représentée ici par la probabilité conjointe de τ, v et λ. Le noyau stochastique MCMC peut être vu comme une application de $f(T) = T'$, où (T, T') est une modification de la topologie, c'est-à-dire, un mouvement stochastique. En appliquant les équations (3.10 et 3.11), le rapport Metropolis-Hastings est exprimé comme suit :

$$MH(T,T') = \overbrace{\underbrace{\frac{P(D|T')}{P(D|T)}}_{\substack{\text{Ratio des} \\ \text{vraisemblances}}} \times \underbrace{\frac{P(T')}{P(T)}}_{\substack{\text{Ratio des} \\ \text{priors}}}}^{\text{Rapport Metropolis}} \times \overbrace{\underbrace{\frac{P(T|T')}{P(T'|T)}}_{\substack{\text{Ratio des} \\ \text{etats proposes}}}}^{\text{Rapport Hastings}} \tag{3.17}$$

soit, le produit du ratio des vraisemblances (i.e., la vraisemblance du nouvel état T' sur la vraisemblance de l'état courant T) fois le ratio des probabilités a priori (i.e., la probabilité a priori du nouvel état T' sur la probabilité a priori de l'état courant T) fois le ratio des états

[17] Les méthodes du maximum de parcimonie partent du principe que le meilleur scénario phylogénétique est celui qui minimise le nombre de substitutions nécessaires afin de rendre compte des données [Farris 1970]. Pour une topologie particulière, on peut inférer le score de parcimonie comme le nombre minimum de substitutions nécessaires à la production des données observées. Parmi toutes les topologies possibles, on choisira celle dont le score est minimal.

proposés (i.e., la proposition du nouvel état T' sur la proposition de l'état courant T). Finalement, la probabilité d'acceptation d'un nouvel état T' est :

$$A(T,T') = \min(1, MH(T,T')) \tag{3.18}$$

Une variable aléatoire uniforme variant entre 0 et 1 est alors tirée. Si ce nombre est inférieur à *MH*, alors l'état proposé est accepté et $T = T'$, sinon la chaîne reste dans son état d'origine T. Ce processus est répété un très grand nombre de fois et la séquence des états visités forme une chaîne de Markov qui peut être échantillonnée de façon périodique. Les différents états par lesquels passent les MCMC sont souvent désignés sous le terme de *générations*. Les échantillons tirés de la chaîne de Markov représentent un échantillon de la distribution des probabilités a posteriori. Décrits ainsi, les échantillons de la chaîne de Markov forment la densité de probabilités conjointes des topologies, des longueurs de branches, et des paramètres du modèle de substitution $\{\tau, v, \lambda\}$.

S'il y a convergence (section 3.6.4.2) vers un état stationnaire, une collection d'arbres est générée, et un arbre phylogénétique unique peut être désigné en calculant le consensus majoritaire.

Les premières expériences phylogénétiques bayésiennes ont montré que les algorithmes MCMC-MH sont computationnellement plus efficaces que les méthodes du maximum de vraisemblance [Larget & Simon 1999]. Des avantages inhérents à ces méthodes sont la facilité d'interprétation des résultats (section 3.6.4.2) et la possibilité d'incorporation d'information a priori.

De plus, il est possible de quantifier l'incertitude dans les hypothèses d'évolution [Larget et al. 2002, 2005]. Grâce à sa flexibilité qui permet l'analyse de jeux de données de taille conséquente sous des modèles d'évolution moléculaire de plus en plus complexes et incorporant un très grand nombre de paramètres [Ronquist & Huelsenbeck 2003 ; Huelsenbeck et al. 2004 ; Lartillot & Philippe 2004], l'approche bayésienne apparaît donc particulièrement prometteuse pour le futur de la reconstruction phylogénétique.

Reste que l'approche bayésienne est encore jeune. Certains problèmes comme la compréhension de la relation entre valeurs de bootstrap (section 3.6.4.1) et probabilités a posteriori bayésiennes nécessitent clairement des études complémentaires [Delsuc & Douzery 2004]. En particulier, l'évaluation de la sensibilité des méthodes bayésiennes à la distribution des paramètres définis a priori et au modèle d'évolution des séquences utilisé apparaît comme l'une des priorités.

L'implémentation de ces algorithmes dans des programmes tels que BAMBE [Simon & Larget 1998] et MrBayes [Huelsenbeck & Ronquist 2001] a contribué à l'essor actuel de l'approche bayésienne. Comme alternative à NJ, le programme MrBayes a été choisi pour la reconstruction phylogénétique. MrBayes a été utilisé pour inférer d'abord, à partir de la matrice binaire une collection d'arbres avant de retenir par consensus majoritaire (i.e., ne conserver que les groupes qui apparaissent souvent, soit ≥50%) l'arbre phylogénétique d'espèces le plus probable (voir la figure $3.5d_1$ et $3.5g_1$ de la section 3.6.1, page 63). Puis, pour inférer, à partir des alignements de séquences de chaque gène particulier, les collections d'arbres avant de retenir par consensus majoritaire les meilleurs arbres phylogénétiques de gène (voir la figure $3.5d_2$ et $9g_2$ de la section 3.6.1, page 63). Notons que les alignements de séquences ont été réalisés préalablement par le programme ClustalW [Thompson et al. 1994].

Le point sur la collection d'arbres et l'arbre consensus sera discuté à la section 3.6.4.2.

3.6.3.2 Principe de Tree-HMM appliqué dans la reconstruction ancestrale

Après avoir reconstruit l'arbre phylogénétique d'espèces, nous avons procédé à la reconstruction des séquences ancestrales. La reconstruction ancestrale s'effectue en deux étapes : la reconstruction du scénario d'insertion et de délétion le plus vraisemblable et l'inférence des aminoacides (i.e., protéines) à chaque position des ancêtres où la présence d'un caractère a été prédite. Ces deux étapes ont été réalisées respectivement par les algorithmes de Diallo et al. [2006] et Felsenstein [1981] implantés dans le programme *Ancestor* [Diallo et al. 2006]. Ancestor utilise le modèle Tree-HMM sous-jacent. Le principe de Tree-HMM appliqué dans la reconstruction ancestrale sera développé plus en détails au prochain chapitre (section 4.4).

Des séquences protéiques ancestrales ont été inférées à partir de l'arbre d'espèces, des arbres de gène particuliers et des séquences alignées des VOG grâce au programme Ancestor (voir la figure 3.5e de la section 3.6.1, page 63).

3.6.4 TECHNIQUES DE VALIDATION DES RÉSULTATS

Un des sujets de controverse est celui de l'évaluation de la confiance que l'on peut avoir en un arbre reconstruit, en un groupe de taxons trouvé ou en des longueurs de branches générées.

Pour estimer le degré de confiance accordé aux nœuds d'un arbre phylogénétique, les méthodes de distances doivent recourir à une technique de validation par rééchantillonnage de données, alors que les méthodes bayésiennes proposent une réponse quasi naturelle grâce à la technique des arbres échantillonnés *de facto* durant l'inférence bayésienne. Les deux techniques seront présentées ici.

3.6.4.1 Rééchantillonnage de données

Les méthodes de distances ne peuvent utiliser directement la statistique classique dû au fait que les distributions de probabilité des paramètres à estimer sont généralement inconnues ou ne peuvent s'exprimer en termes simples [Darlu & Tassy 2004]. Une façon de contourner la difficulté consiste à faire appel à une technique usuelle de rééchantillonnage développée par Efron [1982], c'est-à-dire la méthode de *Bootstrap*[18].

La méthode de bootstrap [Efron 1982 ; Felsenstein 1985] consiste à tirer au hasard avec remise un ensemble de K caractères parmi les K caractères constituant les données [Darlu & Tassy 2004]. Ce tirage se faisant avec remise, cela signifie que le nouvel échantillon, constitué, lui aussi, de K caractères, peut contenir des caractères présents plusieurs fois, car retirés après remise, et au contraire, d'autres caractères absents, n'ayant jamais été tirés. Cela revient à pondérer les caractères de manière aléatoire. Le nouvel échantillon (soit la pseudo-matrice binaire dans notre analyse) fait ensuite l'objet d'une analyse phylogénétique conduisant à l'obtention d'un arbre (soit le pseudo-arbre dans notre

[18] Le bootstrap est une méthode statistique pour trouver la variance d'une mesure due à l'utilisation d'un échantillon fini de données sous-jacentes. Bien que cette méthode soit devenue commune en analyse phylogénétique, l'interprétation des valeurs de bootstrap reste un débat ouvert [Soltis & Soltis 2003].

analyse). Cette procédure de rééchantillonnage peut être effectuée *N* fois, suivie chaque fois par une recherche d'arbre. En fin de bootstrap, on est en possession de *N* arbres qui peuvent éventuellement être différents.

Si l'on souhaite tester la robustesse d'un *clade** ou *monophylie** particulière (soit la ségrégation d'un ensemble particulier d'espèces sur sa propre branche, autrement dit, un ancêtre et tous ses descendants), il suffit de dénombrer combien de fois on le retrouve parmi les *N* arbres. Si l'on donne la valeur 1 à la présence et 0 à l'absence du clade que l'on souhaite tester, le paramètre testé est l'occurrence du clade. Par exemple, un clade retrouvé dans 95% des échantillons signifie qu'il y a 5 chances sur 100 de se tromper en disant que le clade n'existe pas. D'autres techniques de rééchantillonnage dont *Delete-half-Jackknifing** et *Permutation** [Efron 1982 ; Duda et al. 2001] peuvent aussi être utilisées.

La technique de bootstrap a été appliquée dans notre analyse. Des pseudo-matrices binaires ont été générées par le programme SeqBoot inclus dans le package PHYLIP (voir la figure $3.5f_1$ de la section 3.6.1, page 63). Puis, à partir des pseudo-matrices de distances, des pseudo-arbres ont été reconstruits par le programme NJ. Finalement, un arbre d'espèces de consensus majoritaire (i.e., ne conservent que les groupes qui apparaissent souvent, soit ≥50%) a été finalement généré par le programme Consens inclus dans le package PHYLIP (voir la figure $3.5f_2$ de la section 3.6.1, page 63).

3.6.4.2 Génération de collections d'arbres

Pour les méthodes bayésiennes, la validation est inhérente à l'approche proposée. En effet, ces méthodes génèrent une collection d'arbres dont l'information phylogénétique peut être résumée en calculant le consensus majoritaire. La fréquence avec laquelle les différents nœuds apparaissent dans les arbres visités représente leur probabilité a posteriori associée. Par rapport aux pourcentages de bootstrap (section précédente), les probabilités a posteriori bayésiennes présentent l'avantage d'être facilement interprétables statistiquement. Elles représentent en effet la probabilité qu'un clade donné soit vrai étant donné le modèle d'évolution, les probabilités a priori et les données considérées [Huelsenbeck et al. 2001].

Or, pour obtenir une collection d'arbres *intéressants*, c'est-à-dire avec de l'information phylogénétique, il faut que les échantillons tirés soient représentatifs, autrement dit, que les

chaînes MCMC convergent vers un état stationnaire. Pour qu'une chaîne de Markov puisse converger vers un état stationnaire, elle doit être irréductible (ce qui signifie que tout état est accessible à partir de n'importe quel autre état) et apériodique (ou acyclique, c'est-à-dire qu'elle ne doit pas boucler sur elle-même).

Concrètement, cela signifie que dans un paysage multidimensionnel aussi complexe que celui associé aux problèmes phylogénétiques, les minimums locaux peuvent être potentiellement nombreux. Le but est d'éviter d'être piégé par ces minimums locaux. Pour ce faire, une variante de l'algorithme MH utilise un couplage de Metropolis des MCMC (MCMCMC ou MC^3 pour *Metropolis Coupling Markov Chain Monte Carlo*). Cet algorithme permet d'utiliser n MCMC simultanément, dont $n-1$ sont dites *chauffées* de manière graduelle. Ces chaînes *chaudes* – utilisant un pas plus large – permettant une exploration plus vaste de l'espace des paramètres – sont utilisées pour guider la chaîne dite *froide* à partir de laquelle les inférences sont faites. L'utilisation conjuguée des deux types de chaînes permet de sortir d'un minimum local éventuel. L'utilisation de MC^3 contribue ainsi à la réduction des risques de défaut de convergence des MCMC [Huelsenbeck & Ronquist 2001]. Le programme MrBayes implémente une version de cet algorithme.

Parmi la collection d'arbres générés durant l'inférence bayésienne, les derniers arbres les plus stables ont été considérés dans le calcul de l'arbre consensus par le programme MrBayes avec l'option Consensus majoritaire. Le calcul de l'arbre consensus a été effectué à la fois, pour l'arbre d'espèces (voir la figure $3.5g_1$ de la section 3.6.1, page 63) et tous les arbres de gène (voir la figure $3.5g_2$ de la section 3.6.1, page 63) respectivement à partir des collections d'arbres d'espèces et des collections des arbres de gène. De même, ces collections d'arbres ont été préalablement générées à la fois, à partir de la matrice binaire (voir la figure $3.5d_1$ de la section 3.6.1, page 63) et à partir des alignements de séquences de VOG (voir la figure $3.5d_2$ de la section 3.6.1, page 63).

3.6.4.3 Autres sources de validation

En complément aux différentes techniques statistiques de validation discutées, il est suggéré d'utiliser également d'autres sources telles que les classifications virales existantes basées sur d'autres critères (e.g., morphologie) et les informations diverses issues de la littérature

(fonction du gène, existence de gènes *paralogues**, etc.). La confrontation avec ces autres sources d'informations permet de valider les résultats de reconstruction d'arbre obtenus.

3.7 DÉFIS POUR LA PHYLOGÉNIE MOLÉCULAIRE

Nous mettons ici en perspective les différents défis méthodologiques auxquels l'analyse phylogénétique contemporaine doit faire face.

Voici quelques-uns des principaux défis méthodologiques actuels : [1] Reconstruire l'arbre phylogénétique qui représente le mieux possible l'histoire de l'évolution des espèces. La représentation traditionnelle se fait sous forme d'arbre binaire, mais les récents travaux en biologie tendent à démontrer, du moins pour les phages, qu'une topologie en réseaux serait plus appropriée [Liu et al. 2006] ; [2] Estimer correctement les longueurs de branches notamment en fonction du taux d'évolution ou des dates de divergence [Lerat et al. 2003 ; Douzery et al. 2004] ; [3] Caractériser les processus complexes d'évolution, à savoir [3.i] Etablir les écarts entre l'évolution et l'horloge moléculaire [Lepage et al. 2006 ; Douzery et al. 2004], [3.ii] Déterminer la variation du taux d'évolution dans un arbre phylogénétique grâce aux modèles covariation [Galtier 2001], [3.iii] Décrire la coévolution entre site [Maddison 1997 ; Felsenstein 2006], [3.iv] Détecter les transferts horizontaux de gènes [Makarenkov et al. 2008] ; (4) Reconstruire les séquences ancestrales [Blanchette et al. 2007; Diallo et al 2006].

Les points [1], [3.iv] et [4], autrement dit, la reconstruction d'arbres et de réseaux d'espèces, les transferts horizontaux de gènes ainsi que la reconstruction de séquences ancestrales ont été traités dans cette étude sur la classification des bactériophages. Ces points vont faire l'objet de nos prochaines discussions dans les chapitres à venir.

CHAPITRE IV

—

APPROCHE ORIGINALE DE CLASSIFICATION DES BACTÉRIOPHAGES[19]

Plan du chapitre

[19] Ce chapitre a fait l'objet de deux articles de publication [Nguyen et al. 2007a ; Nguyen et al. 2007b]. Voir aussi Annexes F.2 et F.3.

Les bactériophages sont des virus qui infectent les organismes des domaines *Bacteria* et les *Archaea*. Leur évolution est complexe à cause des mécanismes d'évolution réticulée comprenant le transfert horizontal de gènes et la recombinaison génétique. Une représentation phylogénétique sous forme de réseau est nécessaire pour interpréter l'histoire de l'évolution des phages [Liu et al. 2006].

Par ailleurs, la classification de ces microorganismes présente intrinsèquement d'autres difficultés dues, d'une part, à la non-conservation de gènes au cours de l'évolution [Rohwer & Edwards 2002], et d'autre part, à la diversité des tailles de leurs génomes [Liu et al. 2006]. Il existe plusieurs classifications des bactériophages [Jarvis et al. 1991 ; Büchen-Osmond 2003 ; Maniloff & Ackermann 1998]. L'approche de classification développée pour les phages, adoptée au cours des dernières décennies, est basée sur les critères de morphologie et d'homologie des ADN. Par exemple, la grande majorité des phages *Lactococcal lactis* affectant des bactéries du lait, a été classée en trois principaux groupes : *phage936*, *c2* et *P335* [Deveau et al. 2006]. Dès lors, la plupart des études des phages tiennent compte de l'existence de ces groupes. Or, plusieurs travaux récents sur l'analyse comparative des phages semblent démontrer des incohérences dans les regroupements [Spinelli et al. 2005, 2006 ; Ricagno et al. 2006] et suggèrent de ce fait une révision du mode de classification actuelle [Deveau et al. 2006].

Dans cette étude, nous proposons une approche originale en trois principales étapes visant à établir la classification des phages : l'inférence phylogénétique, la détection de transferts horizontaux de gènes [Makarenkov et al. 2008] et la reconstruction de séquences protéiques ancestrales [Diallo et al. 2006 ; Blanchette et al. 2007].

Le présent chapitre est consacré à l'étude de la classification des bactériophages. Nous aborderons les différents aspects ayant traits à la problématique de classification des microorganismes. La discussion portera d'abord sur l'évolution des bactériophages et la reconstruction d'arbres phylogénétiques d'espèces et de gènes avec une méthode de distances (e.g., Neighbor-Joining [Saitou & Nei 1987 ; Studier et Keppler 1988]) et la méthode concurrente bayésienne (e.g., MrBayes [Huelsenbeck & Ronquist 2001]), suivi ensuite, de la détection des transferts horizontaux de gènes grâce à l'approche d'optimisation de distances

topologiques de Ronbinson et Foulds (e.g, HGT-Detection [Makarenkov et al. 2008]), et enfin, de la reconstruction de séquences de protéines ancestrales avec la méthode du maximum de vraisemblance Tree-HMM (e.g, Ancestor [Diallo et al. 2006]).

Le premier et le troisième point sont soumis à l'hypothèse de l'évolution classique (i.e., par héritage vertical d'un ancêtre commun). Tandis que le deuxième point est soumis à l'hypothèse de l'évolution réticulée caractérisée notamment par les transferts horizontaux de gènes.

4.1 EVOLUTION DES BACTÉRIOPHAGES

Cette première partie présente les deux approches traditionnelles de l'analyse phylogénétique : la phénétique et la cladistique. La discussion porte également sur les représentations sous formes d'arbres et de réseaux dans le cadre d'évolution réticulée. Pour terminer, l'accent et mis sur un cas particulier, les bactériophages dsDNA, le lien avec la reconstruction ancestrale et la façon dont nous obtenons les groupes VOG de séquences de génomes viraux.

4.1.1 PHYLOGENÈSÉ ET REPRÉSENTATIONS

La *phylogenèse* s'intéresse à la reconstruction de l'histoire de l'évolution des êtres vivants. La phylogénie moléculaire (i.e., arbre phylogénétique moléculaire) procède par comparaison de gènes d'ADN ou de protéines. Les données analysées consistent généralement en un ensemble d'organismes (*taxons**), et pour chaque organisme en un ensemble de données moléculaires (e.g., les séquences). Pour reconstituer les liens de parenté entre êtres vivants, l'analyse phylogénétique procède selon deux techniques : la cladistique et la phénétique. Les relations de parentés entre des entités supposées avoir un ancêtre commun peuvent être représentées sous formes d'arbres et de réseaux.

4.1.1.1 Cladistique et phénétique

La cladistique

En cladistique, on ne regroupe dans un taxon que les êtres vivants qui partagent des caractères *homologues* : lorsqu'une ressemblance entre deux taxons peut être attribuée à une ascendance commune, on parle d'homologie [source NCBI, voir lien en Annexe B]. La

cladistique repose donc sur l'identification (souvent difficile) de l'homologie des caractères. Elle est pertinente au niveau morphologique (et est donc le seul moyen de classer les espèces fossiles dont l'ADN est rarement conservé) comme au niveau moléculaire. L'homologie est la relation de deux caractères (morphologiques ou moléculaires) qui sont des descendants, le plus souvent par divergence, d'un caractère d'un ancêtre commun [Fitch 2000]. Les caractères *orthologues** et *paralogues* sont des caractères homologues qui sont produits respectivement par spéciation de deux lignées, et par duplication de gènes dans l'une ou les deux lignées. Les caractères xénologues sont deux caractères orthologues tels que l'un ou les deux d'entre eux ont été transférés horizontalement (section 4.1.2) [Liu et al. 2006].

La phénétique

La phénétique repose sur le postulat de base que le degré de ressemblance est corrélé au degré de parenté. Elle suppose donc de quantifier la ressemblance entre les êtres vivants à classer. Cette méthode se révèle peu pertinente lorsqu'on l'applique aux caractères morphologiques en raison des *analogies** (e.g., les ailes d'Oiseau et de Chauve-souris présentent des caractères analogues mais ces espèces n'ont pas un proche ancêtre commun). En revanche, la phénétique devient pertinente dès lors que l'on compare un très grand nombre (au sens statistique) de caractères car le nombre de caractères analogues devient négligeable parmi tous les caractères dont la ressemblance est effectivement due à la parenté. Jusqu'ici, l'expérience nous a enseigné que les organismes étroitement liés ont des séquences semblables (ou similitude), et inversement, les organismes plus éloignés ont des séquences différentes (ou dissimilitude). Par conséquent, les protéines ayant une conservation significative de caractères présentent une relation de parenté et sont qualifiées comme faisant partie de la même famille de protéines [source NCBI, voir lien en Annexe B].

Notons l'analogie du concept de similitude/dissimilitude de séquences avec le même concept discuté au chapitre II, sur la catégorisation humaine, en particulier sur la théorie des prototypes (section 2.1.2.1).

L'utilisation conjointe de la cladistique et la phénétique

Longtemps opposées, la cladistique et la phénétique sont aujourd'hui souvent utilisées conjointement comme étant deux méthodes indépendantes. L'utilisation conjointe de ces deux

approches ainsi que la confrontation des arbres obtenus a révélé l'existence dans la classification traditionnelle (i.e., la classification acceptée jusqu'à récemment qui s'oppose à la nouvelle classification phylogénétique basée sur les séquences biologiques) de nombreux groupes non fondés sur les liens de parenté, et qui sont donc considérés comme non légitimes, et ne devant plus être utilisés en taxonomie. A titre d'exemple, grâce à l'utilisation conjointe des deux méthodes, des changements ont été apportés dans l'arbre phylogénétique au niveau du groupe des reptiles. En effet, au sein de ce groupe étaient regroupés aussi bien les crocodiliens ou *Crocodilia* (en fait génétiquement proches des oiseaux) que les lézards, les serpents et les tortues (qui sont éloignés génétiquement des oiseaux).

4.1.1.2 Représentations phylogénétiques

La similitude moléculaire des organismes étudiés suggère fortement que tous les organismes sur terre aient un ancêtre commun[20] [Mayr 1997]. Par conséquent, n'importe quel ensemble d'espèces est relié entre eux, et cette relation est appelée phylogénie. Les relations d'évolution sont normalement représentées par des arbres phylogénétiques. Mais, depuis quelques années, l'émergence des modèles en réseau viennent complémenter les modèles en arbre pour expliquer certains mécanismes d'évolutions réticulées (voir la section qui suit). En effet, selon McDade [1995], les outils d'analyse qui permettent de générer des topologies réticulées montrent de manière plus précise l'histoire hybride des organismes. Tandis que les méthodes traditionnelles appliquées à de tels cas peuvent produire des confusions dans les résultats puisqu'elles sont contraintes de générer seulement des patrons de type arbre. La tâche des phylogénéticiens consiste donc à inférer des arbres/réseaux à partir des observations des organismes existants.

Les arbres/réseaux phylogénétiques peuvent être utilisés notamment pour trouver des caractères orthologues et paralogues, prédire la structure secondaire des RNA, étudier les relations hôte et parasite, aligner les séquences multiples.

[20] Ceci semble être en contradiction avec les transferts horizontaux de gènes (section 4.1.2 suivante). Nous verrons par la suite du document que l'ascendance ancestrale et les transferts horizontaux de gènes se complète dans l'évolution des bactériophages.

4.1.2 EVOLUTION RÉTICULÉE ET TRANSFERTS HORIZONTAUX DE GÈNES

L'évolution des êtres vivants a longtemps été modélisée uniquement à l'aide des arbres phylogénétiques. Dans un arbre phylogénétique, deux espèces sont toujours reliées par un chemin passant par leur ancêtre commun. Un tel modèle ne peut inclure des scénarios d'*évolution réticulée*. La recombinaison homologue, l'hybridation, le transfert horizontal de gènes, la duplication d'un gène suivie de sa perte et l'évolution convergente sont les principaux mécanismes d'évolution réticulée [Legendre & Makarenkov 2002] (figure 4.1).

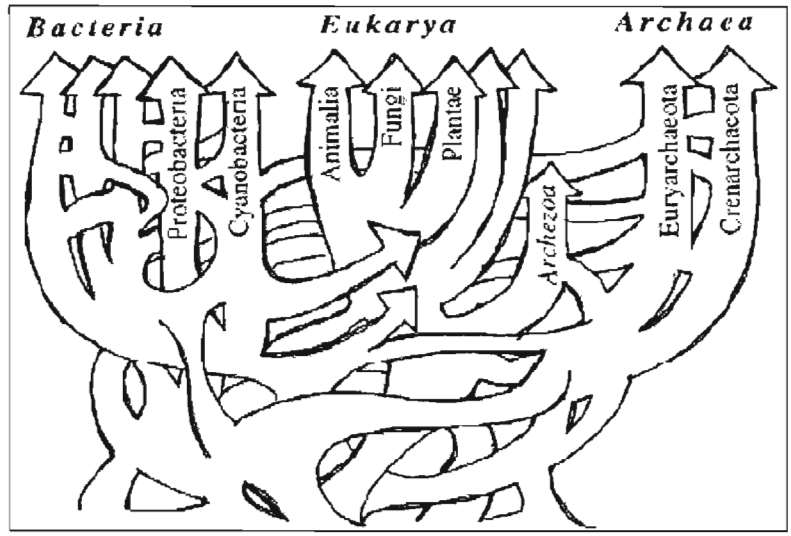


Figure 4.1 : Un arbre réticulé ou un réseau d'espèces
(Tirée de Doolittle 1999)

Le transfert horizontal consiste en un échange direct de matériel génétique d'une lignée à une autre [Doolittle 1999]. Les bactéries et les Archées ont développé des mécanismes sophistiqués pour acquérir rapidement de nouveaux gènes à l'aide du transfert horizontal. Ces mécanismes ont été favorisés par la sélection naturelle par rapport à l'évolution génétique par mutations. Les trois principaux mécanismes de transfert de gènes sont (figure 4.2) : la *transformation* par acquisition d'ADN directement de l'environnement, la *conjugaison* qui est enclenchée par des plasmides conjugaux ou par des transposons conjugaux, et la *transduction* par transfert d'ADN par bactériophages. Ces mécanismes peuvent introduire des

séquences d'ADN de l'espèce donneur ayant très peu de similitude avec le reste de l'ADN de l'espèce hôte.

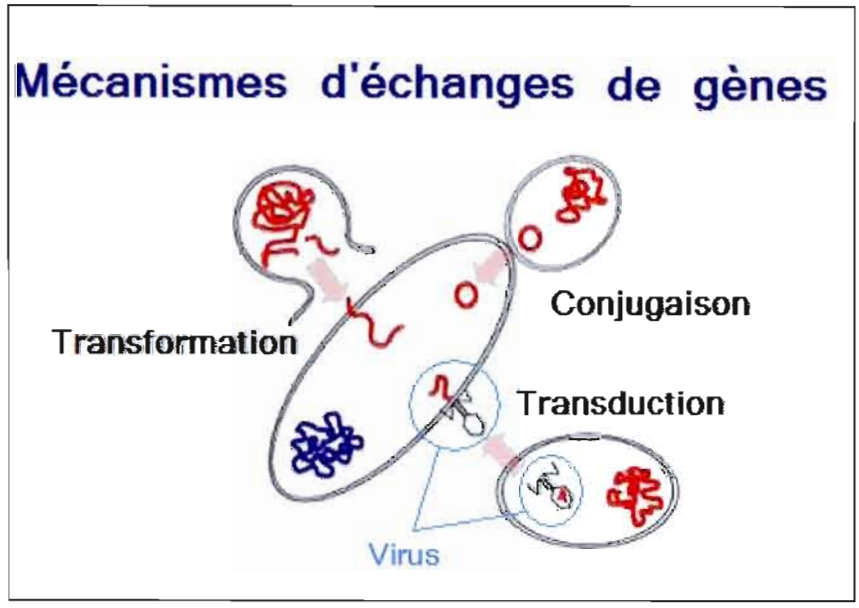


Figure 4.2 : Transfert horizontal de gènes par transformation, conjugaison et transduction
(Adaptée de la figure originale localisée sur le site *http://www.pitt.edu/~heh1/Research.html*)

De plus, on ne peut pas exclure des phénomènes d'homoplasie (i.e., similarités au niveau de la morphologie, des séquences d'ADN ou de protéines qui ne sont pas dues à l'existence d'un ancêtre commun assez proche) dont l'analogie et l'évolution convergente. Ces phénomènes constituent une autre cause de transferts de gènes possible, et donc de confusion supplémentaire pour son interprétation.

4.1.3 CAS D'ÉTUDE : LES BACTÉRIOPHAGES DSDNA

Les bactériophages, en particulier les *double-stranded DNA* (*dsDNA* ou d'ADN à double brin), sont des virus qui infectent les bactéries et les Archées, et sont une des causes de mutations génétiques de ces derniers (figure 4.2). Depuis le séquençage récent d'un grand nombre de ces microorganismes, les chercheurs ont trouvé que les phages sont extrêmement

abondants, avec une concentration typique estimée autour de 10 millions de particules par ml dans l'eau de mer côtière et davantage encore dans d'autres milieux tels que les étangs d'eau douce, soit au moins 10 fois plus que des organismes cellulaires [Bergh et al. 1989 ; Wommack & Colwell 2000]. Suivant ces estimations, la population globale des phages pourrait être de l'ordre de 10^{31}, et par conséquent, cela suggère que ces microorganismes soient la forme de vie la plus abondante sur terre [Hendrix 2002]. De multiple possibilité d'échange de gènes par des recombinaisons non homologues et des réarrangements de séquences à travers les recombinaisons homologues soumettent les phages à une évolution réticulée [Legendre & Makarenkov 2002].

Dans ce contexte, il est très difficile d'étudier l'évolution des phages en comparant uniquement leurs protéines (par homologie et/ou mesures de distances entre séquences) : on préfère ainsi effectuer des comparaisons des groupes de plusieurs gènes ou bien carrément des génomes entiers d'une espèce par rapport à une autre (section 4.1.6) [Receveur-Bréchot & Grob 2006]. En termes de modélisation, le modèle en arbre traditionnel n'est donc pas suffisant pour rendre compte de telle évolution. Il est nécessaire d'utiliser des modèles en réseaux [Liu et al. 2006].

Finalement, les phages sont considérés comme des agents intéressants dans l'étude de transferts horizontaux de gènes entre les microorganismes [Canchaya et al. 2003]. De plus, du fait de leur modèle relativement plus simple en comparaison aux modèles des eucaryotes par exemple, les phages constituent un cas d'étude de choix.

4.1.4 TAXONOMIE EXISTANTE DES BACTÉRIOPHAGES

Il existe plusieurs classifications des virus [Jarvis et al. 1991, Maniloff & Ackermann 1998, Büchen-Osmond 2003]. Par exemple, le comité international de taxonomie des virus ICTV (*International Committee on the Taxonomy of Viruses*) [Büchen-Osmond 2003], organisme de référence sur la taxonomie virale, reconnaît plusieurs groupes de phages, principalement sur la base des propriétés morphologiques partagées (tableau 4.1 de la section 4.1.6.2) telles que la forme, l'hôte infecté, la structure du virion libre, ainsi que des considérations sur la taille des génomes et les propriétés moléculaires comme l'homologie des ADN.

En plus d'être abondants dans la biosphère, les bactériophages jouent aussi d'importants rôles dans la pathogénie de bactéries [Wagner & Waldor, 2002] et dans la dynamique des génomes hôtes [Hendrix et al. 1999].

Ainsi, la difficulté spécifique due à la diversité du mode d'évolution réticulée et à la complexité de l'écosystème des sujets est telle qu'une classification exhaustive et de consensus n'est pas encore disponible. Par exemple, suivant les classifications déjà établies, la majorité des phages infectant les bactéries *Lactococcal lactis* (ou *L. Lactis*) appartiendraient à l'un des trois principaux regroupements : *phage936*, *c2* et *P335* [Deveau et al. 2006]. Or, plusieurs travaux récents sur l'analyse comparative d'un nombre croissant de séquences génomiques et de l'émergence récurrente de nouveaux phages virulents semblent démontrer des incohérences dans les regroupements [Spinelli et al. 2005, 2006 ; Ricagno et al. 2006 ; Deveau et al. 2006 ; Lawrence et al. 2002].

Notamment, l'équipe de Spinelli et al. [2005, 2006] a montré une certaine similitude structurale au niveau de la protéine *Receptor Binding Protein* (RBP) des phages *L. lactis*, avec d'autres types de virus comme les *adénovirus*[21] et *réovirus*[22] qui infectent les cellules des mammifères dont celles de l'homme (lire par ailleurs, la section 5.3.2.1., "*Protéine RBP des phages L. Lactis*"). Ces résultats semblent donc montrer que les virus éloignés sur le plan évolutif pourraient avoir des gènes ancestraux communs malgré le manque de proximité dans leurs séquences respectives [Ricagno et al. 2006]. En d'autres termes, les protéines responsables d'une même fonction dans différents organismes peuvent soit provenir d'un ancêtre commun ou d'une acquisition de gènes indépendante et spécifique à chaque lignée d'espèces. Par conséquent, les classifications des phages devraient également tenir compte des processus d'apparition des fonctions protéiques et l'évolution réticulée de certains gènes. Deveau et al. [2006] par exemple, suggèrent de réviser, du moins pour les phages affectant les bactéries *L. Lactis*, le mode de classification existante.

---

[21] Des adénovirus qui regroupent une centaine de variétés, dont une quarantaine environ peuvent infecter l'homme.

[22] La famille des reoviridae comprend 11 genres et inclut certains virus affectant entre autres le système digestif (tel que le *rotavirus*), ou le système respiratoire. Ce sont des virus qui infectent les invertébrés, les plantes et les vertébrés.

4.1.5 RECONSTRUCTION ANCESTRALE

De manière générale, la reconstruction ancestrale permet d'étudier l'évolution des espèces, la sélection adaptative et la divergence fonctionnelle [e.g., Krishnan et al. 2004]. La reconstruction ancestrale tient compte de la recréation de protéines et de l'évolution d'ADN en laboratoire de sorte qu'elles puissent être étudiées directement [Chang et al. 2005]. Dans le cas des protéines par exemple, on tient compte de la recherche sur l'évolution actuelle de la structure et de la fonction moléculaire. De plus, la reconstruction ancestrale de protéines peut mener aux découvertes de nouvelles fonctions biochimiques qui ont été perdues au cours de l'évolution, c'est-à-dire, absentes dans les protéines modernes [Jermann et al. 1995].

Les séquences de protéines renferment des informations sur leurs passés historiques [Pauling & Zuckerkandl 1963]. La similitude entre les séquences appartenant à une même famille de protéines peut être utilisée pour construire l'arbre d'évolution qui montre leurs degrés de parenté [Benner 2002]. Les séquences ancestrales peuvent être reconstruites en à partir des séquences observées existantes. Des dates peuvent être ainsi placées suivant les événements passés le long de l'histoire moléculaire. Et, ces mêmes événements peuvent être corrélés avec les événements produits au cours de l'histoire géologique et paléontologique. De cette corrélation, enfin, on peut voir émerger une stratégie pour interpréter la protéomique. Autrement dit, si nous parvenions à comprendre le passé d'une protéine, nous pourrions mieux comprendre son évolution à nos jours [Benner 2001].

Dans cette étude, nous nous sommes intéressés en particulier aux fonctions protéiques et à la recherche de séquences ancestrales, appliquées plus spécifiquement aux séquences de protéines ancestrales de bactériophages (section 4.4).

4.1.6 VOG – VIRAL ORTHOLOG GROUP

Du fait que les bactériophages présentent une double caractéristique due à la variabilité de composition génétique et de tailles des génomes, nous avons choisi de considérer un ensemble de regroupements VOG (i.e., regroupement de gènes viraux orthologues, voir les détails dans les sections à venir) prédéfinis comme données initiales dans cette étude phylogénétique.

4.1.6.1 Variabilités de composition génétique et de tailles des génomes

L'étude phylogénétique des bactériophages présente une double difficulté en raison de la grande variabilité à la fois dans la composition génétique et dans les tailles des génomes [Liu et al. 2006].

La première difficulté découle de la divergence totale des séquences protéiques [Rohwer & Edwards 2002]. La très grande variabilité dans la composition génétique des phages résulte de la profusion des échanges et le réassortiment de gènes par des recombinaisons homologues et illégitimes (non homologues). Ainsi, lorsque l'on compare les génomes d'une collection de virus phylogénétiquement proches, on observe des réassortiments de groupes de gènes (ou modules), si bien que les génomes comparés apparaissent comme des mosaïques de modules les uns avec les autres. La figure 4.3 montre un exemple de variabilité de composition génétique des phages *Mu* et *SfV* (famille *Myoviridae*) et des phages *HK97*, *Lambda* et *N15* (famille *Siphoviridae*), mettant en perspective les modules homologues appartenant à des phages différents.

Le lecteur est renvoyé à la section 5.1.2. "*Représentation de l'arbre d'espèces*" (figure 5.3), où les regroupements trouvés par la présente étude représentent les phages *Mu* et *SfV* dans le groupe 19, et les phages *HK97*, Lambda et *N15* dans le groupe 17.

Par conséquent, à cause de cette variabilité génétique extrême, l'étude phylogénétique des phages n'est pas chose aisée. En effet, des marqueurs moléculaires qui fournissent habituellement un cadre pour la compréhension de la phylogénie microbienne tels que le *RNA ribosomial* et l'ensemble des protéines conservées de manière universelle [Wolf et al. 2002] ne sont pas applicables dans l'analyse d'évolution des phages à cause de l'absence de gènes uniques partagés par toutes les espèces [Rohwer & Edwards 2002 ; Liu et al. 2006].

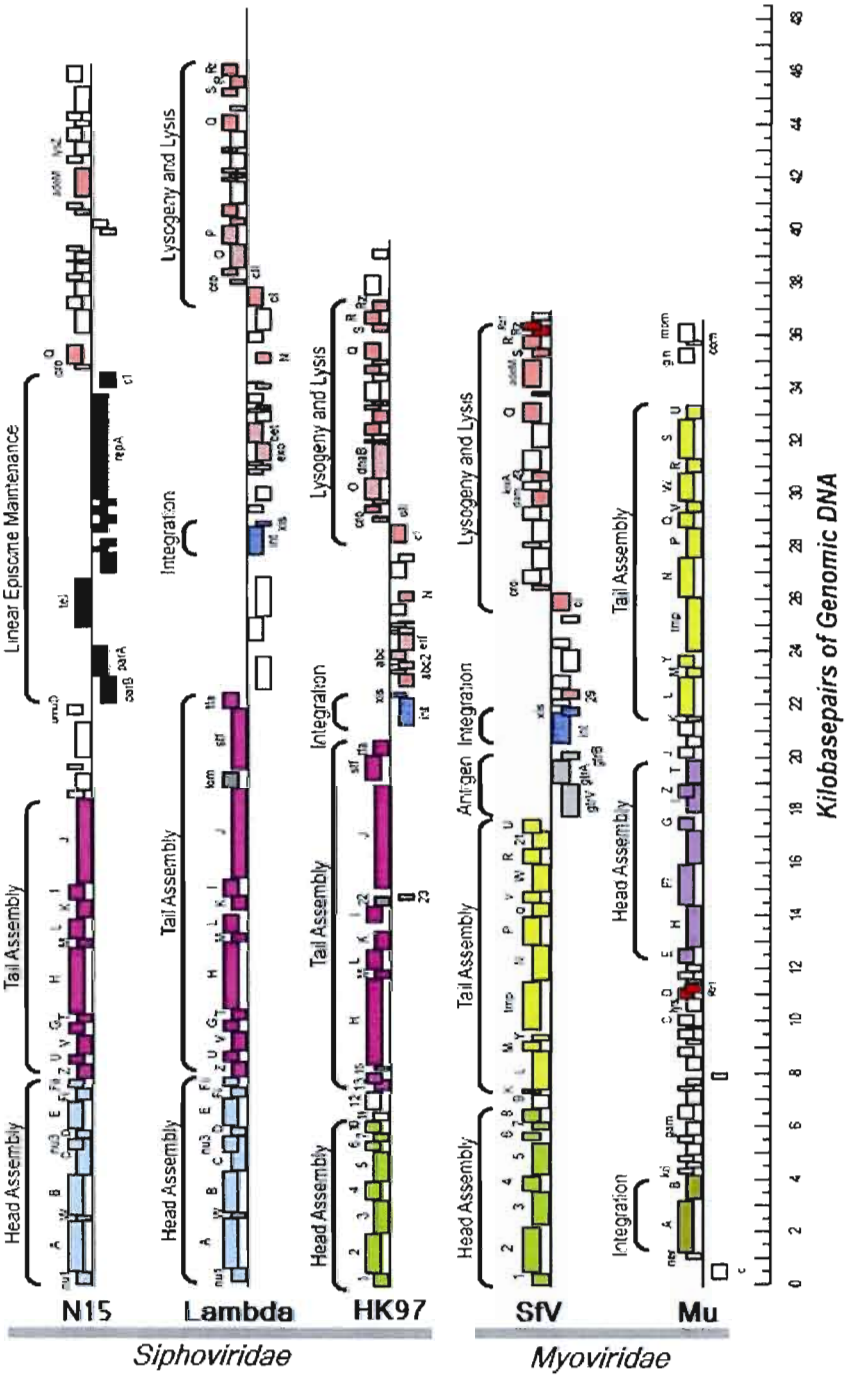


Figure 4.3 : Variabilité de compositions génétiques des phages *Siphoviridae* et *Myoviridae*.

Les modules homologues appartenant à des phages différents sont indiqués avec la même couleur (tirée de Lawrence et al. 2002).

La seconde difficulté est due aux tailles très disparates des génomes (figure 4.4), qui sont de magnitude d'ordre 2 (le nombre de gènes codant en protéines varie de 8 (phage *LoL5*) à 381 (phage *nt1*), voir tableau C.3 en Annexe C), en comparaison aux procaryotes (de ~400 à ~7 000 gènes) et aux eucaryotes (de ~4 000 à ~60 000 gènes), qui sont de magnitude d'ordre 1 [Liu et al. 2006].

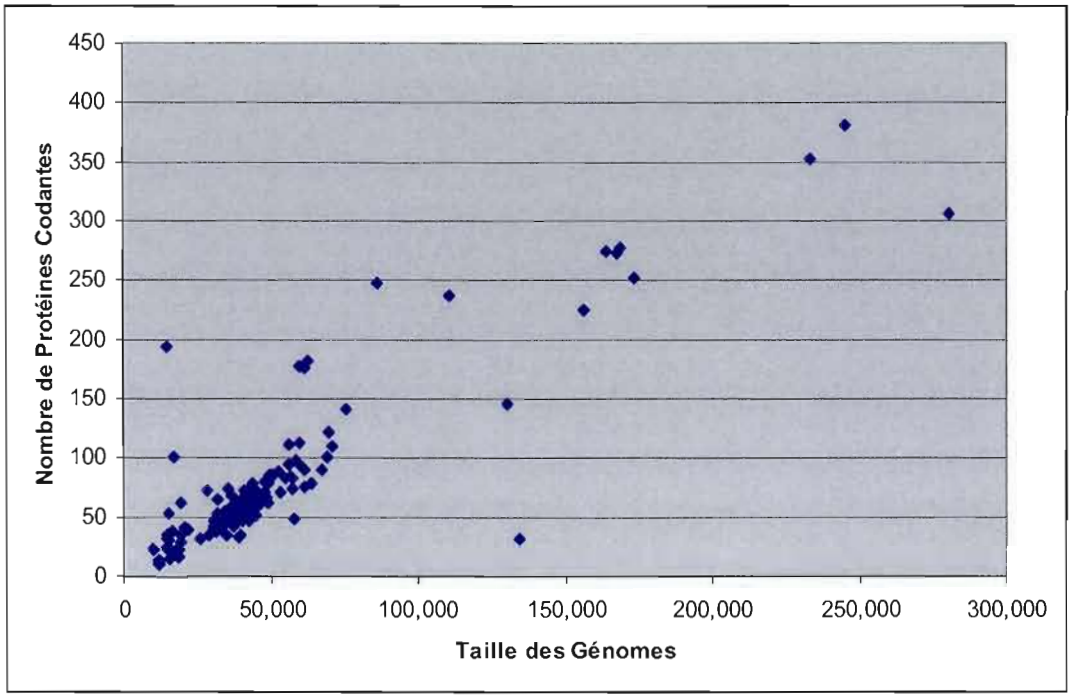


Figure 4.4 : Variabilité de tailles des génomes

Plus la taille du génome (nucléotides) est importante plus il y a de gènes codants (Les informations sur la taille des génomes et le nombre de gènes codants sont données en Annexe C).

Une alternative à la conservation de gènes uniques consiste à étudier l'évolution des phages basée sur le contenu en gènes, notamment sur l'information que l'on retirait des cas de gènes orthologues partagés par différents génomes. Il a été prouvé dans le cas des bactéries, que non seulement les arbres basés sur le contenu en gènes tendent à être similaires au consensus courant de l'arbre de vie [Wolf et al. 2002], mais également cela permet de mettre en lumière des événements évolutifs rares [Snel et al. 2005], voire de l'existence possible de nouveaux clades de bactéries [Wolf et al. 2001]. Cependant, les arbres basés sur

le contenu en gènes ont leurs propres limites. La plus notable est certainement leur grande sensibilité aux méthodes utilisant les mesures de distances entre génomes. En particulier, on peut, à moins que les distances soient correctement normalisées (section 4.2.2.1), voir les bactéries de petits génomes et plusieurs génomes séquencés de parasites humains placés ensemble dans des clades artificiels. La correction de distances (ou normalisation par la taille des génomes) réduit l'ampleur du phénomène *attraction de branche-courtes* [Korbel et al. 2002], et replace par conséquent les parasites près de leurs espèces, conformément au degré de similitude entre les séquences de gènes orthologues pris individuellement [Wolf et al. 2001, 2002]. La meilleure façon de normaliser les distances entre les génomes reste toutefois un débat ouvert [Mirkin & Koonin 2003].

Dans l'état actuel des connaissances, les deux approches d'inférence phylogénétique des phages seraient de combiner l'aspect informationnel basé sur le contenu en gènes avec correction de distances en normalisant par la taille, et l'analyse des alignements des séquences de protéines distribuées dans les génomes de plusieurs phages [Liu et al. 2006].

Grâce aux données VOG extraites de NCBI (section suivante), les regroupements des protéines orthologues apportent des données informationnelles (i.e., contenu en gènes) nécessaires pour résoudre une partie de la première problématique (i.e., la variabilité de composition génétique). L'autre partie consiste à normaliser l'hétérogénéité des tailles des génomes afin de corriger les distances entre les espèces (sections 4.2.2.1 et 4.2.2.2). La résolution de la seconde problématique consiste à aligner les séquences de protéines distribuées dans les VOG, et inférer l'arbre de gène relatif à chacun des VOG. Cette méthode d'alignement et d'inférence de gènes sera présentée plus loin à la section 4.2.3.

4.1.6.2 VOG considérés pour l'étude

La banque de données GenBank, hébergée sur le site du *National Center for Biotehcnology Information* (NCBI) dispose d'une base de données de groupements relatifs aux protéines virales. Cette ressource, nommée *Viral COG – Clusters of Orthologous Groups* (VOG) [Bao et al. 2004], fournit des molécules standard pour la recherche génomique virale. Les données disponibles proviennent de génomes complets présents dans GenBank. Les données VOG sont des séquences de protéines regroupées (*clusters*) de manière prédéfinie en famille selon

la fonction protéique à laquelle elles sont associées. Un VOG comprend plusieurs séquences de gènes (au moins 4 dans notre cas) et autant d'espèces différentes afin de représenter un contenu informationnel (*phageness* [Liu et al. 2006]) suffisamment intéressante pour l'étude. Le contenu informationnel des VOG est utilisé dans les études dont le but est d'améliorer l'annotation fonctionnelle des nouvelles protéines.

La façon de les regrouper est en soi une problématique non triviale, et reste un sujet de recherche à part entière. Sans rentrer dans le détail[23], voici les grandes lignes [Liu et al. 2006] : [1] Prédiction de gènes : à partir des génomes complets de phages extraits de GenBank/NCBI, l'utilisation des algorithmes GeneMarkHMM, GeneMarkS, Violin[24] (référence de liens URL en Annexe B) [Besemer & Borodovsky 1999 ; Besemer et al. 2001] permet de prédire les *Open Reading Frame* (ORF), soit les marqueurs de gènes. Pour la même tâche, NCBI utilise plutôt la famille des algorithmes PSI-BLAST [Altschul et al. 1997, Schaffer et al. 2001] ; [2] Recherche de gènes orthologues (i.e COG) : l'utilisation des algorithmes de la famille COG dont Cognitor [Tatusov 1997] (référence de lien URL en Annexe B), permet de catégoriser en groupes de gènes orthologues. Les regroupements COG sont adaptés pour les organismes tels que les eucaryotes, les unicellulaires, etc. mais non pour les virus ; [3] Regroupement en clusters de gènes viraux (i.e., VOG) : les protéines virales sont comparées suivant leur profil (*profile*) via les algorithmes BLASTP/PSI-BLAST, à d'autres protéines des organismes non viraux qui sont déjà annotées (i.e., les protéines qui sont connues et décrites) dans les banques de données GenBank (Etats-Unis), EMBL (Europe) et DDJB (Japon). Les algorithmes BLASTP/PSI-BLAST permettent de regrouper en familles de protéines virales suivant les différents critères fonctionnels dont la conservation de domaine CDD (*Conservation Domain Database*) [Soding 2004].

Nous avons recensé en juillet 2006 sur le site NCBI (référence de liens URL en Annexe B), tout en s'assurant de la validité des références sur ICTV (référence de liens URL en Annexe B), 163[25] génomes complets de bactériophages dsDNA issus de 8 familles

---

[23] Pour les intéressés, Bao et al. [2004] sont une bonne référence comme point d'entrée

[24] VIOLIN est spécialement conçu pour prédire les séquences de virus.

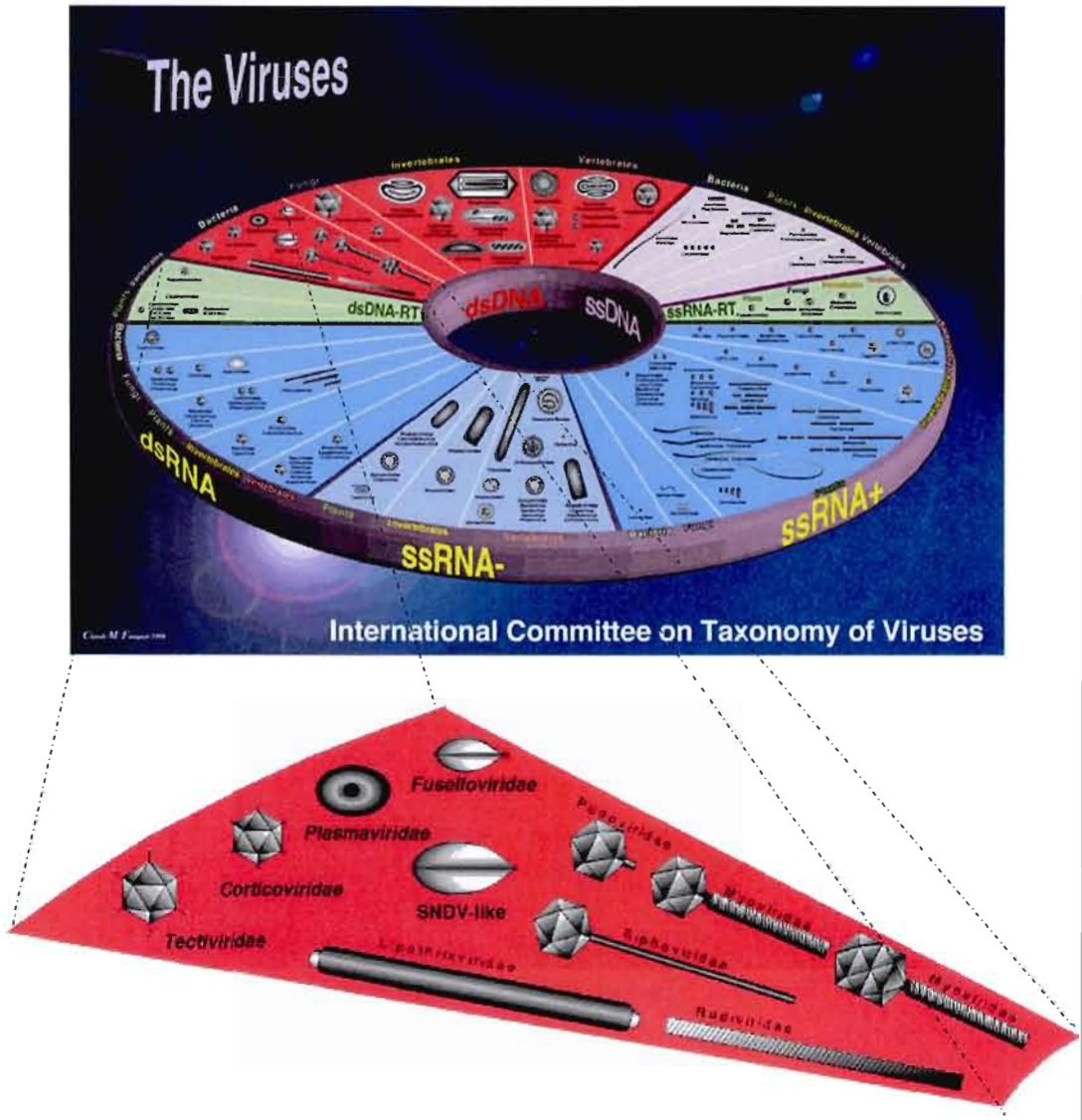
[25] En fait, NCBI en dénombre exactement 482 génomes complets de bactériophages dsDNA en date de juillet 2006, mais de ce nombre la plupart n'est pas encore reconnu (du moins, pas encore complètement annoté) par ICTV.

différentes et un ensemble de phages avec des annotations partielles (*unclassified*). Les séquences de protéines de 163 génomes distribuées dans 602[26] regroupements VOG ont été extraites et considérées dans la présente étude (tableau 4.1.).

[26] NCBI en dénombre exactement 1007 regroupements VOG. De ce nombre, seulement 602 VOG ayant plus de trois espèces différentes regroupées sont phylogénétiquement intéressants à étudier.

Par ailleurs, parmi les 602 VOG, certains ont plus d'un gène d'une espèce donnée dans le regroupement. Ces derniers ne sont pas consistants à étudier due à l'approche retenue de normalisation (voir section 4.2.2). Pour choisir le « meilleur » gène parmi les différents gènes de l'espèce dans le VOG, nous avons effectué d'abord toutes les combinaisons possibles comprenant un des gènes de l'espèces en question avec les gènes des autres espèces, puis pour chaque combinaison, nous avons effectué des alignements de séquences avec le programme ClustalW [Thompson et al. 1994] lequel nous donne un score d'alignement. Nous avons finalement choisi la combinaison qui a le plus haut score.

Bactériophages dsDNA



Source : Virosphere 2000 de ICTV, http://www.virustaxonomyonline.com/virtax/lpext.dll?f=templates&fn=main-h.htm.

Les génomes viraux peuvent être circulaires ou linéaires, et être composés d'ADN simple (ssDNA) ou double brin (dsDNA ou dsDNA-RT, i.e., à transcriptase inverse), ou d'ARN simple (ssRNA-/+, i.e., ssRNA antisens (3' à 5') / sens (5' à 3')) ou double brin (dsRNA).

A noter que certains virus ssARN *satellites* possèdent des génomes d'à peine quelques centaines de paires de bases (pas plus de 0.22 kb pour le *Rice yellow mottle* virus [Collins et al. 1998]), tandis que certains virus géants dsDNA possèdent des génomes de plusieurs centaines de kilobases (jusqu'à 800 kb pour le *Mimivirus* d'amibe [La Scola et al. 2003]), et sont plus gros que certains génomes bactériens.

Tableau 4.1 : Familles de bactériophages dsDNA

Dans notre étude, nous nous intéressons en particulier aux 163 phages dsDNA (tableau 4.2) : *Myoviridae*, *Sipoviridae*, *Podoviridae*, *Tectiviridae*, *Corticoviridae*, *Plasmaviridae*, *Fuselloviridae*, *Lipothrixviridae* et des phages *dsDNA non annotés*.

Classification NCBI-ICTV	Nombre de génomes	Représentation des 163 génomes dans 602 VOG			
		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Myoviridae	27	Détails en Annexe C			
Siphoviridae	81				
Podoviridae	30				
Tectiviridae	2				
Corticoviridae	1				
Plasmaviridae	1				
Fusselloviridae	4				
Lipothrixviridae	2				
dsDNA non annotés	15				
Nombre total	163				

Recensement effectué sur NCBI en juillet 2006 sur l'état des données statué par le site, le 12 avril 2004.

Tableau 4.2 : Familles de bactériophages dsDNA étudiées

4.1.7 APPROCHE ORIGINALE DE CLASSIFICATION

L'objectif de l'étude est de présenter une classification des bactériophages en tenant compte des deux hypothèses d'évolution : l'évolution classique de transmission verticale et l'évolution réticulée de transferts horizontaux de gènes.

Notre approche originale se résume comme suit. Les données d'étude de départ sont des VOG et les résultats attendus sont : l'arbre phylogénétique d'espèces reconstruit, et les reports sur cet arbre sont des statistiques de transferts THG et des séquences de protéines ancestrales reconstruites tels qu'illustrés sur la figure 4.5.

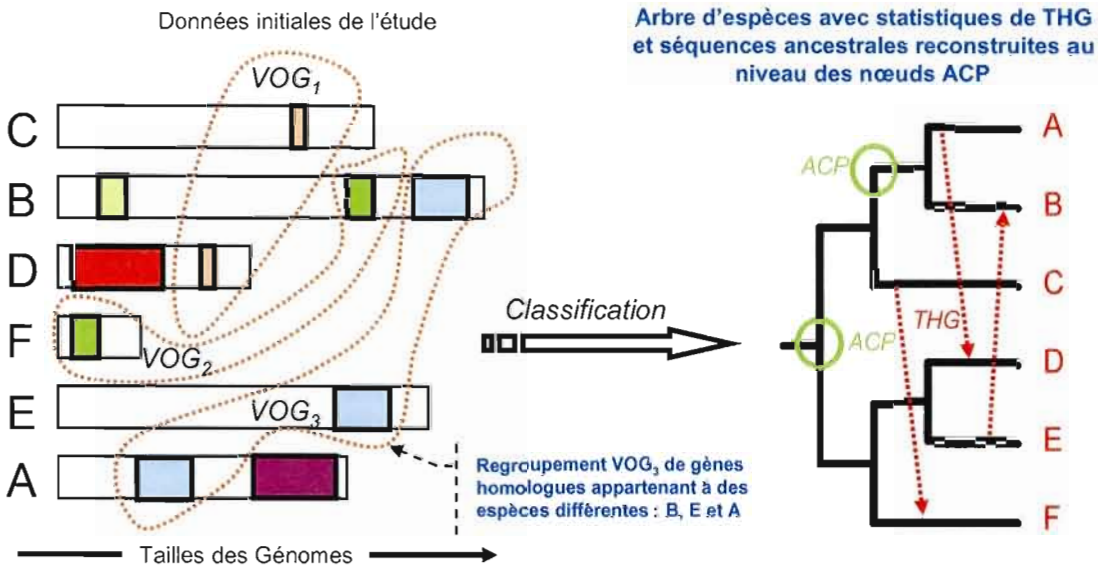


Figure 4.5 : Classification des bactériophages

Les illustrations de regroupements VOG sont censées être une simplification schématique de la figure 4.3.

La figure 4.5 illustre la classification des 6 phages A, B, C, D, E et F. Partant des données VOG réparties parmi les 6 espèces et les regroupements de leurs gènes (partie gauche de la figure 4.5), le but est de classer ces espèces en arbre phylogénétique (partie droite de la figure 4.5) avec des statistiques de transferts THG et la reconstruction de séquences de protéines ancestrales reportées sur les nœuds internes représentant les ancêtres communs les plus proches (ACP).

4.2 RECONSTRUCTION D'ARBRES PHYLOGÉNÉTIQUES DES BACTÉRIOPHAGES

Les expériences et tests décrits ci-après ont été réalisés sur la plate-forme d'inférence phylogénétique que nous avons développée. Nous avons produit d'abord l'arbre phylogénétique d'espèces des bactériophages, et ce suivant la méthode de distances et la méthode bayésienne, ensuite, l'arbre de gène pour chacun des 602 regroupements VOG par la méthode bayésienne.

La reconstruction d'arbre phylogénétique a généré une représentation suivant l'hypothèse d'évolution classique des bactériophages. Cette représentation en arbre ne tient évidemment pas compte des transferts horizontaux de gènes.

La seconde partie de ce chapitre commence par une vue générale sur les différents cheminements de traitement, suit ensuite la description de la reconstruction d'arbre d'espèces et de gène.

4.2.1 VUE GÉNÉRALE

La figure 4.6 montre les cheminements pour reconstruire les arbres phylogénétiques d'espèces et de gène à partir des données VOG.

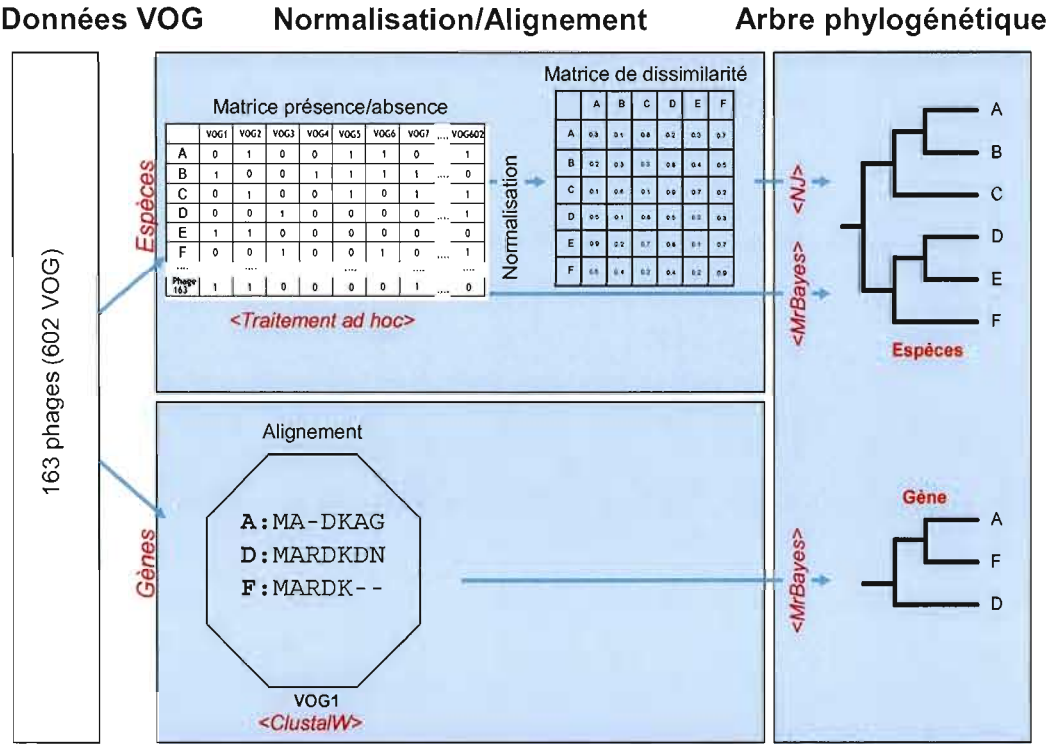


Figure 4.6 : Cheminements de la reconstruction d'arbres d'espèces et de gène

Pour la reconstruction d'arbre d'espèces, on emploie d'abord une méthode de distances (i.e., NJ) puis de manière alternative, une méthode bayésienne (i.e., MrBayes). La génération d'arbre phylogénétique d'espèces par le programme NJ, implémenté dans le package

PHYLIP [Felsenstein, 2004], consiste à prendre en entrée la matrice de dissimilarités inter-génomiques préalablement calculée, tandis que la génération d'arbre d'espèces par le programme MrBayes, consiste à prendre directement en entrée la matrice binaire de présence/absence de gènes dans les regroupements VOG.

Pour la reconstruction d'arbres de gène, on emploie, après avoir comparé les résultats obtenus des deux méthodes (section 4.2.2.6), la méthode bayésienne. La génération d'arbres phylogénétiques de gène par le programme MrBayes, consiste à prendre en entrée, pour chacun des 602 VOG, l'alignement de séquences de protéines réalisé préalablement par le programme ClustalW [Thompson et al. 1994]. Finalement, durant les tests de robustesse des arbres obtenus, on utilise les techniques de bootstrap dans le cas de NJ et de génération de l'arbre consensus sur la collection d'arbres dans le cas de MrBayes.

4.2.2 RECONSTRUCTION D'ARBRES D'ESPÈCES

4.2.2.1 Matrice binaire de présence/absence de gènes

Le point de départ de notre analyse consiste à estimer les distances entre les génomes complets des espèces étudiées. Nous avons commencé par la construction d'une matrice binaire de présence et d'absence de gènes chez les espèces étudiées. Dans le cas des bactériophages, cette matrice comprend 163 lignes (i.e., nombre d'espèces) et 602 colonnes (i.e., nombre de regroupements VOG) contenant des '1' (présence du gène dans le regroupement correspondant) et des '0' (absence du gène) (figure 4.7a).

4.2.2.2 Dissimilarités inter-génomiques

Pour mesurer les dissimilarités inter-génomiques, nous avons évalué différents coefficients : le coefficient de Jaccard [Glazko et al. 2005], le coefficient de Maryland Bridge [Mirkin & Koonin 2003] et la Moyenne Pondérée [Dutilh et al. 2004] (section 3.6.2.1). Les résultats obtenus étaient très similaires, compte tenu qu'il n'y a pas d'ordre *a priori* dans les regroupements VOG. Aussi, nous avons retenu arbitraitement pour la suite de l'étude, le coefficient de Jaccard pour le calcul de la matrice de distances (figure 4.7b).

(a) *Matrice binaire de présence/absence de gènes dans les VOG*

	VOG1	VOG2	VOG3	VOG4	VOG5	VOG6	VOG7	....	VOG602
A	0	1	0	0	1	1	0		1
B	1	0	0	1	1	1	1	....	0
C	0	1	0	0	1	0	1		1
D	0	0	1	0	0	0	0	....	1
E	1	1	0	0	0	0	0		0
F	0	0	1	0	0	1	0	....	1
....		....			....		....		....
Phage 163	1	1	0	0	0	0	1	....	0

(b) *Matrice de dissimilarité inter-génomique*

	A	B	C	D	E	F
A	0.3	0.1	0.8	0.2	0.3	0.7
B	0.2	0.3	0.3	0.8	0.4	0.5
C	0.1	0.6	0.1	0.9	0.7	0.2
D	0.5	0.1	0.8	0.5	0.2	0.3
E	0.9	0.2	0.7	0.8	0.1	0.7
F	0.5	0.4	0.2	0.4	0.2	0.9

Figure 4.7 : Matrices de présence/absence de gènes (a) et de dissimilarité inter-génomique (b)

4.2.2.3 Reconstruction d'arbre d'espèces avec NJ

Une fois la matrice de dissimilarité calculée, elle a été utilisée en entrée du programme NJ lequel a permis d'inférer l'arbre phylogénétique d'espèces non enraciné.

L'arbre phylogénétique d'espèces inféré par NJ est présenté à la figure 5.1b de la section 5.1.1, page 120.

4.2.2.4 Reconstruction d'arbre d'espèces avec MrBayes

L'emploi de la méthode concurrente bayésienne permet de confronter les résultats obtenus avec la méthode de distance précédente (i.e., NJ). Le programme MrBayes a été utilisé avec les options suivantes : [1] Une distribution uniforme de probabilité *a priori* (*prset*), [2] Un modèle de vraisemblance avec une vitesse de variation gamma (*lset*), [3] 2 millions de générations échantillonnées périodiquement à toutes les 100 générations, [4] Une température à 0.2, [5] 4 chaînes et 2 exécutions indépendantes. En résultat, on obtient 20 000 arbres. Un arbre de consensus a été inféré à partir des 1000 derniers arbres les plus stables (i.e., générations stationnaires).

L'arbre phylogénétique d'espèces inféré par MrBayes est présenté à la figure 5.1a de la section 5.1.1, page 120.

4.2.2.5 Tests de robustesse

Des tests de robustesse ont été effectués pour mesurer le taux de regroupements d'espèces présents dans les arbres obtenus respectivement par NJ et MrBayes.

Dans le cas de NJ, l'évaluation de la stabilité des regroupements présents dans les topologies a été faite à partir des différents échantillons de la matrice binaire, générés par la technique de bootstrap (voir la figure $3.5f_1$ et $3.5f_2$ de la section 3.6.1, page 63). Les différents échantillons aléatoires ont été obtenus à l'aide du programme SeqBoot inclus dans le package PHYLIP [Felsenstein, 2004]. Dans cette étude, 100 échantillons ont été générés. Suite à l'inférence phylogénétique de chacun des échantillons, un arbre de consensus a été finalement produit. Le programme Consense inclus également dans le package PHYLIP a servi à générer l'arbre consensus par la règle de consensus majoritaire (≥50%). Lire par ailleurs, la section 3.6.4.1, "*Rééchantillonnage de données*".

Quant à MrBayes, l'évaluation a été faite à partir des arbres a posteriori générés par la technique d'estimation bayésienne (voir la figure $3.5d_1$ et $3.5g_1$ de la section 3.6.1, page 63). Dans cette étude, 40 000 arbres ont été générés, mais seuls les 1000 arbres les plus stables ont

été considérés dans le calcul de l'arbre consensus. L'option *Consensus* dans MrBayes a servi à construire l'arbre consensus par la règle de consensus majoritaire (≥50%). Lire par ailleurs, la section 3.6.4.2, "*Génération de collections d'arbres*".

4.2.2.6 Sélection du meilleur arbre d'espèces

L'arbre d'espèces peut être reconnu comme robuste dans le cas de MrBayes avec des scores de probabilité a posteriori supérieur à 50% et une majorité proche de 100% (figure 5.2 de la section 5.1.2). Dans le cas de NJ, les scores de bootstrap ont été très variables, variant de 0 à 100%. Il n'a pas été possible de ce fait, d'obtenir avec NJ un arbre consensus avec des regroupements d'espèces stables dans la plupart des cas.

L'étape de sélection entre les deux arbres inférés retient naturellement l'arbre d'espèces généré par MrBayes. Par conséquent, le programme MrBayes a été retenu pour la suite de l'étude.

Le fait que l'arbre généré par NJ soit peu robuste s'explique sans doute par la technique d'échantillonnage de bootstrap qui modifie de manière aléatoire la matrice binaire initiale très sensible. Cette sensibilité trouve-t-elle son origine dans l'approche de regroupement VOG (section 4.1.6) ? Ceci est une question ouverte qui se prête bien à une analyse approfondie, mais qui ne sera pas couverte par la présente étude. A noter, toutefois, que la même matrice binaire a été utilisée avec succès par MrBayes. Tout ceci interpelle évidemment notre questionnement qui a été posé à la section 3.6.3.1 concernant la recherche de la relation entre les valeurs de bootstrap et les valeurs de probabilité a posteriori bayésiennes [Delsuc & Douzery 2004].

L'arbre NJ trouvera cependant son intérêt lorsque nous allons confronter l'arbre généré par NJ avec celui généré par MrBayes pour constituer des groupes d'espèces (i.e., clades) retrouvés dans notre analyse (voir la figure 5.1 de la section 5.1.1).

4.2.3 RECONSTRUCTION D'ARBRES DE GÈNE

Avec la méthode bayésienne choisie, il s'agit maintenant de reconstruire un arbre de gène pour chaque groupement VOG.

Le programme MrBayes a été utilisé pour inférer les 602 arbres de VOG (i.e., arbre de gène) à partir des alignements de séquences protéiques de 602 VOG qui ont été préalablement alignées avec ClustalW. Étant donné le nombre modeste de séquences, 7 en moyenne, dans un VOG, seuls 400 arbres ont été générés par groupe de VOG. Ce nombre d'arbres suffisent à atteindre l'état stationnaire généré par MrBayes. La totalité des 400 arbres a été considéré dans le calcul de l'arbre consensus majoritaire. L'option *Consensus* dans MrBayes a servi à construire l'arbre de gène par la règle de consensus majoritaire (≥50%) (voir la figure $3.5d_2$ et $3.5g_2$ de la section 3.6.1, page 63). Lire par ailleurs, la section 3.6.4.2, "*Génération de collections d'arbres*".

Bien qu'il ne soit pas pertinent de représenter les 602 arbres de gène (résultats intermédiaires) dans ce document, ils sont néanmoins disponibles sur demande.

Les résultats que nous en retirions de ces expériences ont été mis à contribution dans la détection des THG (i.e., la confrontation de l'arbre d'espèces et chacun des 602 arbres de gène, voir la section suivante) d'une part, et d'autre part, dans la reconstruction de séquences ancestrales (i.e., qui utilise en entrée les 602 arbres de gène et alignements VOG, et l'arbre d'espèces, voir la section 4.4). Les représentations finales sont montrées respectivement à la figure 5.3 de la section 5.2.2, page 127, et à la figure 5.4 de la section 5.3.1.1, page 129.

4.3 DÉTECTION DES TRANFERTS HORIZONTAUX DE GÈNES

Les expériences et tests décrits ci-après ont été réalisés sur la plate-forme d'inférence phylogénétique que nous avons développée. Nous avons calculé les statistiques de transferts THG, en confrontant l'arbre d'espèces et les 602 arbres de gène.

En complément à la reconstruction d'arbres phylogénétiques d'espèces et de gène de l'étape précédente, la détection des transferts horizontaux de gènes s'effectue par le biais d'une représentation en réseaux des espèces suivant l'hypothèse d'évolution réticulée des bactériophages. Pour le cheminement des traitements, voir la figure 3.5c de la section 3.6.1, page 63.

Nous présentons ici les modèles de représentation en réseaux et la méthode avec laquelle la détection des THG inter/intra groupes a été réalisée.

4.3.1 MODÈLES DE REPRÉSENTATION EN RÉSEAUX

Les THG sont décrits typiquement par le scénario hypothétique des transferts. Des modèles en réseaux sont nécessaires pour décrire ce genre de scénario.

4.3.1.1 Scénario hypothétique

Les transferts horizontaux de gènes sont caractérisés par des processus qui suppriment les barrières, les frontières entre espèces (en héritant d'autres espèces), de type unidirectionnel dans la majorité des cas (pas d'échange réciproque d'ADN), et qui peuvent impliquer plusieurs gènes ou seulement une partie d'un gène. Le transfert se fait par l'acquisition d'ADN ou via des virus (section 4.1.3).

C'est aussi un processus caractérisé par six étapes importantes d'un scénario hypothétique de transferts [Eisen 2000 ; Ochman et al. 2000] (figure 4.9) :

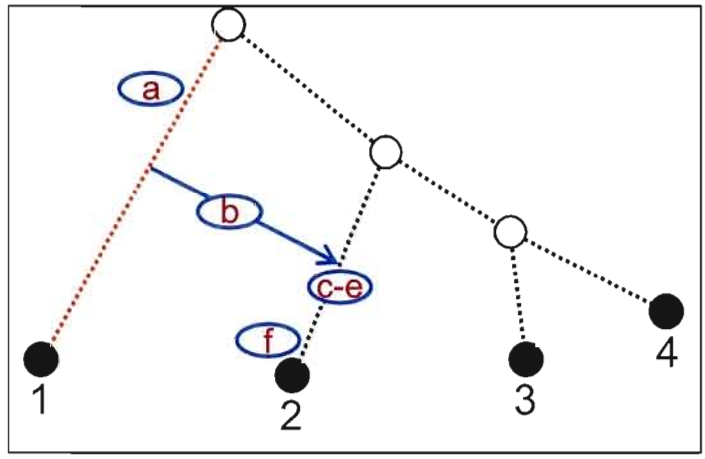


Figure 4.8 : Les six étapes d'un scénario hypothétique

Les six étapes d'un scénario hypothétique de transferts sont décrites comme suit [Eisen 2000] : l'étape (a) : un gène évolue et il est adapté à l'espèce 1 ; l'étape (b) : le gène est transféré de la lignée 1 à la lignée 2 via un vecteur (virus, accouplement, échange d'ADN) ; l'étape (c) : les gènes doivent être dans un format adéquat pour la maintenance et l'évolution à long terme face au mécanisme de spéciation de l'espèce 2 ; l'étape (d) : durant cette étape, le gène est soumis à la pression de la sélection quant à son utilité pour l'espèce (e.g., si le

gène est résistant aux antibiotiques, on le garde) ; l'étape (e) : le gène doit se répandre dans la population ; l'étape (f) : c'est le processus d'évolution « amélioration » (i.e., adaptation à son nouveau milieu).

4.3.1.2 Modèles en réseaux

Le scénario hypothétique des transferts horizontaux peut être décrit par une approche de modélisation en réseaux.

Il existe plusieurs approches de modélisation et de détection des transferts horizontaux. Page et Charleston [1998] ont décrit un ensemble de règles d'évolution devant être prises en compte lors de la modélisation des transferts. Mirkin et al. [1995] ont suggéré une méthode de réconciliation d'arbres permettant de combiner plusieurs phylogénies de gènes afin de recontruire un arbre d'espèces unique. Hallett et Lagergren [2001] ont proposé un modèle de détection de transferts permettant d'inscrire les phylogénies de gènes en phylogénie d'espèces. Boc et Makarenkov [2003] et Makarenkov et al. [2004] ont introduit deux méthodes de détection impliquant des scénarios uniques et multiples des transferts horizontaux.

On a, parmi les modèles proposés, NeighborNet et SplitsTree [Bryant & Moulton 2002 ; Huson 1998] et HGT-Detection [Makarenkov et al. 2008].

Le modèle implémenté dans le programme SplitsTree est basé sur la méthode de reconstruction de réseaux et de représentation combinant le principe de *Split Decomposition* [Bandelt & Dress 1992] qui transforme les distances d'évolution en une somme de bifurcations (*splits*) faiblement compatibles.

Quant au modèle implémenté dans le programme HGT-Detection du package T-Rex (www.trex.uqam.ca), il est basé sur la méthode de réconciliation topologique de Makarenkov et al. [2008] entre l'arbre de gène et l'arbre d'espèces, et la différence se traduit par l'ajout de branches THG supplémentaires dans l'arbre d'espèces pour produire finalement une représentation en réseau.

Parallèlement à ces méthodes, mais dans la même veine, notons également le programme HGT-Simulator [Nguyen et al. 2005] qui, comme son nom l'indique, simule l'évolution des séquences d'ADN le long d'une phylogénie donnée, en générant aléatoirement les transferts horizontaux, tout en respectant certaines règles d'évolution biologiques de base dont l'interdiction de transferts sur la même lignée [Page & Charleston 1998].

Pour un état de l'art sur les approches et modèles en réseaux, on peut se référer à Huson [2005] et Makarenkov et al. [2008]. Outre les modèles pré-cités, on recense aussi Median Networks [Bandelt et al. 1995, 2000], Median-Joining Networks [Foulds et al. 1979 ; Bandelt et al. 1999], Molecular Variance Parsimony [Excoffier & Smouse 1994], Pyramides [Diday & Bertrand 1984] et Hiérarchies Faibles [Bandelt & Dress 1989].

4.3.2 DÉTECTION DE THG

4.3.2.1 Méthode de détection

Après avoir reconstruit l'arbre d'espèces et identifié les 22 groupes (voir la figure 5.1 de la section 5.1.1, page 120), le traitement se concentre à présent sur la détection des transferts THG à travers les mouvements d'échanges inter et intra-groupes. Nous avons retenu l'approche de détection proposée par Makarenkov et al. [2008].

Les méthodes de détection impliquant des scénarios partiels et complets des transferts horizontaux utilisent le concept de distances et sont basées sur la réconciliation des topologies des arbres d'espèces et de gène reconstruits pour le même ensemble d'espèces. Le premier scénario suppose le transfert partiel. Il est basé sur le calcul et l'optimisation de la distance en longueur du chemin minimum dans un réseau orienté. Dans ce modèle, l'arbre phylogénétique est transformé en un graphe connecté et orienté dans lequel une paire d'espèces peut être liée par plusieurs chemins. Alors que le second scénario suppose le transfert complet : l'arbre d'espèces est graduellement transformé en arbre de gène par ajout de THG à chaque étape. Durant cette transformation, seules les topologies d'arbres sont considérées et modifiées. Bien que le second modèle soit moins général, un algorithme efficace et rapide est décrit pour résoudre le problème.

En ce qui concerne l'optimisation, deux critères sont proposés : le coefficient des moindres carrés et la distance de Robinson & Foulds (RF) [Robinson & Foulds 1981], l'un est métrique, l'autre est topologique. Selon Makarenkov et al. [2008], l'utilisation du critère topologique, comparé au critère métrique, donne de meilleurs résultats de détection.

La distance topologique de Robinson & Foulds entre deux arbres phylogénétiques est égale au nombre minimum d'opérations élémentaires permettant de fusionner ou séparer les nœuds afin de transformer un arbre (e.g., *T* de la figure 4.9a) en un autre (e.g., *T'* de la figure 4.9a). La distance de RF entre *T* et *T'* (figure 4.9b) est égale à deux opérations de transformation. Le transfert THG qui minimise la distance de RF entre les arbres phylogénétiques d'espèces et de gène peut être considéré comme le meilleur candidat pour réconcilier les phylogénies d'espèces et de gène.

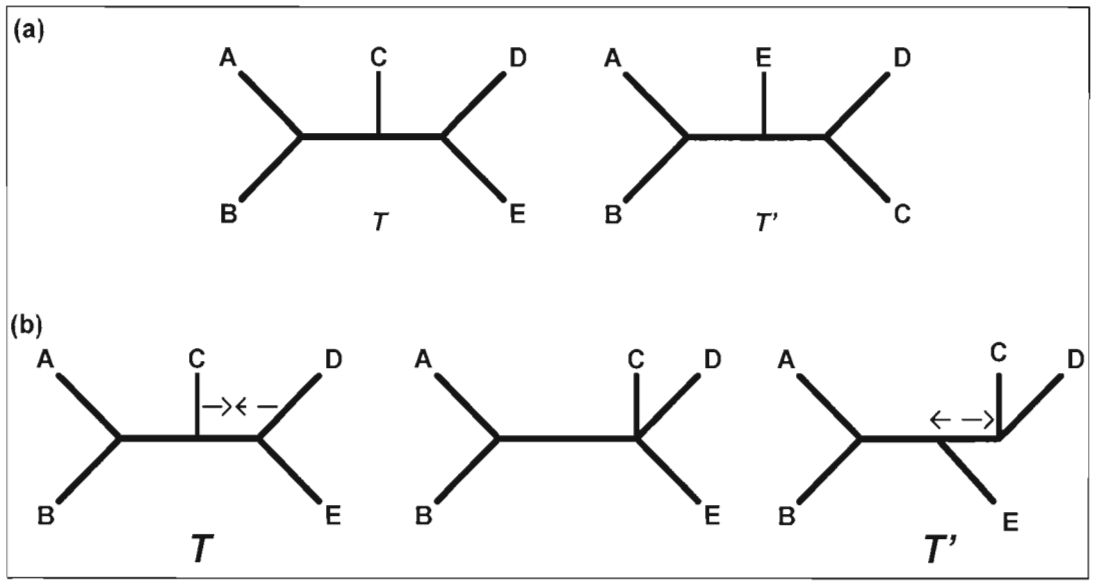


Figure 4.9 : Calcul de la distance topologique de RF par fusion et séparation des nœuds
(Tirée de Boc et al. 2004)

4.3.2.2 Algorithme de détection des THG

La détection des THG a été effectuée, en utilisant le programme HGT-Detection, suivant une version améliorée de l'algorithme de réconciliation topologique entre l'arbre de gène et l'arbre d'espèces [Makarenkov et al. 2008]. HGT-Detection prend en entrée un arbre d'espèces et un arbre de gène pour le même ensemble d'espèces. Les THG sont ainsi calculés, en indiquant en sortie l'origine et la destination pour chacun des transferts inférés. Les principales étapes de l'algorithme heuristique pour identifier des THG sont les suivantes :

Pas préliminaire

Inférer les arbres phylogénétiques d'espèces et celui de gène, notés respectivement T et T'. Dans cette étude, l'arbre T est un arbre réduit de l'arbre complet construit pour 163 bactériophages et contenant seulement les phages présents dans le VOG considéré. Les deux arbres doivent être enracinés (e.g., par la technique du point médian, *midpoint*). S'il existe dans T et T' des sous-arbres identiques ayant au moins 2 feuilles, réduire la taille du problème en remplaçant dans T et T' les sous-arbres identiques par les mêmes éléments auxiliaires.

Pas 1 ... k

Tester tous les THG possibles entre les paires d'arêtes dans l'arbre T_{k-1} ($T_{k-1} = T$ au Pas 1) à l'exception des transferts entre les arêtes adjacentes et ceux qui violent les contraintes d'évolution (pour plus de détails voir Page et Charleston [1998]). Choisir en tant que THG optimal, le déplacement d'un sous-arbre dans T_{k-1} qui minimise la valeur de la distance topologique de RF entre l'arbre obtenu après le déplacement de ce sous-arbre et de son greffage sur une nouvelle arête, c'est-à-dire l'arbre T_k, et l'arbre de gène T'. Réduire ensuite la taille du problème en remplaçant des sous-arbres identiques, ayant au moins 2 feuilles, dans l'arbre transformé T_k et l'arbre de gène T'. Dans la liste des THG retrouvés, rechercher et éliminer les THG inutiles en utilisant une procédure de programmation dynamique de parcours en arrière. Un transfert inutile est celui dont l'élimination ne change pas la topologie de l'arbre T_k.

Conditions d'arrêt et complexité algorithmique

L'algorithme s'arrête quand la distance de RF devient égale à 0 ou quand aucun autre déplacement de sous-arbres n'est possible suite à des contraintes biologiques. Théoriquement, une telle procédure requiert $O(kn^4)$ opérations pour prédire k transferts dans un arbre phylogénétique à n feuilles. Cependant, due à des réductions inévitables des arbres d'espèces et de gène, la complexité pratique de cet algorithme est plutôt $O(kn^3)$.

4.3.3 REPORT DES TRANSFERTS THG

La figure 4.10 illustre la confrontation entre l'arbre d'espèces T et l'arbre de gène T' pour le même ensemble d'espèces (i.e., A, B, C, D, E et F). Les arbres T et T' n'ont pas la même topologie au niveau des espèces B, D et F (partie gauche de la figure 4.10). Les trois transferts THG transforment successivement T en $T + \text{THG}_1$, puis en $T + \text{THG}_2$, et finalement en $T + \text{THG}_3$, ce qui est équivalent à la topologie T' de l'arbre de gène au départ (partie centrale de la figure 4.10). On obtient en résultat un arbre d'espèces ayant détecté trois transferts caractérisés par un THG de l'espèce A à l'espèce D, un THG de l'espèce E à l'espèce B et un THG de l'espèce C à l'espèce F (partie droite de la figure 4.10).

Au total, 1451 transferts THG ont été détectés au cours de notre analyse en combinant les arbres de gène (602) et les arbres d'espèces (163). Il était donc impossible de tous les représenter sur une même figure. Par conséquent, nous les avons mis sous forme de tableaux de statistiques (voir tableau 5.2 de la section 5.2.1.2, page 125), et les avons reportés sur la figure 5.3 de la section 5.2.2, page 127.

A noter que parmi ce nombre, il y a aussi des transferts entre espèces n'appartenant à aucun groupe de clades identifié. Les résultats finaux des statistiques de transferts inter et intra groupes sont montrés à la figure 5.2 de la section 5.2.2, page 122.

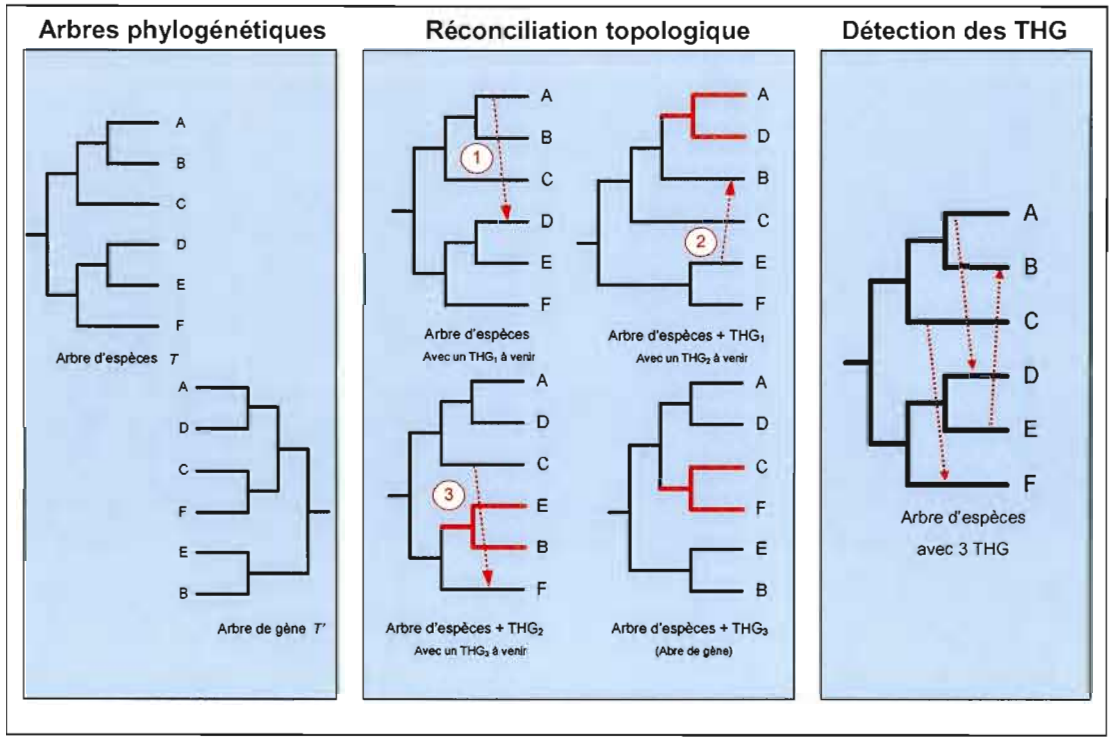


Figure 4.10 : Illustration de 3 THG par confrontation d'arbres d'espèces *T* et de gène *T'*
(Tirée de Boc et al. 2004)

4.4 RECONSTRUCTION ANCESTRALE

Les expériences et tests décrits ci-après ont été réalisés sur la plate-forme d'inférence phylogénétique que nous avons développée. Nous avons généré les séquences de protéines ancestrales, en confrontant l'arbre d'espèces, les 602 arbres de gène et les alignements de séquences des 602 VOG.

En complément à la reconstruction d'arbres phylogénétiques d'espèces et de gène, ainsi qu'à l'alignement des séquences des 602 VOG, effectués préalablement, la reconstruction de séquences de protéines ancestrales produit des séquences ancestrales suivant l'hypothèse d'évolution classique par héritage vertical des bactériophages. Pour le cheminement des traitements, voir la figure 3.5e de la section 3.6.1, page 63.

La reconstruction des séquences protéiques ancestrales s'effectue en deux étapes : reconstruction des ancêtres et représentation des séquences obtenues dans l'arbre d'espèces déjà construit.

4.4.1 APPROCHE

Si le problème de reconstruction ancestrale suivant les scénarios de substitution a toujours été bien étudié (e.g., Felsenstein [1981]), celui suivant les scénarios d'insertions et de délétions (*Indel*), en revanche, a retenu moins d'attention. Cette section présente les scénarios Indel et la recherche du scénario le plus vraisemblable (IMPL).

4.4.1.1 Scénarios Indel

Deux principales approches sont possibles pour la reconstruction du scénario le plus parcimonieux des Indels [Diallo et al. 2006] : l'approche selon le critère de parcimonie et l'approche selon le critère de vraisemblance.

Approches de parcimonie et de vraisemblance

L'approche de parcimonie a été introduite par Fredslund et al. [2004] et Blanchette et al. [2004]. Le principe consiste à essayer de trouver la reconstruction ancestrale avec correction phylogénétique qui minimise le nombre total d'insertions et de délétions. Cette approche a été montrée NP-difficile [Chindelevitch et al. 2006]. De plus, elle ne propose qu'une seule solution en sortie et ne fournit pas de mesures d'incertitude aux différents endroits de la reconstruction.

Quant à l'approche de vraisemblance, elle considère la reconstruction Indel comme un problème de type probabiliste, similaire à celui décrit par le modèle de Thorne-Kishino-Felsenstein [Thorne et al. 1992]. Le principe consiste à définir la probabilité de transition entre une rangée d'alignement et sa rangée descendante. Diallo et al. [2006] ont proposé via les techniques combinant l'arbre phylogénétique et les HMM (soit Tree-HMM) de trouver des scénarios Indel ayant le maximum de vraisemblance afin de résoudre le problème appelé IMLP – *Indel Maximum Likelihood Problem.*

Illustration du scénario Indel

La figure 4.11 illustre une reconstruction ancestrale suivant le scénario Indel. Les entrées consistent en un alignement multiple Indel en format binaire obtenu à partir des séquences orthologues (i.e., les séquences dans les VOG dans notre cas) d'espèces A, B, C, D et E (sur la partie gauche de la figure 4.11, les gaps ('-') sont remplacés par des ('0') et les caractères par des ('1')), ainsi qu'une topologie et des longueurs de l'arbre phylogénétique (scénario Indel, sur la partie centrale). Les sorties consistent en un ensemble de séquences incluant des insertions et des délétions, placées le long des nœuds de l'arbre.

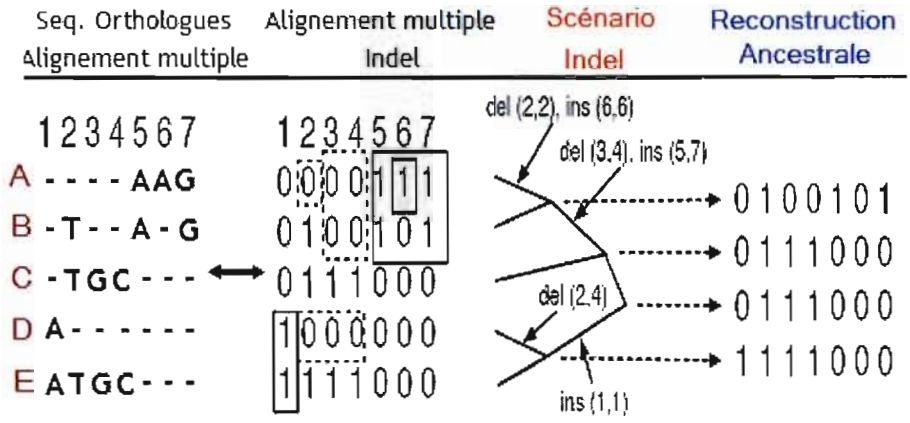


Figure 4.11 : Illustration du scénario Indel
(Adaptée de Diallo et al. 2006)

Les zones en pointillées dans l'alignement multiple Indel indiquent les délétions, et les zones grisées indiquent les insertions dans le scénario Indel. Le scénario Indel spécifie les différentes opérations d'insertion et de délétions (e.g., del(2,2), ins(6,6), del(3,4), etc.) le long des branches de l'arbre phylogénétique. La reconstruction ancestrale résultante est composée de séquences ancestrales binaires d'insertion et de délétion de nucléotides associées aux différents nœuds de l'arbre.

4.4.1.2 Problème IMLP

Le problème IMLP est basé sur le modèle probabiliste où il est nécessaire de définir les probabilités de transition entre une rangée d'alignement A_x^* et sa rangée descendante A_y^*. Cette probabilité est en fait une fonction de probabilité des opérations d'insertion, de délétion et de conservation (quand il n'y a ni insertion ni délétion) qui s'est produite entre la séquence ancêtre et sa descendante.

Formellement, étant donné un arbre $T=(V_T, E_T)$, de nœuds V_T, de branches E_T et la longueur $\lambda(b)$ pour chaque b de E_T, on définit de manière raisonnable les probabilités de délétion, d'insertion et de conservation comme suit [Diallo et al. 2006] :

$$\begin{aligned} P_{DelStart}(\lambda(b)) &= 1 - e^{-\psi_D \lambda(b)} \\ P_{InsStart}(\lambda(b)) &= 1 - e^{-\psi_I \lambda(b)} \\ P_{Cons}(\lambda(b)) &= 1 - e^{-(\psi_D + \psi_I)\lambda(b)} \end{aligned} \tag{4.1}$$

où ψ_I et ψ_D sont les paramètres relatifs à la vitesse d'insertion et de délétion. On suppose que la taille d'une délétion (respectivement d'une insertion) suit une distribution géométrique, où la probabilité d'une délétion de taille k est égale à $\alpha_D^{k-1}(1-\alpha_D)$ (respectivement $\alpha_I^{k-1}(1-\alpha_I)$). Ainsi, la probabilité pour que la rangée d'alignement A_x^* soit transformée en rangée d'alignement A_y^* le long de la branche b peut être définie comme suit :

$$\begin{aligned} \Pr(A_y^* \mid A_x^*, b) = & \prod_{(s,e):\, Deletion\ de\ A_x^*\ a\ A_y^*} P_{DelStart}(\lambda(b)) \cdot \alpha_D^{l(s,e)-1}(1-\alpha_D)) \cdot \\ & \prod_{(s,e):\, Insertion\ de\ A_x^*\ a\ A_y^*} P_{InsStart}(\lambda(b)) \cdot \alpha_I^{l(s,e)-1}(1-\alpha_I)) \cdot \\ & \prod_{(s,e):\, Conservation\ de\ A_x^*\ a\ A_y^*} P_{Cons}(\lambda(b))^{l(s.e)} \end{aligned} \tag{4.2}$$

Finalement, le problème IMLP de reconstruction ancestrale est défini par le maximum de vraisemblance $L(A^*)$, soit :

$$L(A^*) = \prod_{b=(x,y)\in E_T} \Pr(A_y^* \mid A_x^*, b) \tag{4.3}$$

Le problème IMLP peut être résolu par le biais du modèle Tree-HMM.

4.4.2 RECONSTRUCTION DE SÉQUENCES ANCESTRALES

Si le calcul du chemin le plus probable à travers le modèle Tree-HMM, et les algorithmes Viterbi ou Forward-Backward permet d'inférer le scénario Indel, le programme Ancestor permet, quant à lui, de compléter l'inférence Indel avec la prédiction de caractères de nucléotides ou d'aminoacides pour générer des séquences ancestrales.

4.4.2.1 Calcul du chemin le plus probable

Diallo et al. [2006] ont adapté l'algorithme de Viterbi standard [Durbin et al. 2006] (dont l'équation A.7 est donnée en Annexe A.2) pour calculer la vraisemblance du chemin le plus probable à travers le modèle Tree-HMM. Lire par ailleurs, la section 3.4.2, "*Estimation du maximum de vraisemblance Tree-HMM*" (équation 3.7).

Pour diminuer la complexité exponentielle $O(n3^{2n})$ due au nombre des états valides possibles, des optimisations ont été apportées dans l'adaptation de l'algorithme, en faisant des recherches, par le parcours préalable de l'arbre, des états valides réels : étant donné un alignement A, il est possible de calculer pour chaque colonne $A[i]$, l'ensemble S_i des états réellement valides qui peuvent émettre $A[i]$ avec une probabilité non nulle.

Par ailleurs, un des avantages de l'approche du maximum de vraisemblance sur l'approche de parcimonie mentionnée plus tôt, est que celle-ci permet l'évaluation de l'incertitude relative à certains aspects de la reconstruction [Diallo et al. 2006]. Par exemple, il est utile de pouvoir calculer la probabilité qu'une base était présente à une position i d'un nœud ancestral donné u : $\Pr\left[A_u^*[i] = 1 \middle| A\right] = \sum_{s\in S:O_s(u)=1} \Pr\left[\pi_i = s \middle| A\right]$. Cela permet de calculer la probabilité de faire une prédiction incorrecte de la position donnée d'un ancêtre donné. L'algorithme Forward-Backward (dont l'équation est donnée en Annexe A.2, équation A.8) est un algorithme HMM standard [Durbin et al. 2006] qui permet justement de calculer la probabilité $\Pr\left[\pi_i = s \middle| A\right]$ [Diallo et al. 2006] :

$$\Pr[\pi_i = s|A] = \frac{F(i,s)\,B(i,s)}{\sum_{s' \in S_i'} F(i,s')\,B(i,s')}$$

avec

$$F(i,s) = \begin{cases} 1, & si\ i = 0\ et\ s = c \\ 0, & si\ i = 0\ et\ s \neq c \\ e(A[i]|s) \cdot \sum_{s' \in S_{i-1}'} (F(i-1,s') \cdot a(s|s'), & si\ s > 0 \end{cases} \qquad (4.4)$$

$$B(i,l) = \begin{cases} 1, & si\ i = L+1\ et\ l = c \\ 0, & si\ i = L+1\ et\ l \neq c \\ \sum_{s' \in S_{i+1}'} e(A[i+1]|s') \cdot F(i+1,s') \cdot a(s'|s), & si\ i < L+1 \end{cases}$$

où $F(\cdot)$ est le chemin calculé par l'algorithme Forward (Annexe A.2, équation A.5), $B(\cdot)$ est le chemin calculé par l'algorithme Backward (Annexe A.2, équation A.6), $e(\cdot)$ est la probabilité d'émission, $a(\cdot)$ est la probabilité de transition, L est la longueur d'Indel et s' est l'état qui précède l'état s. Les optimisations développées pour l'algorithme de Viterbi ci-dessus peuvent être facilement adaptées à ce nouvel algorithme. Pour plus de détails, se référer à Diallo et al. [2006].

4.4.2.2 Génération de séquences protéiques ancestrales

Au préalable, les séquences protéiques de chaque VOG ont été alignées en utilisant le programme d'alignement de séquences multiples ClustalW [Thompson et al. 1994]. Les arbres phylogénétiques représentant l'histoire évolutive de chacun des VOG ont été reconstruits à l'aide du programme MrBayes. L'arbre consensus de gène a été inféré, puis enraciné, en utilisant la technique du point médian (*midpoint*).

Étant donné un alignement de séquences de régions orthologues et un arbre phylogénétique, la reconstruction de séquences ancestrales consiste à l'inférence, pour chaque nœud interne de l'arbre phylogénétique, des séquences génomiques correspondante. Cette inférence s'effectue en deux étapes : la reconstruction du scénario Indel le plus vraisemblable et l'inférence des séquences à chaque position des ancêtres où la présence d'un caractère a été prédite. Ces deux étapes ont été réalisées respectivement par les algorithmes

de Diallo et al. [2006] et Felsenstein [1981] qui sont implantés dans le programme *Ancestor* disponible à l'URL suivant : <www.mcb.mcgill.ca/~banire/ancestor>.

La génération de séquences protéiques ancestrales a été effectuée pour chacun des 602 VOG. Les séquences ancestrales obtenues sont représentées sur l'arbre d'espèce par le biais des nœuds internes en guise d'ancêtres communs les plus proches (ACP). Cette représentation permet d'identifier au cours de l'évolution les diverses apparitions de nouvelles fonctions associées aux protéines étudiées.

4.4.3 REPORT DES SÉQUENCES ANCESTRALES

La reconstruction de séquences ancestrales est illustrée à la figure 4.12 montrant la confrontation entre l'arbre de gène 1 (VOG_1), l'arbre de gène 2 (VOG_2) et l'arbre d'espèces (partie gauche de la figure 4.12). La séquence ancestrale g2 est l'ancêtre commun le plus proche des espèces A, B et C, et la séquence ancestrale g1 est l'ancêtre commun le plus proche des espèces A, F et D (partie centrale de la figure 4.12). Les deux séquences protéiques ancestrales g1 et g2 sont reportées sur les deux nœuds internes de l'arbre d'espèces (partie droite de la figure 4.12).

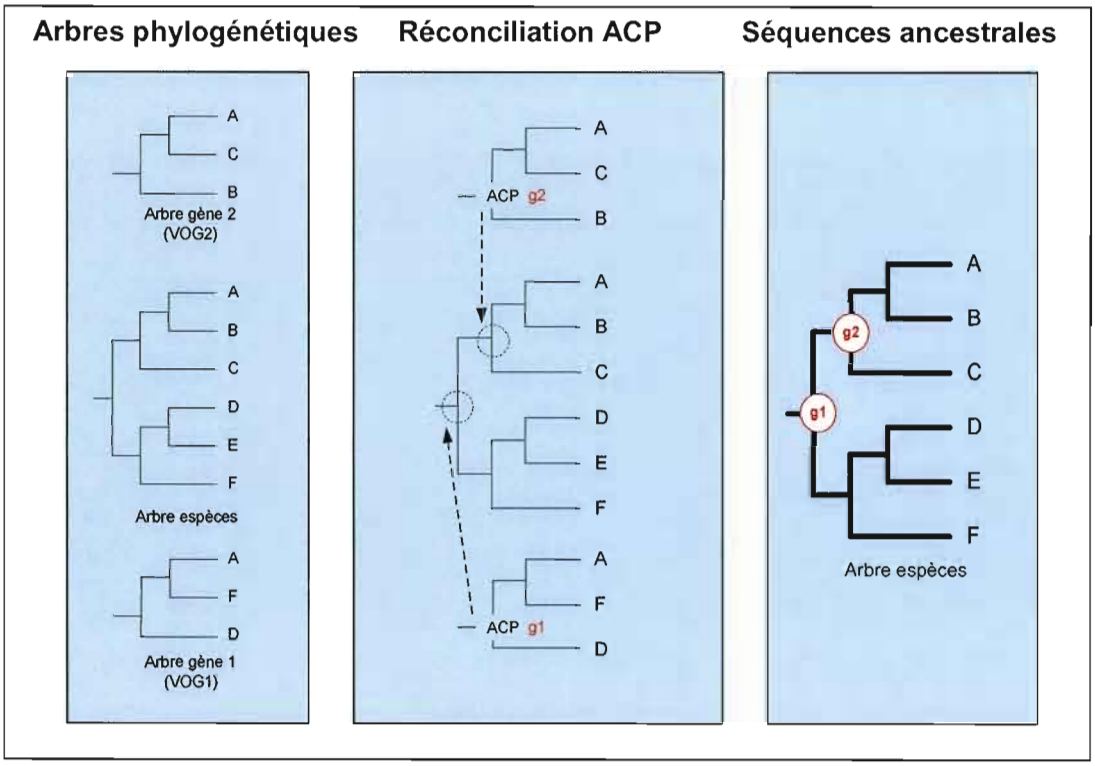


Figure 4.12 : Illustration deux séquences ancestrales g1 et g2 représentées sur l'arbre d'espèces

Avec l'arbre d'espèces produit précédemment (section 4.2.2.6) et représenté à la figure 5.2 de la section 5.1.2, on en énumère 114 nœuds internes numérotés à partir de 0 (i.e., la racine). Ces nœuds internes représentent potentiellement des nœuds ACP d'ancêtres communs les plus proches. Comme dans le cas des transferts THG, il était impossible de tous les représenter sur une même figure, d'autant plus que chaque ACP peut avoir une ou plusieurs séquences protéiques ancestrales. Par conséquent, nous avons reporté seulement les numéros des nœuds internes sur la figure 5.4 de la section 5.3.1.1. Les séquences ancestrales associées à chacun des nœuds internes sont listées en Annexe E.

4.5 CLASSIFICATION DES BACTÉRIOPHAGES

De ce chapitre, nous retiendrons les points suivants :

D'un côté, les bactériophages constituent l'un des groupes d'organismes les plus abondants dans la biosphère (section 4.1.3). Leur mode d'évolution est complexe et régi en partie par des mécanismes réticulés dont le transfert horizontal de gènes (sections 1.1.1.3 et 4.1.2). Ceci a sans doute pour conséquence des données observées caractérisées par des grandes variations à la fois dans la taille et la composition génétique des génomes (section 4.1.6.1).

De l'autre côté, le recensement des phages est toujours en cours et les classifications proposées sont nombreuses et diverses (section 4.1.4). La difficulté intrinsèque, due à la diversité des modes d'évolution et à la complexité de l'écosystème des sujets, est telle qu'une classification exhaustive et de consensus des bactériophages reste à établir.

Nous avons présenté dans ce chapitre, une approche originale de classification des bactériophages qui combine les méthodes de détection des transferts horizontaux de gènes d'une part (section 4.3) et de reconstruction de séquences ancestrales d'autre part (section 4.4). Les résultats expérimentaux montrés au chapitre suivant viendront souligner la pertinence de l'approche proposée.

CHAPITRE V

–

RÉSULTATS

Plan du chapitre

Cet ultime chapitre comprend les principaux résultats obtenus au cours de nos expériences. D'autres détails sont donnés en Annexes B, C, D et E qui seront référencés plus loin.

La première partie présentera l'arbre phylogénétique des bactériophages qui nous a permis d'identifier 22 groupes de phages dont 12 référencés par l'organisme ICTV. La seconde partie est consacrée à la représentation de la détection des transferts horizontaux de gènes à travers les mouvements d'échanges inter et intra-groupes. La dernière partie consistera en la représentation de l'arbre d'espèces et des séquences protéiques ancestrales, et se focalisera plus spécifiquement sur l'exemple de la famille des phages infectant les bactéries du lait, les *Lactococcal lactis*. Notons que cet exemple qui consistera en fait en la comparaison structurale de protéine sera donné afin de souligner la pertinence des résultats obtenus sur la classification. Nous n'avons pas la prétention ici d'attaquer à la problématique de comparaison atomique des différentes protéines.

5.1 ARBRE PHYLOGÉNÉTIQUE DES BACTÉRIOPHAGES

On identifie les groupes en confrontant les arbres inférés, et on représente sur l'arbre d'espèces, les différents groupes en soulignant ceux qui correspondent aux groupes reconnus par NCBI/ICTV.

5.1.1 IDENTIFICATION DE GROUPES

Deux arbres phylogénétiques d'espèces ont été inférés par les programmes NJ et MrBayes. Nous les avons confrontés afin de faire ressortir des groupes de clades. La figure 5.1 montre à la fois les similarités et les différences entre les deux arbres.

Comme similarités, on peut noter qu'après confrontation, les regroupements des espèces sont semblables dans les deux arbres. On identifie ainsi 22 groupes d'espèces de tailles différentes variant de 3 (groupes 4, 9 et 10) à 10 espèces (groupe 14). De manière remarquable, les 22 groupes sont composés chacun des mêmes espèces identifiées dans les deux arbres. Les détails de la figure 5.1 sont donnés en Annexe D.

La première différence des deux arbres réside dans leur topologie respective : l'arbre inféré par NJ est binaire alors que celui inféré par MrBayes admet des nœuds multibifurcations. La seconde différence est au niveau de la robustesse des arbres inférés.

Celui inféré par MrBayes (i.e., de la figure 5.1) est soumis aux tests de robustes tandis que celui par NJ (i.e., de la figure 5.1) ne l'est pas. Si on l'avait soumis aux tests de bootstrap (section 4.2.2.6), l'arbre inféré par NJ (i.e., qui n'est pas celui de la figure 5.1) ne donnerait pas de clades aussi consistants, tel qu'on peut voir sur la figure 5.1, et donc ne peut être confronté avec l'arbre inféré par MrBayes pour identification de groupes.

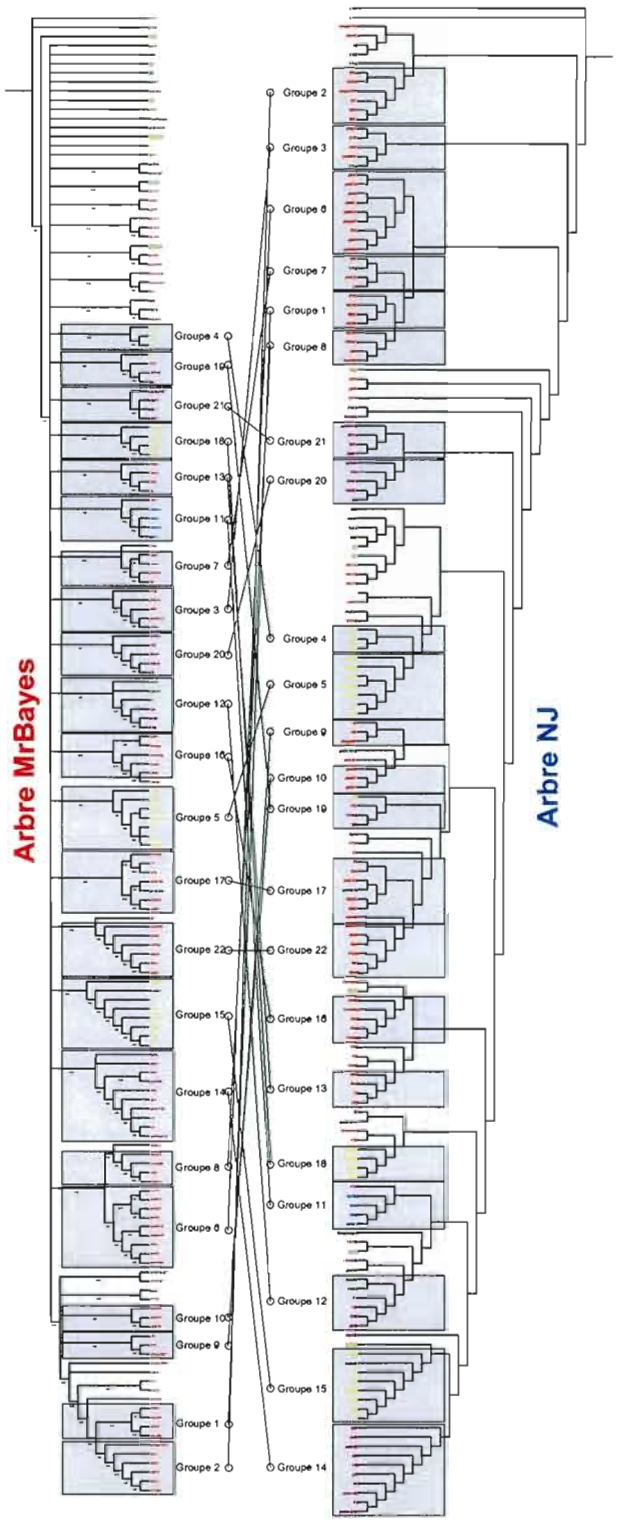


Figure 5.1 : Les 22 groupes identifiés par confrontation des arbres générés par MrBayes et NJ

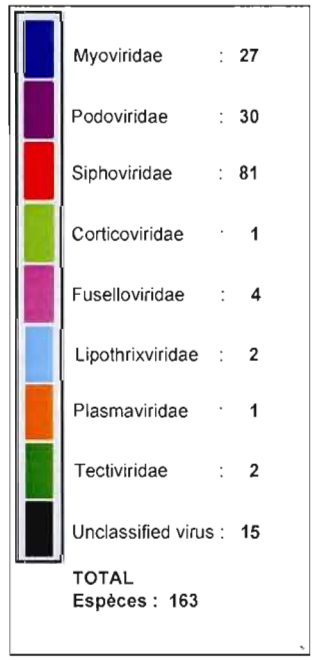
5.1.2 REPRÉSENTATION DE L'ARBRE D'ESPÈCES

Comme cela a été discuté à la section 4.2.2.6, l'arbre d'espèces des phages inféré par MrBayes a été retenu parce qu'il donnait les meilleurs scores de robustesse.

La figure 5.2 montre l'arbre phylogénétique d'espèces inféré par le programme MrBayes. Les scores de robustesse sont indiqués en rouge pour les arêtes internes, variant de 50 à 100 % avec une grande majorité à 100 %. Ce qui confère un degré de confiance élevé quant aux groupes trouvés.

La figure 5.2 a été dessinée à l'aide de l'outil de représentation d'arbres *iTol* (disponible sur : http://itol.embl.de) [Letunic & Brok 2007].

Note : le code de couleurs présenté ci-après correspond aux différentes familles de phages étudiées. Ce code est utilisé pour les figures d'arbre (5.2, 5.3, 5.4 et 5.7) dans les sections à venir.



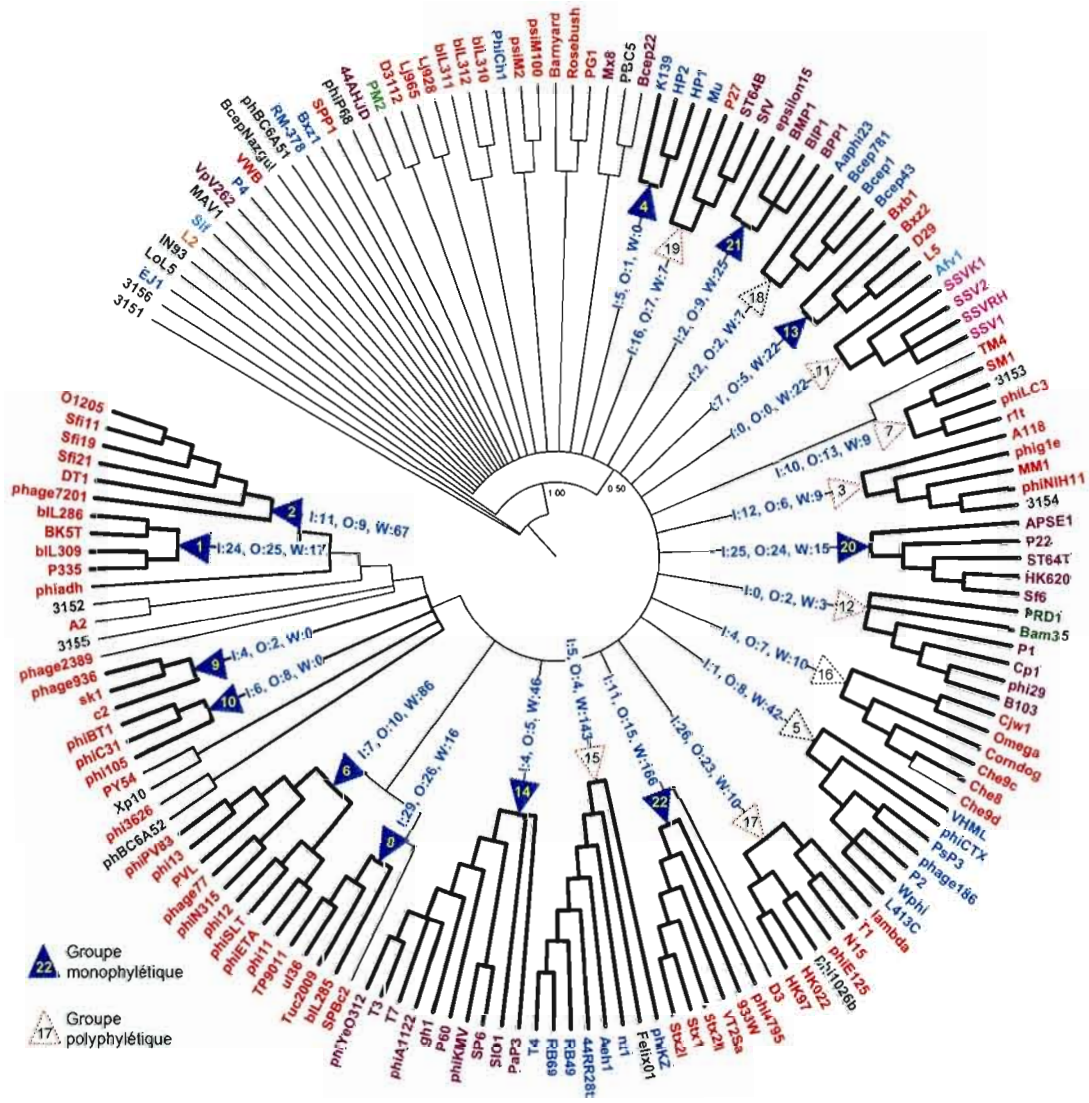


Figure 5.2 : Arbre phylogénétique d'espèces inféré par MrBayes

Note : les 22 groupes sont les mêmes que ceux de la figure 5.1.
Cette représentation sphérique est plus lisible que celle de la figure 5.1.

Les groupes sont représentés par des triangles creux et pleins ainsi que des traits en gras. Globalement, l'arbre phylogénétique d'espèces incorpore un grand nombre de signaux phylogénétiques : au total, 116 sur 163 phages, c'est-à-dire 71% des génomes étudiés, ont été classés dans 22 groupes avec des scores de probabilités *a posteriori* supérieur à 50%. Ces

groupes robustes (la plupart ont un score proche de 100%) contiennent entre 3 et 10 phages, avec une taille moyenne de clades de 6 phages. Certains groupes sont *monophylétiques**, d'autres *polyphylétiques*.

Plusieurs groupes monophylétiques, 12 (triangles pleins) sur 22, référencés par NCBI/ICTV[27] ont été retrouvés par notre analyse : *Siphoviridae* (groupes 1, 2, 6, 8, 9, 10, 13, 22), *Podoviridae* (groupes 14, 20, 21) et *Myoviridae* (groupe 4). Parmi les 10 autres groupes restant (triangles creux), 7 groupes en pointillés rouges sont composés de deux (groupes 3, 7, 11, 12, 15 et 17) voire trois familles de phages différentes (groupe 19). Ce sont des groupes polyphylétiques. Cela démontre-t-il l'existence de regroupements d'espèces qui n'ont pas été identifiés par les classifications classiques (e.g., NCBI/ICTV) ? C'est peut-être une piste à explorer dans une étape subséquente.

Cependant, plusieurs clades ou espèces seules demeurent non résolus. Cela est dû à l'absence d'information convergente au niveau du contenu en gène (i.e., VOG), traduit par la présence de différentes topologies associées à ces organismes parmi les arbres générés par MrBayes. Par ailleurs, on constate que la plupart des phages seuls ou appartenant à des petits clades font partie des espèces partiellement annotées (*unclassified*). Étant donné que cette étude a été basée sur les VOG lesquels ont été eux-mêmes formés à partir des gènes de phages (section 4.1.6.2), notre classification est dépendante du degré d'annotation des phages : plus celui-ci est conséquent plus la classification résultante est précise.

5.2 REPRÉSENTATION DES THG INTER ET INTRA GROUPES DANS L'ARBRE D'ESPÈCES

L'interprétation des THG revient à analyser manuellement les statistiques de transferts générées. La grande difficulté dans la détection de l'évolution réticulée, dont les THG font partie, vient du fait que n'importe quel signal contradictoire dans les données testées peut suggérer une réticulation. En effet, la non-additivité des distances peut s'expliquer par plusieurs facteurs, tels que par exemple, l'approche de regroupement des VOG (section 4.1.6.2), ou, plus en amont, le problème de l'échantillonnage des données, voire aussi le séquençage des données intrinsèquement bruitées à la source, etc.

---

[27] Possibilité de vérifier avec le nom long des phages (Annexe C) aux adresses suivantes :
NBCI : http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=35237&filter=genome_filter&p=7,
ou ICTV : http://www.ncbi.nlm.nih.gov/ICTVdb/Ictv/index.htm.

Aussi, à défaut d'avoir une très grande précision dans la détection, l'objectif est de chercher à confirmer globalement les THG par des indications de principaux transferts horizontaux de gènes. De ces indications, il serait ensuite possible d'analyser un peu plus en détails les transferts THG avec l'aide des biologistes, notamment, du point de vue biologique et physiologique du problème. Notre ambition dans cette étude est de proposer les premiers éléments de réponse quant aux transferts possibles des THG, et de là suggérer des perspectives de recherche (voir la section *Perspectives* plus loin).

Cette partie présente les statistiques des transferts THG inter et intra-groupes et contient une représentation de ces mêmes statistiques sur l'arbre d'espèces.

5.2.1 STATISTIQUES DES TRANSFERTS THG

5.2.1.1 Statistiques globales

Au total, 1451 transferts THG ont été détectés pour les 163 phages étudiés (tableau 5.1). Ce nombre inclut les transferts entre les groupes (transferts inter : 211), à l'intérieur de chacun des groupes (transferts intra : 722) et à partir ou vers les phages qui n'appartiennent à aucun groupe (transferts hors : 518).

	Transferts THG	Nb de phages (%)
Inter groupes (I/O)	211	116 (71%)
Intra groupes (W)	722	
Hors groupes	518	47 (29%)
Total	1451	163 (100%)

Tableau 5.1 : Total des transferts inter, intra et hors groupes

Ces chiffres montrent que les transferts entre les espèces appartenant au même groupe (intra) sont supérieurs de deux tiers par rapport aux transferts entre les espèces de groupes

différents (inter). Ce qui semble être logique, proportionnellement parlant, au regard de l'évolution classique et de l'évolution réticulée (sections 1.1.1.3 et 4.1).

Notons qu'environ le tiers des phages (29%) appartiennent à des hors groupes. Les transferts hors groupes (i.e., entre les hors groupes mais aussi entre les hors groupes avec les autres groupes identifiés) comparés aux inter et intra groupes sont quand même très importants (518). Cela se traduit-il par l'une des origines de la difficulté d'annotation (*unclassified*) discutée précédemment ? Il s'agit sans doute là, si ce n'est pas déjà fait, d'une piste d'investigation de plus pour un travail subséquent d'annotation.

5.2.1.2 Statistiques détaillées

Le tableau 5.2 détaille les statistiques des transferts THG. Ces dernières présentent plusieurs points remarquables :

Taille	Groupe	1	10	11	12	13	14	15	16	17	18	19	2	20	21	22	3	4	5	6	7	8	9	Out
4	1	17	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	2	19	1	25
3	10	0	0	0	0	2	0	0	0	2	0	1	1	0	0	0	0	0	0	0	0	1	1	8
5	11	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	12	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2
4	13	0	0	0	0	22	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
10	14	0	0	0	0	0	46	3	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	5
8	15	0	0	0	0	1	0	143	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	4
5	16	0	1	0	0	4	2	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
7	17	0	0	0	0	0	0	1	0	10	1	3	0	12	0	6	0	0	0	0	0	0	0	23
4	18	0	0	0	0	0	0	0	0	1	7	0	0	0	0	0	0	0	0	0	0	1	0	2
4	19	0	2	0	0	0	0	0	0	1	0	7	0	2	0	2	0	0	0	0	0	0	0	7
6	2	3	0	0	0	0	0	0	0	0	0	0	67	0	0	0	3	0	0	2	0	1	0	9
5	20	0	0	0	0	0	0	1	0	15	0	3	0	15	2	3	0	0	0	0	0	0	0	24
4	21	0	1	0	0	0	0	0	1	0	0	2	1	4	25	0	0	0	0	0	0	0	0	9
6	22	0	0	0	0	0	0	0	0	3	0	5	0	7	0	166	0	0	0	0	0	0	0	15
5	3	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	9	0	0	3	0	0	0	6
3	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
7	5	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	5	42	0	0	0	0	8
9	6	2	1	0	0	0	0	0	0	0	0	0	2	0	0	0	4	0	0	86	1	0	0	10
4	7	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	9	5	0	13
4	8	14	1	0	0	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	6	16	1	26
3	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	2
	In	24	6	0	0	7	4	5	4	26	2	16	11	25	2	11	12	5	1	7	10	29	4	

Tableau 5.2 : Statistiques de transferts inter et intra-groupes

[1] La ligne *Groupe 1* se lit 25 transferts extrants (notés *Out*) de ce groupe vers les autres groupes. La colonne *Groupe 1* se lit 24 transferts intrants (notés *In*) des autres groupes vers ce groupe ;

[2] Il n'y a pas de transferts inter-groupes (ligne et colonne 11) ni d'intra-groupes (diagonales 10, 4 et 9) ;

[3] Il n'y a pas de transferts inter-groupes qui soient supérieurs ou égaux aux transferts intra-groupes (groupes 1, 10, 17, 19, 20, 3, 4, 7, 8 et 9) ;

[4] Les groupes 1, 5, 6, 7, 10, 12, 14, 16, 21 et 22 en donnent plus qu'ils en reçoivent, et inversement pour le reste ;

[5] Les groupes qui en donnent beaucoup plus que la moyenne sont : le groupe 1 au groupe 8 (19 transferts), le groupe 17 au groupe 20 (12 transferts), le groupe 20 au groupe 17 (15 transferts) et le groupe 8 au groupe 1 (14 transferts) ;

[6] On constate que, plus la taille d'un groupe est grande (voir colonne Taille), plus il y a de transferts intra-groupes (groupes 15, 22 et 6), exceptés les groupes 12 et 14.

5.2.2 Représentation des THG

Pour chaque groupe, I (*In*) désigne le nombre de THG entrant dans le groupe, O (*Out*) le nombre de THG sortant du groupe et W (*Within*) le nombre de THG à l'intérieur de ce groupe.

La figure 5.3 a été dessinée à l'aide de l'outil de représentation d'arbres *iTol* (disponible sur : http://itol.embl.de) [Letunic & Brok 2007].

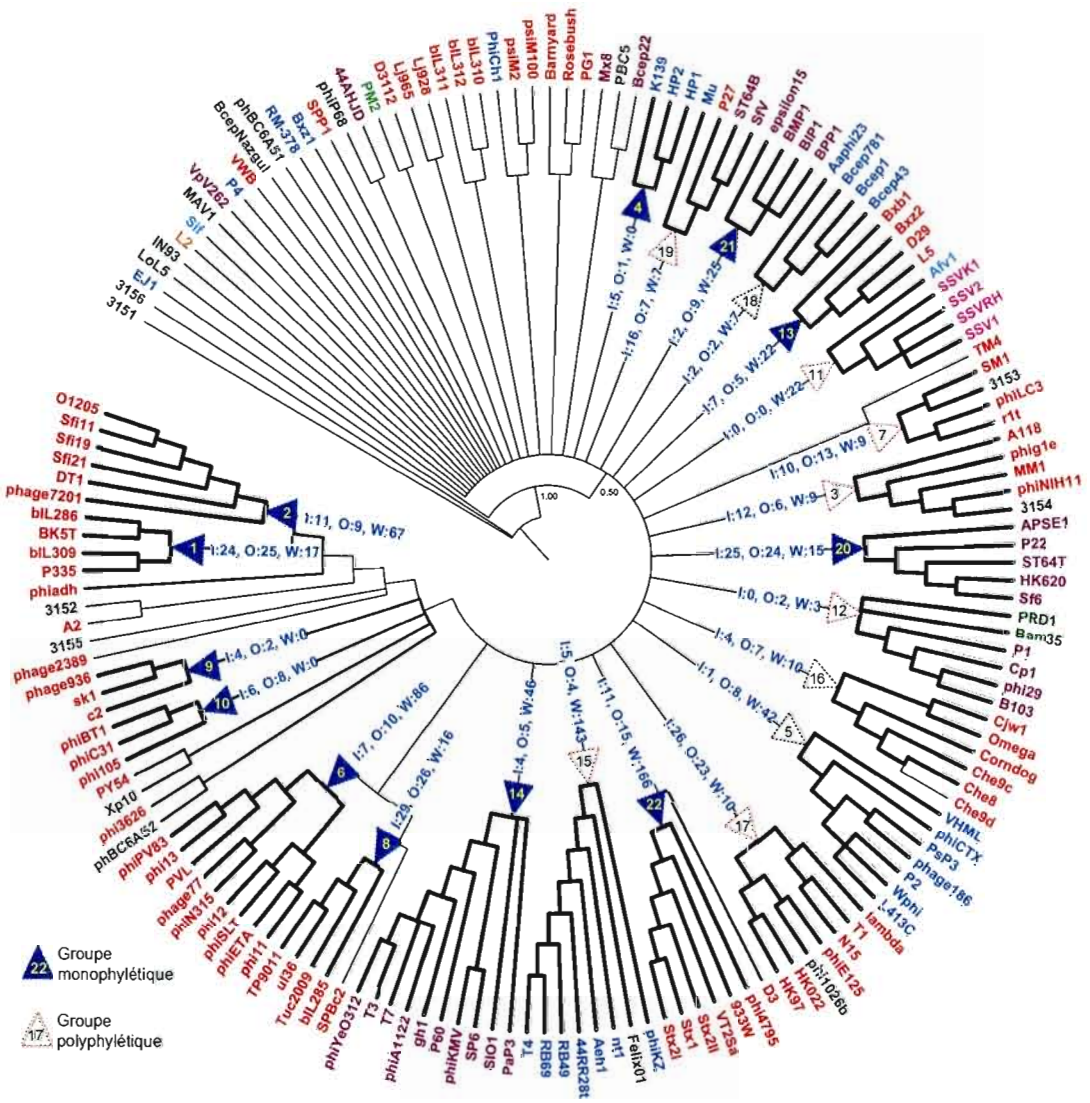


Figure 5.3 : Statistiques des transferts horizontaux inter et intra-groupes

La figure 5.3 montre les principales statistiques In/Out/Within, reportées sur l'arbre d'espèces de MrBayes.

5.3 REPRÉSENTATION DES SÉQUENCES ANCESTRALES DANS L'ARBRE D'ESPÈCES

On montre dans un premier temps, une représentation générale où apparaissent les nœuds d'ancêtres communs les plus proches et les séquences de protéines ancestrales. On se focalise, ensuite, sur le cas des phages *L. Lactis* avec une représentation partielle de l'arbre d'espèces.

5.3.1 REPRÉSENTATION GÉNÉRALE

Les résultats de la reconstruction des séquences protéiques ancestrales sont présentés sous forme d'arbres et de tableaux. Ainsi, nous déterminons pour chaque VOG, sa ou ses séquences de protéines ancestrales et le nœud ancestral ACP correspondant dans l'arbre d'espèces.

Ce travail permet d'identifier, à des fins de comparaison de génomes, l'ensemble des fonctions assignées à chaque nœud ancestral de l'arbre d'espèces.

5.3.1.1 Nœuds d'ancêtres communs les plus proches

La figure 5.4 montre les nœuds internes qui représentent les ancêtres communs les plus proches (ACP). Seuls 47 (en rouge) des 114 nœuds internes sont identifiés comme des nœuds d'ancêtres des phages étudiés. Ces 47 nœuds ACP ont été reportés comme des nœuds ancestraux des 602 VOG, c'est-à-dire des 602 fonctions protéiques annotées (car un VOG est associé à une fonction, voir la section 4.1.6.2).

La figure 5.4 a été dessinée à l'aide de l'outil de représentation d'arbres *iTol* (disponible sur : http://itol.embl.de) [Letunic & Brok 2007].

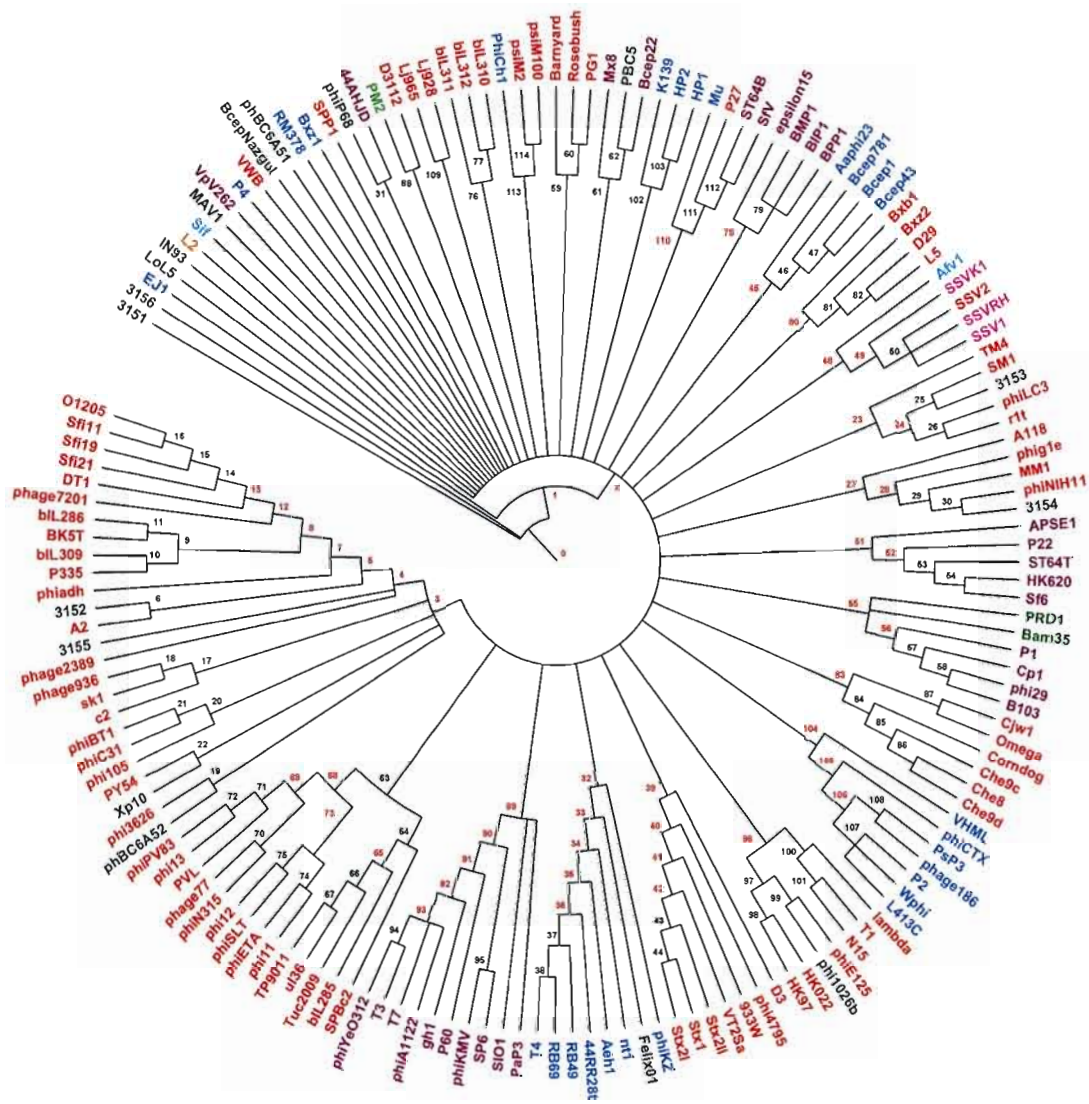


Figure 5.4 : Représentation générale des nœuds internes ACP sur l'arbre d'espèces de MrBayes

5.3.1.2 Séquences de protéines ancestrales

Le tableau 5.3 montre un extrait de la liste des nœuds internes ACP (la liste complète est donnée en Annexe E). Chaque nœud est associé à une ou plusieurs fonctions protéiques et autant de séquences ancestrales correspondantes.

Nd Int.	Fonction	VOG	Séquence ancestrale
27	[S] Minor capsid protein	vogp0029	LIQQIKDLVRSGANYIGITQSLTNITDHPKITINQEEYDRJNTNLDYYQSK[illegible] TKKRQLNTINAKTAAKNIASLVFNEKAKINVKDNAANEFFSDILKNNRFNKNFERYLEYCL ALGGLAMRPYIDGNKNNVAFVQAHLFYPLQSNTQDVSZAAIATNSHKTKNYYTLLEFHQ WQASDYVITNELYQSDYPDKVGTQVPLSKLFKDMEVDAQVTDFTRPIFTYTKSNSTNNKDI NSPLGIN
27	[U] Uncharaterized	vogp0187	MLSLAYPLDNTFKFGGKEYNMDLSFNKVLHVFNLMEDDSLTDSYQAYLAFDILLGQDMN NIEETAYFLIYIITNFIDKEHQDSFRYDIEGNPMPLANNKEEQENFFSLTQDADYIYASFLQD YHINLLEYQGKLPWNKFKALLDALPDNTIIQRIIAIRQCESPCGEGEKERNNLFKLKDHYYL DDQDEEDCWSSQ
28	[S] Minor capsid protein	vogp1149	MRLLTAMDKCLLTDILTIKKVTDKDDZGHLVYCEPFTIKHFRFHPHIVSSGNNNSHTGTSNN LVFIYFYPSYSLVTVNNSWVGSKVVYGGREYTIHKIITNYHPFSNEIFSZEMEVI
28	[S] Minor capsid protein	vogp0509	MNYFENFVSQLIKVSNLPIKPRLDYLTDHDDLAIYPMPGCKVNDEYMDGTQEVSLPFEIAIK TKNQQLANTTMWLVTSALANFNSDNPSSNNSYKFMSLDVNSNSMKDQDNQGYYTYILDI TANIDIZGNNQ
3	[T] Putative DNA binding protein	vogp0362	EIMNRIKQLRKSRKMTRVELAEKIGVTKCTILNWEQCTSYTNPHNSQRKLADFFDVSVPYL LGZDTNZTYSNNEMALKDAIGDSIVTLVVLCLQLGYDVEECLKIAYNNIKDRQGVMN
3	[U] Uncharacterized	vogp0126	MFNIDHSQAKDFGSIKDGTYEVIIDNANQDATKNGAEFIDIHFRIRKDFQQEFQNNNIFHRIW NDKDANKYPMAAFNNIAKAAGFPNGTKFNSLEDZLNHLLNKAFQVTVKNEKSEYKGKTY KNLNVKALAESNIPCNANPVEISEEDLPFF
32	[L] dTMP (thymidylate) synthase	vogp0801	MKQYSNLFRDILDNGYHEEDRTGIGTFSMFGPKLRWDSREGFPFVTTEKMASKSVIGESLW FTSGSTEINHSHEIAHGTHNDFCDZEHDVWESNYKDKEVNNNVGYTNDDLGHMYSKHWY NRNIHPVYKFIIDIDHVNANPTYHNLMVKAWNPYNENEDHVALAPCHFFFQVFVSKQGRFS FEWYKDSEVNFLGLAFDVRSYRLSVHVMSKRSSLEVSNLVFSGSNLDIYNNRVIQVYEPFN HEHRQFAYFANIY
32	[L] Frd dihydrofolate reductase	vogp0802	MMIKSVFASGKSSTFHKNGKLAFGNKNGLPWGHIHEDMLNFKETTKDSFLVMGTKTFKSL PNNLPNRINIVLSTSNHTYRINAKNNZGQRPNIYMHCHFTHSSSKLQNSYNNISVIGGLTML KEALHLADQVFHTIILKATEEDTFDSDIQLSKNFLQNIYFDFYVMSNHYFGNEAISYTSIHKN KKQF
etc...	etc...	etc...	etc...

La liste exhaustive des 47 nœuds internes ACP et les 602 séquences protéiques ancestrales associées est donnée en Annexe E.

Tableau 5.3 : Liste de nœuds ACP avec fonctions et séquences ancestrales associées (extrait)

Par exemple, au nœud interne ACP 27 (tableau 5.3), deux séquences ancestrales (''LIQQIKDLVRSGANYIG...'' et ''MLSLAYPLDNTFKFGGK...'' ont été générées à partir des VOG0029 et VOG0187 qui correspondent à des fonctions : *[S] Minor capsid protein* et *[U] Uncharaterized.*

5.3.1.3 Fonctions protéiques identifiées et inconnues

Parmi les 602 VOG équivalants à 602 variantes de 6 types de protéines, 55% des protéines de phages sont identifées au niveau de leurs fonctions moléculaires (*Auxiliary Proteins*, *DNA Replication*, *Regulation of Cellular Metabolism*, *Structural Proteins*, *RNA Replication*). Les 45% restant sont encore à découvrir (*Unknown Proteins*). Voir figure 5.5.

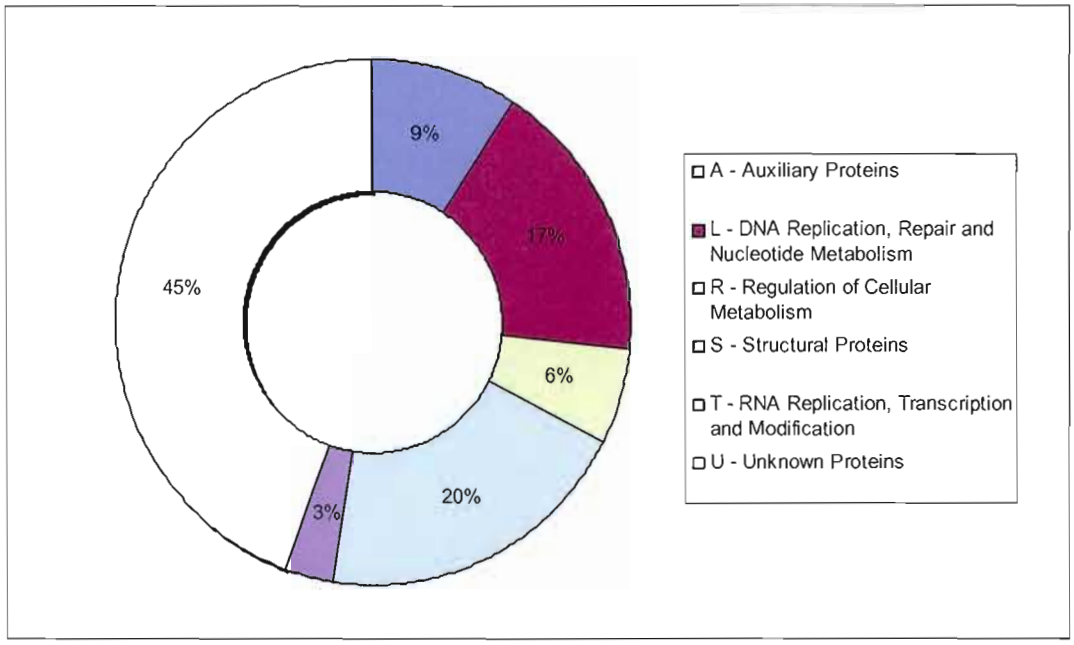


Figure 5.5 : Répartition de fonctions protéiques identifiées et inconnues
(Les informations sur les fonctions protéiques sont données en Annexe C).

La reconstruction ancestrale de séquences protéiques peut être sinon un des moyens pour découvrir au moins pour apporter des informations complémentaires relatives aux rôles des 45% de protéines inconnues recensées dans notre étude.

5.3.2 REPRÉSENTATION PARTIELLE – CAS DES PHAGES *L. LACTIS*

Nous nous sommes intéressés ici en particulier au cas des phages affectant la famille des bactéries *Lactococcal Lactis* afin de présenter un exemple typique de reconstruction ancestrale ayant des impacts concrets dans la pratique.

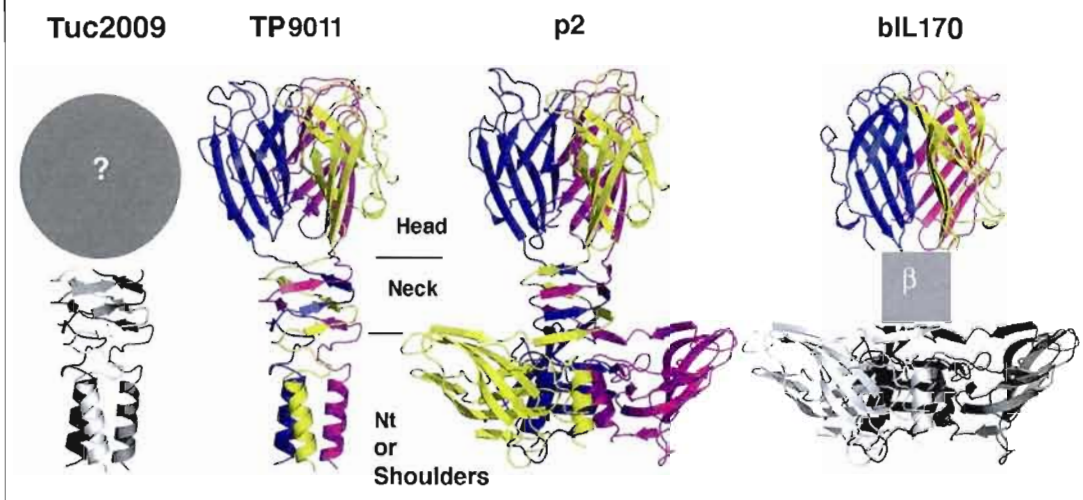
5.3.2.1 Protéine RBP des phages *L. Lactis*

Tous les phages connus infectant les bactéries *Lactococcal lactis* ont un génome à double brin et une queue non contractile [Deveau et al. 2006]. Suivant la taxonomie ICTV, les phages infectant les bactéries *L. Lactis* sont membres de l'ordre des *Caudovirales* (bactériophages à queue), un très grand groupe, morphologiquement et génétiquement varié qui comprend plus de 95% de bactériophages connus [Maniloff & Ackermann 1998]. Cet ordre regroupe trois familles : *Myoviridae* (avec des longues queues contractiles),

Siphoviridae (avec des longues queues non contractiles) et *Podoviridae* (avec des queues courtes). Les phages infectant les *L. Lactis* sont principalement membres de la famille *Siphoviridae* incluant quelques individus de la famille *Podoviridae*.

Or, des analyses comparatives récentes [Spinelli et al. 2005, 2006 ; Ricagno et al. 2006; Deveau et al. 2006] semblent remettre en cause la taxonomie établie. En effet, Spinelli et al. [2005] ont montré que la protéine RBP (*Receptor Binding Protein*, lire par ailleurs, la section 4.1.4, page 84 "*Taxonomie existante des bactériophages*") comprend trois sections : des épaules, un cou et une tête. Semblables à des blocs Lego, certaines espèces, comme le virus *phage p2*, a une structure au niveau de leur tête proche de certains virus humains, alors que d'autres, ont plutôt des épaules ou le cou semblables. Il semble que la protéine RBP soit de nature modulaire et que les différentes sections soient interchangeables selon les espèces.

Dans la même veine des expériences, Ricagno et al. [2006] ont analysé plusieurs modèles atomiques (ou repliement 3D) de la protéine RBP (figure 5.6) au sein de la famille des phages infectant les *L. Lactis*. Ainsi, la comparaison structurale de la protéine RBP des phages *phage p2* et *biL170* (représentés en rectangle de traits pleins sur l'arbre de la figure 5.7a), les sous-espèces de *phage936*, et des phages *TP9011* et *Tuc2009* (rectangles de traits en pointillés sur la figure 5.7a) suggère une très grande modularité des RBP des phages infectant les *L. Lactis*. Bien que les deux groupes d'espèces comparés soient éloignés, les phages sont, semble-t-il, plus proches au niveau des séquences et probablement au niveau de la structure que ne l'aient les phages *phage p2* versus *biL170*, et *TP9011* versus *Tuc2009* [Spinelli et al. 2006].



Comparaison des modèles atomiques RBP de quatre phages : *phage p2* (ou simplement *p2* sur la figure ci-dessus) et *biL170*, des sous espèces de *phage 936*, avec une autre espèce, les phages *TP9011* et *Tuc2009*. Les protéines RBP sont représentées sur la même échelle, et leurs trois domaines, le N-terminal (ou les épaules dans le cas de *p2*), le cou et la tête, sont notés respectivement *Nt*, *Neck* et *Head*.

Les structures déterminées expérimentalement sont représentées en chaînes de rubans de couleur bleue, rose et jaune. Les structures modélisées sont représentées en chaînes de rubans grises. Le carré et le cercle gris sont des structures qui restent inconnues.

Figure 5.6 : Comparaison structurale de la protéine RBP de quatre phages *L. Lactis*
(Tirée de Ricagno et al. 2006)

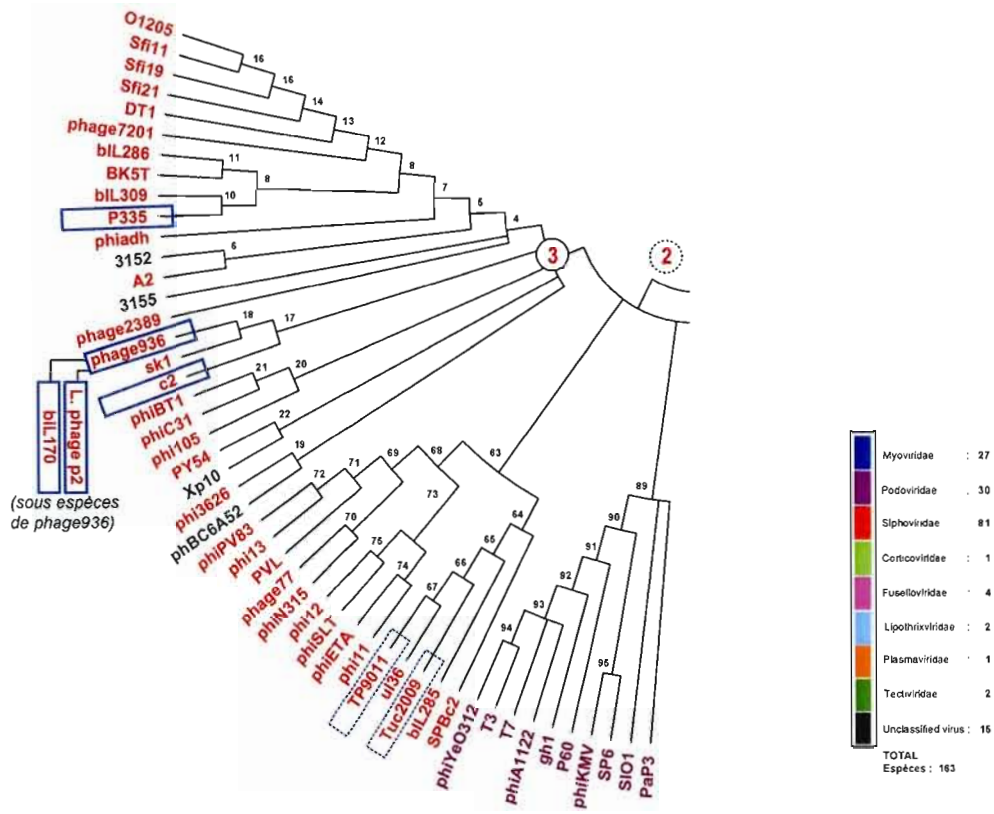
5.3.2.2 Cas des noeuds ACP 2 et 3

Puisque la présente étude est basée sur les données VOG résultantes elles-mêmes de la classification de référence ICTV, il est donc naturel que nous trouvions un nœud ACP (i.e., le nœud 3 ici) qui soit « proche » des phages reconnus de longue date (phages *phage936* et sa sous-espèce *L. phage p2*, *c2*, et *p335,* Cf. classification ICTV) versus ceux mis en évidence récemment (*TP9011* et *Tuc2099* [Ricagno et al. 2006 ; Spinelli et al. 2006]).

La figure 28a montre un exemple du nœud 3 représentant le nœud d'ancêtre commun le plus proche, entre autres, des phages *phage936* et ses sous espèces *L. phage p2* et *biL170*, *c2*, et *p335*. A ce nœud interne est associé deux séquences protéiques ancestrales

(''EIMNRIKQLRKSRKMTRVELAE…''et ''MFNIDHSQAKDFGSIKDGTYEVIIDN…'') générées à partir des VOG0362 et VOG0126 qui correspondent aux deux fonctions : *[T] Putative DNA binding protein* et *[U] Uncharaterized* (figure 5.7b).

a)



b)

Nœud interne	Fonction	VOG	AncSeq
3	[T] Putative DNA binding protein	vogp0362	EIMNRIKQLRKSRKMTRVELAEKKIVTKCTLNWEQCTSYTNPHNSQRKLADFFDVSVPYLLGIZDTNZTYSANNEMALADAIGDSIVTLVVLCLGLGYDVEECLRIAYNNKDRQQ VMN
3	[U] Uncharacterized	vogp0126	MFNIDHSQAKDFGSIKDGTYEVIIDNANQDATKNGAEFIDIHFRIRKDFQQEFQNNNIFHRIW NDKDANKYPMAAFNNIAKAAGFPNGTKFNSLEDZLNHLLNKAFQVTVKNEKS EYKGKTYKNLNVKALAESNIPQNANPVEISEEDLPFF

Figure 5.7 : Cas des nœuds ACP 2 et 3 (sous-arbre d'espèce)

Ainsi, si on veut reconsidérer l'ancêtre le plus proche de tous les phages mentionnés, c'est-à-dire en incluant par exemple, aux premiers phages cités, les phages *TP9011* et *Tuc2009*, il faut « remonter » au nœud ACP 2 de l'arbre d'espèces (figure 5.7a).

Finalement, nos expériences de reconstruction révèlent la position et la composition de séquences protéiques ancestrales. À l'instar des phages infectant les *L. Lactis* discutés ici, les biologistes moléculaires pourraient maintenant étudier en détail la nature et les fonctions des séquences ancestrales générées. Ainsi, au regard des travaux de Spinelli et al. [2006], notre approche de classification en générale et de génération de séquences ancestrales en particulier, pourrait effectivement apporter des éléments d'information complémentaires susceptibles d'améliorer la taxonomie existante des phages (section 4.1.4).

CONCLUSION

CONCLUSION

Le thème de la classification était notre sujet principal tout le long de la présente étude. Elle est appelée catégorisation humaine lorsqu'elle est abordée sous l'angle de la psychologie cognitive ou classification machine quand on l'employait en termes de traitement de l'information. Dans le cadre de l'analyse phylogénétique, les différentes approches méthodologiques de la classification machine étaient utilisées pour classer et regrouper les données biologiques virales. Ainsi donc, comme conclusion de l'étude, nous retiendrons les principaux points suivants :

- Processus de catégorisation exprimés selon les différents modèles : nous avons vu, au chapitre II différents modèles de catégorisation qui s'agit de modèle des exemplaires versus modèle des prototypes ou de modèle basé sur la similarité versus modèle basé sur l'inférence, tous suggèrent des approches complémentaires dont le but est de comprendre les différentes facettes de la catégorisation.

- Transposition entre les sciences cognitives et les sciences de traitement de l'information : on peut raisonnablement dire, suite aux discussions menées aux chapitres II et III, que les modèles de catégorisation humaine et les méthodes de classification machine sont équivalents, du moins à haut niveau.

- Apport mutuel entre les sciences cognitives et la phylogénie : grâce aux sciences cognitives, nous avons appris que l'inférence bayésienne (e.g., modèle rationnel) peut être une approche alternative à l'approche de distances (e.g., modèle des exemplaires) dans la catégorisation des objets. Par transposition, nous avons fait un choix éclairé concernant la méthode d'estimation bayésienne (e.g., MrBayes) comme alternative à la méthode de distances (e.g., NJ) dans la recherche de méthodes donnant les meilleurs résultats de classification (section 4.2.2.6 et chapitre V). En retour, grâce à l'intérêt croissant en phylogénie pour les méthodes bayésiennes, soutenu par les récents travaux

[Huelsenbeck & Ronquist 2001 ; Larget & Simon 1999 ; Blanquart & Lartillot 2006], sans oublier la présente étude, nous informons les sciences cognitives que l'approche bayésienne pourrait jouer un rôle grandissant et réellement pertinent dans la compréhension des processus de catégorisation humaine [Rehder 2003 ; Rehder & Burnett 2005].

- Concurrence entre les méthodes de distances et les méthodes d'inférence bayésienne : pour reconstruire un arbre phylogénétique, on a utilisé la méthode de distances (i.e., Neighbor-Joining) et la méthode concurrente d'inférence bayésienne (i.e., MrBayes) afin de valider les résultats et par là même, de conforter les arbres obtenus.

- Complémentarité entre les méthodes de distances et les méthodes d'inférence bayésienne : c'est grâce au principe d'optimisation de distances de Robinson et Foulds (implémenté dans HGT-Detection) que la détection des transferts THG a pu être effectuée, et c'est à travers notamment du principe du maximum de vraisemblance utilisé avec le modèle Tree-HMM (implémenté dans Ancestor) que les séquences de protéines ancestrales ont pu être générées. Mais, c'est avec la complémentarité des deux principes que l'on a entrepris l'analyse de l'histoire évolutive des bactériophages.

- Approche originale de la classification des bactériophages : la méthode de détection de transferts THG a été combinée avec la méthode de reconstruction de séquences ancestrales afin de proposer une approche originale de la classification des bactériophages. Par rapport à la taxonomie existante, les résultats obtenus (chapitre V) ont apporté des informations additionnelles visant à donner un éclairage sur la très complexe histoire évolutive des phages. L'issue de cette étude a permis effectivement de [1] fournir une classification des bactériophages en tenant compte des hypothèses de l'évolution classique et de l'évolution réticulée, [2] fournir des statistiques sur les différents transferts horizontaux inter et intra-groupes, [3] générer des séquences de protéines ancestrales des phages et identifier leur origine avec les nœuds ACP (ancêtres communs les plus proches) dans l'arbre d'espèces. Nous n'avons toutefois pas approfondi le point sur la comparaison structurale entre les protéines qui dépasse le cadre de la présente étude.

À noter par ailleurs, que tout le travail d'analyse et de modélisation repose sur l'état de connaissances actuelles qui représente selon certains 1%[28] de la biodiversité microbienne [Pace 1997]. Il ne fait donc aucun doute qu'une meilleure connaissance (au niveau génomique en particulier) de cette extraordinaire diversité pourrait apporter des éclairages nouveaux sur les questions qui ont été abordées dans cette étude.

- Complémentarité de classifications : notre objectif a été de générer à partir des données observées, une classification des phages visant à reconstruire un arbre d'espèces original tout en révélant des indications sur les signaux phylogénétiques de types transferts THG et séquences ancestrales. Cette démarche complète celle de la classification traditionnelle arborescente proposée par ICTV où les espèces sont discriminées de manière manuelle et très détaillée, en donnant des informations précises sur la nature et le genre des bactériophages : taille, morphologie, hôtes infectés, etc.

- Points en suspens : nous avons occulté cependant, plusieurs problèmes dont ceux liés aux autres mécanismes de l'évolution réticulée (e.g., l'évolution convergente, la duplication d'un gène suivie de sa perte, l'hybridation), l'exactitude des alignements obtenus ainsi que les scénarios de reconstructions ancestrales alternatifs. Pour autant, nous estimons raisonnablement que la présente étude peut contribuer sinon à une meilleure compréhension des phages, du moins à progresser vers une étape importante d'interprétation de la phylogénie de ces microorganismes.

[28] Ce chiffre a très certainement progressé depuis ce qu'a avancé Pace [1997].

PERSPECTIVES

A l'issu de cette étude, nous voyons s'ouvrir des perspectives exploratoires très intéressantes pour ceux qui voudront s'y intéresser :

- Vers une théorie de catégorisation unifiée : certains auteurs, comme Griffiths et al. [2008], travaillent à faire converger les théories basées sur la similarité et les théories basées sur l'inférence causale. D'autres auteurs, tels que Danks [2007], avancent l'idée qu'il est possible de mettre les différentes théories de catégorisation sous une théorie unifiée : "*Theory Unification and Graphical Models in Human Categorization*". Ces champs exploratoires suggérés devraient toutefois être confirmés par d'autres travaux à venir.

- Investigation au niveau des groupes polyphylétiques et de transferts THG : au moins deux pistes d'investigation qui mériteraient d'être étudiées un peu plus en détail. D'une part, des groupes polyphylétiques qui ne sont pas reconnus par ICTV comportent plusieurs espèces différentes dans le même regroupement (section 5.1.2), et, d'autre part, en plus des transferts THG inter et intra groupes, d'autres transferts hors groupes peuvent être porteurs de signaux phylogénétiques intéressants à considérer (section 5.2.1.1).

  De manière générale, nous pensons que notre approche de détection des transferts THG pourrait jouer, au même titre que les critères de morphologie ou d'homologie d'ADN, un rôle de critère discriminant pour ceux qui cherchent à classer les espèces virales.

- Suggestion d'analyse de séquences ancestrales : au regard des expériences de Spinelli et al. [2005, 2006] et Ricagno et al. [2006] par exemple, qui se sont interrogés sur une origine ancestrale commune probable de la protéine RBP phylogénétiquement différente des phages infectant les bactéries du lait *L. Lactis* (section 5.3.2), notre approche a permis de générer des séquences de fonctions protéiques ancestrales qui pourraient être examinées plus en détail par les biologistes (section 5.3.2.2, figure 5.7b et Annexe E).

  De manière générale, nous pensons que les séquences ancestrales générées pourraient jouer un rôle, grâce aux séquences représentatives de groupes de phages, pour ceux qui

analysent la similitude structurale des protéines. Nos résultats ont permis effectivement de réduire la complexité des analyses. Les séquences protéiques ancestrales et leur probabilité *a posteriori* au niveau de chaque caractère ont été prédites. Ces protéines ancestrales pourraient donc servir comme représentants de familles de bactériophages lors de différentes analyses génomiques comparatives.

- Connaissance sur les virus : notre aperçu de la biodiversité virale demeure parcellaire et très biaisé [Pace 1997 ; Forterre et al. 2002]. Malgré les connaissances acquises depuis de nombreuses années en biologie, en biochimie et en génétique, les interactions impliquées dans la machinerie bactérienne restent encore inconnues [Galus 2006]. Il n'est donc pas impossible que nos connaissances sur l'évolution de ce domaine soient largement remises en cause dans les années qui viennent. Mais, chose certaine, le séquençage d'un beaucoup plus grand nombre de génomes viraux à venir (notamment l'annotation des virus *unclassified*, voir la section 5.1.2) nous permettra de reconstruire pas à pas l'histoire évolutive des virus, en reliant entre eux des virus de plus en plus divergents.

- Enfin, la plate-forme informatique : cette plate-forme pourrait prendre deux directions possibles comme environnement généralisé de classification (phylogénétique ou non) et comme environnement d'évaluation d'hypothèses informatiques cognitives.

  Dans le premier cas, la présente plate-forme développée pour l'étude des bactériophages pourrait être généralisée, grâce à son approche modulaire, à d'autres méthodes de classification par ajout de programmes disponibles dans la communauté scientifique, notamment les méthodes du maximum de parcimonie [Farris 1970 ; Fitch 1971] et les méthodes du maximum de vraisemblance [Felsenstein 1981]. La généralisation peut s'effectuer aussi au niveau des sujets traités. Ces sujets peuvent être reliés à la phylogénie [e.g. Delwiche & Palmer[29] 1996 ; Matte-Tailliez et al.[30] 2002] tout comme à d'autres problématiques, telles que la classification des langues indo-européennes [e.g. Gray &

[29] Les auteurs discutent de l'hypothèse du THG et de la duplication et perte du gène *rbcL* dans une phylogénie contenant des Protéobactéries, des Cyanobactéries et des plantes.

[30] Les auteurs discutent de l'évidence du transfert horizontal de gène du gène *rpl12e* entre le groupe des Thermoplasmatales et celui des crenarchaeota dans une phylogénie de 14 espèces d'Archaea.

Atkinson[31] 2003] et la classification textuelles [e.g. De Pascuale & Meunier[32] 2002]. Autrement dit, tous les sujets susceptibles d'adopter des scénarios réticulés comme évolution au sens large.

Quant à l'environnement d'évaluation d'hypothèses de type informatique cognitive, on peut imaginer en effet que les cogniticiens utiliserait cette plate-forme, à l'instar des chapitres II et III, comme un moyen concret de procéder à « l'opérationnalisation de l'informatique du problème de la classification et de la catégorisation » ayant comme objectif la recherche d'approche de modèle unifié de catégorisation [Griffiths et al. 2008 ; Danks 2007] (voir le premier point de la présente section).

[31] Les auteurs discutent de l'évolution des langues indo-européennes.

[32] Les auteurs discutent d'analyses thématiques et conceptuelles des textes.

CONTRIBUTION

MA CONTRIBUTION

Ma contribution dans cette étude s'est manifestée à plusieurs égards, notamment dans :

- La recherche sur NCBI, ICTV et autres banques de séquences génomiques, ainsi que la sélection des VOG dont le but était de recueillir les informations afin de constituer les données de base pour l'analyse phylogénétique des bactériophages.

Ce travail conjugué à l'accumulation de connaissances en biologie sur les phages est une des trois composantes nécessaires, avec les méthodes de détection de THG et de reconstruction de séquences ancestrales dont disposent l'équipe, qui ont permis de formuler la proposition originale de classification des espèces.

- La proposition et la programmation d'une plate-forme originale d'inférence phylogénétique permettant de tester et de valider de manière intégrée les hypothèses de transferts THG des phages et la possibilité de reconstruire les séquences de protéines ancestrales.

Cette plate-forme intègre judicieusement des approches et outils novateurs dont les programmes HGT-Detection et Ancestor pour inférer et analyser les différentes modes d'évolution, classique et réticulée, des bactériophages. À noter que la modularité de la plate-forme permet facilement d'ajouter ou de supprimer au besoin les outils d'analyse suivant l'approche ou l'hypothèse testée.

- La transposition de l'approche de catégorisation pratiquée en sciences cognitives en approche de classification employée en inférence phylogénétique. De cette transposition, il a été permis d'avancer que les deux approches présentent de fortes similitudes, et qu'entre les deux disciplines, il peut y avoir un apport enrichissant mutuel.

À ma connaissance, il n'existe pas de travaux semblables à ce jour, du moins dans l'effort de concilier l'approche de catégorisation des processus cognitifs à l'approche de classification machine.

NOTRE ÉQUIPE

Notre équipe de travail est composée des personnes suivantes :

- Vladimir Makarenkov, directeur de recherche et professeur à l'UQAM,
- Pierre Poirier, directeur de recherche et professeur à l'UQAM,
- Abdoulaye Baniré Diallo, étudiant au doctorat en informatique à McGill et, puis, professeur à l'UQAM.
- Alix Boc, étudiant au doctorat en informatique à l'UQAM.

GLOSSAIRE

ADN (acide désoxyribonucléique) : macromolécule constituée de deux chaînes enroulées en double hélice. Ses deux brins sont assemblés à partir de nucléotides. Chaque nucléotide comprend un sucre, le désoxyribose, un phosphate et une des quatre bases azotées (adénine, guanine, cytosine, thymine). L'ADN est le support de l'information génétique des organismes vivants.

ADN (et ARN) polymérase : enzyme qui polymérise les nucléotides. Les désoxyribonucléotides sont utilisés dans la synthèse de l'ADN. Les ARN polymérases, elles, polymérisent des ribonucléotides pour donner des ARN.

Alignement : opération qui consiste à disposer les unes en dessous des autres des portions de séquences similaires en minimisant leurs différences (on peut aligner entre eux des gènes d'une même famille multigénique, des gènes d'espèces différentes.). Si ces gènes sont homologues, les différences d'aminoacides ou d'acides nucléiques entre les séquences actuelles sont le témoignage de mutations qui ont eu lieu dans le passé.

Aminoacide (Acide aminé) : unité constitutive des protéines. Il existe 20 acides aminés communs : alanine, arginine, asparagine, aspartate, cystéine, glutamine, glycine, histidine, isoleucine, leucine, lysine, méthionine, phénylalanine, proline, glutamate, sérine, thréonine, tryptophane, tyrosine, valine.

Analogie : des caractères qui se ressemblent (ou similaires), mais non homologues, remplissent les mêmes fonctions biologiques. L'analogie est un cas particulier de l'homoplasie.

Archaea : les **Archées** ou ***Archaea*** (anciennement appelés **archéobactéries**, du grec *archaios*, « ancien » et *backterion*, « bâton ») sont un groupe majeur de microorganismes. Elles constituent un taxon du vivant caractérisé par des cellules sans noyau et se distinguant des Eubactéries (vraies bactéries) par certains caractères biochimiques, comme la constitution de la membrane cellulaire ou le mécanisme de réplication de l'ADN.

ARN (acide ribonucléique) : polymère linéaire dont la sous-unité de base, un ribonucléotide, contient le sucre ribose.

Bacteria : les bactéries appartiennent au vaste ensemble des microbes qui comprennent également les virus, les champignons et les parasites. Microorganismes invisibles à l'œil nu, les bactéries sont constituées d'une seule cellule dépourvue d'un vrai noyau. Elles contiennent un seul chromosome formé d'un long filament d'ADN.

Clade : vient du grec clados qui signifie branche. Taxon strictement monophylétique, c'est-à-dire contenant un ancêtre et tous ses descendants.

Chimère : individu composite constitué de cellules provenant de différents zygotes.

Delete-half-jackknifing : c'est une technique alternative à celle du bootstrap. Elle permet l'échantillonnage aléatoire de la moitié des caractères, et les ré-intégrer ensuite dans l'ensemble des données tout en éliminant les autres. L'ensemble des données résultant est de taille moitié moindre par rapport à l'original. La variation aléatoire obtenue par cette technique devrait être très similaire à celle obtenue avec le bootstrap. Cette technique est préconisée par [Wu 1986]. Définition donnée par Felsenstein (*http://cmgm.stanford.edu/phylip/seqboot.html*).

Eucarya : les **Eucaryotes** (du grec *eu*, vrai et *karuon*, noyau) comprennent 4 grands règnes du monde vivant : les animaux, champignons, les plantes et les protistes. Ils constituent donc un très large groupe d'organismes, uni et pluricellulaires, définis par leur structure cellulaire (noyau, ADN, cytosquelette, etc.).

Homologie : signifie que deux séquences (ou plus) ont un ancêtre commun. Deux structures sont dites homologues si elles ont été acquises par descendance d'un ancêtre commun possédant cette même structure. Les différences observées dans les descendants sont dues à la divergence génétique.

Homoplasie : similarité chez une ou plusieurs espèces, d'organes, de parties d'organes ou de séquences d'ADN ou de protéines, lorsque l'on peut présumer que cette correspondance ne provient pas de l'héritage d'un ancêtre commun. On distingue plusieurs cas d'homoplasies : l'analogie, la convergence, le parallélisme, la réversion. Remarque : Dans le cas des caractères moléculaires, l'homoplasie le plus souvent n'est pas détectable *a priori* et elle est révélée par l'arbre le plus parcimonieux. Dans le cas des caractères morphologiques, l'analyse fine des caractères et de leur homologie primaire permet plus souvent de détecter des homoplasies. Comme dans le cas précédent, celles qui n'auront pas été décelées seront révélées par l'arbre le plus parcimonieux.

Horloge moléculaire (hypothèse d') : hypothèse selon laquelle les molécules d'une même classe fonctionnelle évoluent régulièrement dans le temps et à un rythme égal dans différentes lignées. Ainsi la quantité des différences moléculaires constatées de nos jours dans des séquences homologues d'espèces distinctes peut être utilisée pour estimer le temps écoulé depuis le dernier ancêtre commun à ces deux espèces (ou temps de divergence).

Monophylétique (groupe) : c'est un groupe qui comprend une espèce ancestrale et tous ses descendants. On dit aussi d'un groupe monophylétique qu'il est un clade.

Orthologue : ce sont des gènes d'espèces différentes dont les séquences sont homologues, qui dérivent d'un même gène ancestral et ont divergé à la suite d'un évènement de spéciation (et non pas par duplication comme c'est le cas des gènes paralogues). Ils peuvent ou non

avoir la même fonction. Avec les gènes paralogues, ils forment des superfamilles de gènes homologues. Voir Paralogue.

Paralogue : ce sont des gènes d'une même espèce dont les séquences sont homologues et résultent de la duplication d'un même gène ancestral. Voir Orthologue.

Permutation : la permutation des espèces à l'intérieur des caractères. Cette méthode de ré-échantillonnage est introduite par [Archie 1989] et [Faith 1990]. Elle permet la permutation les colonnes de la matrice de données de manière séparée. Cela produit des matrices de données qui ont le même nombre et le même type de caractères mais sans la structure taxonomique. Elle est utilisée à d'autres fins que celles du bootstrap puisqu'elle ne teste pas la variation autour d'un arbre estimé mais l'hypothèse selon laquelle il n'y a pas de structure taxonomique dans les données : si une statistique telle que le nombre d'étapes est sensiblement plus petit dans les données réelles que celles dans les réplications qui sont permutées, alors on peut dire qu'il y a une certaine structure taxonomique dans les données. Définition donnée par Felsenstein (*http://cmgm.stanford.edu/phylip/seqboot.html*).

Plasmides : molécule d'ADN circulaire douée de réplication autonome et transmise de façon stable au cours des générations. Un plasmide porte de multiples gènes et fréquemment des gènes de résistance aux antibiotiques.

Polyphylétique (groupe) : c'est un groupe qui contient un certain nombre d'espèces ou de taxons, mais ne contient pas l'ancêtre commun à tous. En d'autres termes, un groupe polyphylétique dérive de deux ou plusieurs espèces ancestrales. Un groupe polyphylétique est défini par au moins une homoplasie.

Raciné (arbre) : arbre muni d'une racine. On dit plutôt *enraciné*.

Racine : segment de branche en amont du nœud du rang le plus important, définissant le groupe extérieur (voir Extragroupe). En d'autres termes, c'est la position dans l'arbre du groupe extérieur. En même temps, elle définit le taxon *ingroup* (voir ci-dessous) La racine peut être considérée comme un point de référence pour l'interprétation des caractères : les états de caractères de l'extragroupe (*outgroup*) sont des états plésiomorphes, les états qui en diffèrent sont apomorphes. Remarque : pour pouvoir comparer aisément deux arbres, il faut les enraciner chacun sur la même espèce ou sur le même taxon.

- **Extragroupe** (outgroup) : on dit aussi groupe extérieur ou encore "outgroup" tiré de l'anglais. Groupe que l'on sait a priori placer en dehors d'un ensemble de taxons dont on cherche les relations de parenté.
- **Ingroup** : Terme anglais désignant un ensemble de taxons dont on recherche les relations de parenté. Cet ensemble s'oppose à l'Extragroupe, groupe que l'on sait a priori placer à l'extérieur de l'ingroup.

Rétrotransposons : classe de transposons dont la transposition nécessite la transcription inverse de leur produit de transcription.

Transposons : séquence d'ADN qui présente la particularité de pouvoir se déplacer du chromosome vers un plasmide et d'un plasmide à un autre. Porteurs de gènes de résistance, les transposons jouent un rôle majeur dans la dissémination de résistances entre bactéries d'espèces éloignées.

Taxon : ensemble des organismes reconnus et définis dans chacune des catégories de la classification biologique hiérarchisée. En d'autres termes : contenu concret d'une catégorie. Exemple : Canis lupus, le Loup, est un taxon de rang spécifique (catégorie : espèce) ; les canidés (Chien, Loup, Renard) constituent un taxon de rang familial (catégorie : famille).

Virion : unité élémentaire d'un virus ayant atteint la maturité.

VOG (Viral Otholog Group) : les données VOG sont des séquences de protéines virales regroupées (*clusters*) de manière prédéfinie en famille selon la fonction protéique à laquelle elles sont associées. C'est une base de données utilisées comme données initiales dans la présente étude. Une description détaillée est donnée à la section 4.1.6.2.

BIBLIOGRAPHIE

- Ahn W-K. et Medin D.L. (1992), "A two-stage model of category construction", Cognitive Science, 16, p. 81-121.
- Ahn W-K., Marsh J.K., Luhmann C.C. et Lee K. (2002), "Effect of theory-based feature correlations on typicality judgments", Memory & Cognition, 30, p. 107-118.
- Alpaydin E. (2004), "Introduction to Machine Learning", Cambridge, Massachusetts: The MIT Press.
- Altschul S.F., Madden T.L., Schaffer A.A., Zhang J., Zhang Z., Miller W. et Lipman D.J. (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25(17), p. 3389-402.
- Anderson A.L., Ross B.H. et Chin-Parker S. (2002), "A further investigation of category learning by inference", Memory & Cognition, 30, p. 119-128.
- Anderson J.R. (1990), "The adaptive character of thought", Hillsdale, N J: Erlbaum.
- Anderson J.R. (1991), "The adaptive nature of human categorization", Psychological Review, 98, p. 409-429.
- Anderson T.W. (1958), "An introduction to Multivariate Statistical Analysis", New York, Jonh Wiley.
- Archie J.W. (1989), "A randomization test for phylogenetic information in systematic data", Systematic Zoology, 38, p. 219-252.
- Ashby F.G. et Alfonso-Reese L. (1995), "Categorization as probability density estimation", Journal of Mathematical Psychology, 39, p. 216-233.
- Ashby F.G. et Maddox, W. T. (1993), "Relations between prototype, exemplar, and decision bound models of categorization", Journal of Mathematical Psychology, 37, p. 372-400.
- Ashby, F.G. et Townsend J.T. (1986), "Varieties of perceptual independence", Psychological Review, 93, p. 154-179.
- Asselin de Beauville J-P. et Kettaf F-Z. (2005), "Bases théoriques pour l'apprentissage et la decision en reconnaissance de formes", Ed. Cépaduès.
- Atteson K. (1999), "The performance of the neighbor-joining methods of phylogenetic reconstruction", Algorithmica 25, p. 251-278.
- Baldi P. et Brunak S. (2001), "Bioinformatics: The Machine Learning Approach", 2nd edition. MIT Press.
- Bandea C.I. (1983), "A new theory on the origin and the nature of viruses", J Theor Biol. 105:591-602.
- Bandelt H-J, Forster P et Rohl A (1999), "Median-joining networks for inferring intraspecific phylogenies", Mol Biol Evol 16, p. 37-48.
- Bandelt H-J, Forster P, Sykes BC et Richards MB (1995), "Mitochondrial portraits of human populations using median networks", Genetics 14, p. 743-753.
- Bandelt H-J, Macaulay V et Richards M (2000), "Median networks:speedy construction and greedy reduction, one simulation, and two case studies from human mtDNA", Mol Phylogenet Evol 16,p. 8-28.
- Bandelt H-J. et Dress A.W.M. (1989), "Weak hierarchies associated with similarity measures – an additive clustering technique", Bull Math Biol 51(1), p. 133-166.

- Bandelt H-J. et Dress A.W.M. (1992), "Split decomposition: a new and useful approach to phylogenetic analysis of distance data", Mol Phylogenet Evol 1, p. 242-252.
- Bao Y., Federhen S., Leipe D., Pham V., Resenchuk S., Rozanov M., Tatusov R. et Tatusova T. (2004), "NCBI Genomes Project", Journal of Virology, 78(14), p. 7291-7298.
- Barthélemy J.-P. et Guénoche A. (1988), "Les arbres et les représentations des proximités", Paris, Masson.
- Baxter, J. (2000), "A model of inductive bias learning", Journal of Artificial Intelligence Research, 12, p. 149-198.
- Benner S.A. (2001), "Natural progression", Nature, p. 409-459.
- Benner S.A. (2002), "The past as the key to the present: Resurrection of ancient proteins from eosinophils", Proceeding of the National Academy of Sciences of the U.S.A., p. 4760-4761.
- Benson S.D., Bamford J.K., Bamford D.H. et Burnett R.M. (1999), "Viral evolution revealed by bacteriophage PRD1 and human adenovirus coat protein structures", Cell. 98:825-33.
- Bergh O, Borsheim KY, Bratbak G et Heldal M (1989), "High abundance of viruses found in aquatic environments", Nature, 6233, p. 467-8.
- Berkhin P. (2002), "Survey of Clustering Data Mining Techniques", Accrue Software, Inc.
- Besemer J. et Borodovsky M. (1999), Heuristic approach to deriving models for gene finding, Nucleic Acids Res. 27, p. 3911-3920.
- Besemer J., Lomsadze A. et Borodovsky M. (2001), "Genemarks: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions", Nucleic Acids Res. 29, p. 2607-2618.
- Birney E. (2001), 'Hidden Markov models in biological sequence analysis", Deep computing for the life sciences. Vol 45, Numbers ¾, Http://www.research.ibm.com/journal/rd/453/birney.html.
- Birney E. (2001), "Hidden Markov models in biological sequence analysis, Deep computing for the life sciences", Vol 45, http://www.research.ibm.com/journal/rd/453/birney.html.
- Blanchette M., Diallo A.B., Green E.D., Miller W. et Haussler D. (2007), "Computational reconstruction of ancestral DNA sequences". Chaptire du livre Phylogenomics and Comparative Genomics, Humana Press. A paraître.
- Blanchette M., Diallo A.B., Green E.D., Miller W., Haussler D., "Computational reconstruction of ancestral DNA sequences". Chaptire du livre Phylogenomics and Comparative Genomics, Humana Press (2007), 36 pages, à paraître.
- Blanchette M., Green E.D., Miller W. et Haussler D. (2004), "Reconstructing large regions of an ancestral mammalian genome in silico", Genome Res, 14(12), p. 2412-2423.
- Blanquart S. et Lartillot N. (2006), "A Bayesian Compound Stochastic Process for Modeling Nonstationary and Nonhomogeneous Sequence Evolution", Molecular Biology and Evolution, 23(11), p. 2058-2071.

- Boc A. et Makarenkov V. (2003), "New Efficient Algorithm for Detection of Horizontal Gene Transfer Events", Algorithms in Bioinformatics, G. Benson and R. Page (Eds.), 3rd Workshop on Algorithms in Bioinformatics, Springer-Verlag, p. 190-201.
- Boc A., Makarenkov V. et Diallo A.B. (2004), " Une nouvelle methode pour la detection de transferts horizontaux de gene : la reconciliation topologique d'arbres de gene et d'especes ", JOBIM 2004, Montreal, Canada.
- Bruner J.S., Goodnow J.J et Austin G.A. (1956), "A Study of Thinking", New York: John Wiley and Sons.
- Bryant D. et Moulton V. (2002), "NeighborNet: an agglomerative method for the construction of planar phylogenetic networks", Algorithms in Bioinformatics: Second International Workshop, WABI 2002, Rome, Italy, September 17-21, p 375-391.
- Büchen-Osmond C. (2003), "Taxonomy and Classification of Viruses", In: Manual of Clinical Microbiology, 8th edition, Volume 2, ASM Press, Washington DC, p. 1217-1226.
- Buneman P. (1971), "The recovery of trees from measures of dissimilarity", Mathematics in Archaeological and Historical Sciences, F.H. Hodson, D.G. Kendall, P. Tautu (Eds.), Edimburg University Press, 387-395.
- Canchaya C., Proux C., Fournous G., Bruttin A. et Brussow H. (2003),"Prophage Genomics", Microbiology and Molecular Biology Reviews, 67, p. 238-276.
- Chang S.W., Ugalde J.A. et Matz M.V. (2005), "Applications of ancestral protein reconstruction in understanding protein function: GFP-like proteins", Methods in Enzymology, 395, p. 652-670.
- Cheng P.W. et Novick L.R. (1990), "A probabilistic contrast model of causal induction. Journal of Personality and Social Psychology", 58, p. 545–567.
- Chindelevitch L., Li Z., Blais E. et Blanchette M. (2006), "On the inference of parsimonious indel evolutionary scenarios", Journal of Bioinformatics and Computational Biology, 0:In press.
- Chin-Parker S. et Ross B.H. (2004), "Diagnosticity and prototypicality in category learning: A comparison of inference learning and classification learning", Journal of Experimental Psychology: Learning, Memory, & Cognition, 30, p. 216-226.
- Clapper J.P. et Bower G.H. (1994), "Category invention in unsupervised learning", Journal of Experimental Psychology: Learning, Memory, & Cognition, 20, p. 443-460.
- Clapper, J. P., & Bower, G. H. (1994). Category invention in unsupervised learning. Journal of Experimental Psychology: Learning, Memory, & Cognition, 20, p. 443-460.
- Cohen J. et Basu K. (1987), "Alternative Models of Categorization: Toward a contingent Processing Framework", Journal of Consumer Research, 13.
- Collins R.F., Gellatly D.L., Sehgal O.P. et Abouhaidar M.G. (1998), "Self-cleaving circular RNA associated with rice yellow mottle virus is the smallest viroid-like RNA", Virology 241, p. 269-75.
- Cordier F. et Dubois D. (1981), "Typicalité et représentation cognitive, Cahiers de Psychologie cognitive", 1, p. 299-333.
- Danks D. (2007), "Theory Unification and Graphical Models in Human Categorization", In A. Gopnik & L. Schulz, eds. Causal Learning: Psychology, Philosophy, and Computation, p. 173-189, Oxford: Oxford University Press.

- Darlu P. et Tassy P. (2004), "La Reconstruction phylogénétique. Concepts et Méthodes", Société Française de Systématique, Dernière modification septembre 2004.
- Dayhoff M.O., Schwartz R.M. et Orcutt B.C. (1978), "A model of evolutionary change in proteins, matrixes for detecting distant relationships", In Atlas of Protein Sequence and Structure, vol 5, p. 345-358, National Biomedical Research Foundation, Silver Spring.
- De Pasquale J-F. et Meunier J-G. (2002), "Categorisation techniques in computer assisted reading and analysis of text (CARAT) in the humanities", Proceeding of the ACH/ACLL Conference, Computer and the Humanities, Kluwer, Volume 37, No. 1, Février 2003.
- Delsuc F. et Douzery E.J.P. (2004), "Les méthodes probabilistes en phylogénie moléculaire, (2) L'approche bayésienne", Biosystema 22, Avenir et pertinence des méthodes d'analyse en phylogénie moléculaire, p. 75-86.
- Delwiche C.F. et Palmer J.D. (1996), "Rampant Horizontal Transfer and Duplication of Rubisco Genes in Eubacteria and Plastids", Mol. Biol. Evol. 13, p. 873-882.
- Dempster A., Laird N. et Rubin D. (1977), "Maximum likelihood from incomplete data via the EM algorithm", Journal of the Royal Statistical Society, Series B, 39, 1, p. 1-38.
- Deschamps J.C. (1977), "Effect of crossing category memberships on quantitative judgement", European Journal of Social Psychology, 7, p. 122-126.
- Deveau H., Labrie M.-C. Chopin S.J. et Moineau M. (2006), "Biodiversity and classification of lactococcal phages", Applied and Environmental Microbiology, p. 4338-4346.
- Diallo A.B., Makarenkov V. et Blanchette M. (2006), "Finding Maximum Likelihood Indel Scenarios", Comparative Genomics, p 171-185.
- Diday E et Bertrand P (1984), "An extension of hierarchical clustering: the pyramidal representation", In: ES Gelsema and LN Kanal eds., Pattern Recognition in Practice, Amsterdam, North-Holland, p. 411-424.
- Doolittle W.F. (1999), "Phylogenetic classification and the universal tree, Science 284, p. 2124-2129.
- Dopkins S. et Gleason T. (1997), "Comparing exemplar and prototype models of categorization", Canadian Journal of Experimental Psychology 51(3), p. 212-230.
- Douzery E.J.P., Snell E.A., Bapteste E., Delsuc F. et Philippe H. (2004), "The timing of eukaryotic evolution: does a relaxed molecular clock reconcile proteins and fossils?", Proceedings of the National Academy of Sciences of the USA 101, p. 15386-15391.
- Dubois D. (1983), "Analyse de 22 catégories sémantiques du français : Organisation catégorielle, lexique et représentation", L'Année Psychologique, 83, p. 465-489.
- Dubois D. (1991), "Catégorisation et cognition : "10 ans après", une évaluation des concepts de Rosch", dans : Sémantique et cognition. Catégories, prototypes, typicalité. Dubois, Danièle (ed.). Paris : CNRS, p. 31-54.
- Duda R., Hart P. et Stock D.G. (2001), "Pattern Classification", Wiley InterScience, Second Edition.
- Durbin R., Eddy S., Krogh A. et Mitchison G. (2006), "Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids", Cambridge University Press, 1998. Corrected 10th printing 2006.
- Dutilh B.E., Huynen M.A., Bruno W.J. et Snel B. (2004), "The consistent phylogenetic signal in genome trees revealed by reducing the impact of noise", Journal of Molecular Evolution, 58, p.527-539.

- Efron B (1982), "The Jackknife, the Bootstrap and Other Resampling Plans", CBMS-NSF Regional Conference Series in Applied Mathematics, Monograph 38, SIAM, Philadelphia.
- Eisen J.A. (2000), "Horizontal gene transfer among microbial genomes: New insights from complete genome analysis", Current Opinion in Genetics & Development 10, p. 606-611.
- Ewens W.J. et Grant G.R. (2005), "Statistical Methods in Bioinformatics: An Introduction (Statistics for Biology and Health)", Springer.
- Excoffier L et Smouse PE (1994), "Using allele frequencies and geographic subdivision to reconstruct gene trees within a species: molecular variance parsimony", Genetics 136, p. 343-359.
- Faith D.P. (1900), "Chance marsupial relationships", Nature, 345, p. 393-394.
- Farris J.S. (1970), "Methods for computing Wagner trees", Systematic Zoology, 19, p. 83-92.
- Feller W. (1971), "An Introduction to Probability Theory and Its Applications", Vol 2, John Wiley and Sons.
- Felsenstein J. (1981), "Evolutionary trees from DNA sequences: a maximum likelihood approach", Journal of Molecular Evolution, 17, 6, p. 368-76.
- Felsenstein J. (1985), "Confidence limits on phylogenies: an approach using the bootstrap", Evolution 39, p. 783-791.
- Felsenstein J. (1997), "An alternating least-squares approach to inferring phylogenies from pairwise distances", Syst Zool 46, p. 101-111.
- Felsenstein J. (2004), PHYLIP (http://evolution.genetics.washington.edu/phylip.html - software download page and software manual) - PHYLogeny Inference Package.
- Felsenstein J. (2006), "Accuracy of Coalescent Likelihood Estimates: Do we need More Sites, More Sequences, or More Loci?", Molecular Biology and Evolution, 13, 3, p. 691-700.
- Felsenstein J. et Churchill G. (1996), "A hidden markov model approach to variation among sites in rate of evolution", Molecular Biology and Evolution, 13, p. 93-104.
- Fitch W.M. (1971), "Toward defining the course of evolution: Minimum change for a specific tree topology", Systematic Zoology, 20, p. 406-416.
- Fitch W.M. (2000), "Homology a personal view on some of the problems", Trends Genet. 16, p. 227-231.
- Forterre P., Brochier C. et Philippe H. (2002), "Evolution of the Archaea", Theor Popul Biol.61:409-22.
- Foulds LR, Hendy MD et Penny D (1979), "A graph theoretic approach to the development of minimal phylogenetic trees". J Mol Evol 13, p. 127-149.
- Fredslund J., Hein J. et Scharling T. (2004), "A large version of the small parsimony problem", In Proceedings of the 4th Workshop on Algorithms in Bioinformatics (WABI).
- Fried L.S. et Holyoak K.J. (1984), "Induction of category distributions: A framework for classification learning", Journal of Experimental Psychology: Learning, Memory and Cognition 10, p. 234-257.
- Galtier N. (2001), "Maximum-likelihood phylogenetic analysis under a covarion-like model", Molecular Biology and Evolution, 18(5), p. 866-873.

- Galus C. (2006), "La capacité d'adaptation des bactéries étudiée par l'Europe", article paru dans LeMonde, Déc 2006.
- Gascuel O. (1997), "BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data", Molecular Biology and Evolution, 14, 7, p. 685-95.
- Geman S. et Geman G.D. (1984), "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, p. 721-741.
- Ghahramani Z. (1999), "Probabilistic Models for Unsupervised Learning", University College London, UK, NIPS Tutorial.
- Ghahramani Z. (2004), "Unsupervised Learning", University College London, UK.
- Glazko G., Gordon A. et Mushegian A. (2005), "The choice of optimal distance measure in genome-wide datasets", Bioinformatics, p. iii3-iii11.
- Gluck M. et Bower G. H. (1988), "From conditioning to category learning: An adaptive network model", Journal of Experimental Psychology: General, 117, p. 227-247.
- Glymour C. (1998), "Learning causes: Psychological explanations of causal explanation", Minds and Machines, 8, p. 39-60.
- Gorbalenya A.E., Koonin E.V. et Wolf Y.I. (1990), "A new superfamily of putative NTP-binding domains encoded by genomes of small DNA and RNA viruses", FEBS Lett. 262:145-8.
- Gray R.D. & Atkinson Q.D. (2003), "Language-tree divergence times support the Anatolian theory of Indo-European origin", Nature, vol. 426, p. 435-439.
- Griffiths T.L., Sanborn A.N., Canini K.R. et Navarro D.J. (2008), "Categorization as nonparametric Bayesian density estimation", in M. Oaksford and N. Chater (Eds.). The probabilistic mind: Prospects for rational models of cognition. Oxford: Oxford University Press.
- Guénoche A., Leclerc B. et Makarenkov V. (2004), "On the extension of a partial metric to a tree metric", Discrete Mathematics 276(1-3), p. 229-248.
- Hadjichristidis C., Sloman S., Stevenson R. et Over D. (2004), "Feature centrality and property induction", Cognitive Science, 28, p. 45-74.
- Hallett M.T. et Lagergren J. (2001), Efficient algorithms for lateral gene transfer problems, In: Proceedings of the 5th Ann Int Conf Compt Mol Biol (RECOMB 01), New York, ASM Press, p. 149-156.
- Harnad S. (1987), "Introduction: Psychological and cognitive aspects of categorical perception: A critical overview", In S. Harnad (Ed.), Categorical perception, New York: Cambridge University Press, p.1-28.
- Hartigan J. (1975), "Clustering Algorithms", John Wiley & Sons, New York, NY.
- Hastie T., Tibshirani R. et Friedman J. (2001), "The Elements of Statistical Learning: Data Mining, Inference and Prediction", Springer-Verlag, New York.
- Hastings W. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications", Biometrika, 57(1) :97-109.
- Hendrix R.W. (2002), "Bacteriophages: evolution of the majority", Theoretical Population Biology, 61, p. 471-480.
- Hendrix R.W., Smith M.C., Burns R.N., Ford M.E., Hatfull G.F. (1999), "Evolutionary relationships among diverse bacteriophages and prophages: all the world's a phage", Proceedings of the National Academy of Sciences of the USA 96, 2192-2197.

- Henikoff S. et Henikoff J.G. (1992),"Amino acid substitution matrices from protein blocks", Proceedings of the National Academy of Sciences USA, 89, p. 10915-10919.
- Holyoak K.J. et Thagard P. (1995), "Mental leaps", Cambridge, MA: MIT Press.
- Homa D., Sterling S. et Trepel L. (1981), "Limitations of exemplar-based generalization and the abstraction of categorical information", Journal of Experimental Psychology: Human Learning and Memory, 7, p. 418-439.
- Huelsenbeck J.P. & Ronquist F. (2001), "Mrbayes : Bayesian inference of phylogenetic trees", Bioinformatics, 17, p. 754-755.
- Huelsenbeck J.P., Larget B. et AlfaroM.E. (2004), "Bayesian phylogenetic model selection using reversible jump Markov chain Monte Carlo", Molecular Biology and Evolution, 21, p. 1123-1133.
- Huelsenbeck J.P., Ronquist F., Nielsen R. et Bollback J.P. (2001), "Bayesian inference of phylogeny and its impact on evolutionary biology", Science, 294, p. 2310-2314.
- Huson D.H. (1998), "SplitsTree: a program for analyzing and visualizing evolutionary data", Bioinf 141, p. 68-73.
- Huson D.H. (2005), "ISMB-Tutorial:Introduction to Phylogenetic Networks", Center for Bioinformatics, Tubingen University, Sand 14, 72075 Tubingen, Germany, June 25, 2005.
- Iyer L.M., Aravind L. et Koonin E.V. (2001), "Common origin of four diverse families of large eukaryotic DNA viruses", Journal of Virology, 75:11720-34.
- Jain A. et Dubes R. (1988), "Agorithms for Clustering Datal", Prentice-Hall, Englewood Cliffs, NJ.
- Jarvis A.W., Fitzgerald G. F., Mata M., Mercenier A., Neve H., Powell I. B., Ronda C., Saxelin M. et Teuber M. (1991), "Species and type phages of lactococcal bacteriophages", Intervirology, 32, p. 2-9.
- Jaynes E.T. (1994), "Probability theory: The logic of science", documentation en ligne, Web : http://omega.albany.edu:8008/JaynesBook.html.
- Jermann T.M., Optiz J.G., Stackhouse J. et Benner S.A. (1995), "Reconstructing the evolutionary history of the artiodactyl ribonuclease superfamily", Nature, 374, p. 57-59.
- Jukes T.H. et Cantor C. (1969), "Mammalian Protein Metabolism", p. 21-132 dans H. N. Munro, editor, Evolution of protein molecules, Academic Press, New York, 1969.
- Kasabov N. et Pang S. (2004), "Transductive support vector machines and applications in bioinformatics for promoter recognition", Neural Information Processing, Letters & Review, v3 i2, p. 31-38.
- Kimura M. (1981), "Estimation of evolutionary distances between homologous nucleotide sequences", PNAS USA 78, p. 454-458.
- Koonin E.V. et Ilyina TV (1992) Geminivirus replication proteins are related to prokaryotic plasmid rolling circle DNA replication initiator proteins. J Gen Virol. 73:2763-6.
- Korbel J.O., Snel B., Huynene M.A. et Bork P. (2002), "SHOT: a web server for the construction of genome phylogenies", Trends Genet. 18, p. 158-162.
- Krishnan N.M., Seligmann H., Stewart C-B., Jason de Koning A.P. et Pollock D.D. (2004), "Ancestral sequence reconstruction in primate mitochondrial DNA: compositional bias and effect on functional inference", Molecular Biology and Evolution, 21, p. 1871-1883.

- Kruschke J.K. (1992), "ALCOVE: An exemplar-based connectionist model of category learning", Psychological Review, 95, p. 471-484.
- Kuhner M. K., Yamato J. et Felsenstein J. (1995), "Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling", Genetics, 140(4), p 1421-1430.
- Kuhner M.K. et Felsenstein J. (1994), "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates", Molecular Biology and Evolution, 11, 3, p. 459-68.
- La Scola B., Audic S., Robert C., Jungang L., de Lamballerie X., Drancourt M., Birtles R., Claverie J.M. et Raoult D. (2003), "A giant virus in amoebae", Science, 299:2033.
- Lanave C., Preparata G., Saccone C. et Serio G. (1984), "A new method for calculating evolutionary substitution rates", Journal of Molecular Evolution, 20(1), p. 86-93.
- Larget B. et Simon D.L. (1999), "Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees", Molecular Biology and Evolution, 16, p. 750-759.
- Larget B., Simon D.L. et Kadane J.B. (2002), "Bayesian phylogenetic inference from animal mitochondrial genome arrangements (with discussion)", Journal of the Royal Statistical Society:Series B 64, p. 681-693.
- Larget B., Simon D.L., Kadane J.B. et Sweet D. (2005), "A bayesian analysis of metazoan mitochondrial genome arrangements", Molecular Biology and Evolution, 22(3), p. 486-95.
- Lartillot N. et Philippe H. (2004), "A Bayesian mixture model for across-site rate heterogeneities in the amino-acid replacement process", Molecular Biology and Evolution, 21, p. 1095-1109.
- Law A.M. and Kelton W.D. (1991), "Simulation Modelling and Analysis", McGraw-Hill.
- Lawrence JG, Hatfull GF et Hendrix RW (2002), "Imbroglios of viral taxonomy: genetic exchange and failings of phenetic approaches", Journal of Bacteriology,184, p. 4891-905.
- Leclerc B. (1996), "Minimum Spanning Trees and Types of Dissimilarities", Eur. J. Comb. 17(2-3), p. 255-264.
- Leclerc B. et Makarenkov V. (1998), "On some relations between 2-trees and tree metrics", Discrete Mathematics 192(1-3), p. 223-249.
- Legendre P. et Makarenkov V. (2002), "Reconstruction of biogeographic and evolutionary networks using reticulograms", Systematic Biology, 51, p. 199-216.
- Lepage T., Lawi S., Tupper P. et Bryant D. (2006), "Continuous and Tractable models for the Variation of Evolutionary Rates", Mathematical Biosciences, Vol 199, Issue 2, p. 216-233.
- Lerat E, Daubin V et Moran N.A. (2003), "From gene trees to organismal phylogeny in prokaryotes: the case of the γ –proteobacteria", PLoS Biology 1, p. 101-9.
- Letunic I. et Bork P. (2007), "Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation", Bioinformatics 23(1), p. 127-8.
- Lewis P.O. (2001), "Phylogenetic systematics turns over a new leaf", Trends in Ecology & Evolution, 16, p. 30-37.
- Li S. (1996), "Phylogenetic tree construction using Markov chain Monte Carlo", Ph. D. Dissertation, Ohio State University.

- Liu J., Glazko G., Mushegian A. (2006), "Protein repertoire of double-stranded DNA bacteriophages", Virus Research, 117, p. 68-80.
- Love B.C. (2002), "Comparing supervised and unsupervised category learning", Psychonomic Bulletin & Review, 9, p. 829-835.
- Love, B.C., Medin, D.L, Gureckis, T.M (2004), SUSTAIN: A Network Model of Category Learning, Psychological Review, 11, p. 309-332.
- Luce R.D. (1963). Detection and recognition. In R. D. Luce, R. R. Bush & E. Galanter (Eds.), Handbook of mathematical psychology. New York: Wiley.
- MacKay D.J.C. (1992), "Bayesian interpolation ", Neural Computation 4, p. 415-447.
- Maddison W.P., (1997), "Gene trees in species trees", Systematic Biology, Vol 46, 3, p. 523-536.
- Makarenkov V et Leclerc B (1999), "An algorithm for the fitting of a tree metric according to a weighted least-squares criterion", Journal of Classification, 16, p. 3-26.
- Makarenkov V., Boc A. et Diallo B. (2004), "Representing lateral gene transfer in species classification, Unique scenario", P. 439-446 dans D. Banks, L. House, F. R. McMorris, P. Arabie et W. Gaul, eds. Classification, Clustering and Data Mining Applications, Springer Verlag, proceeding of IFCS 2004, Chicago.
- Makarenkov V., Boc A., Diallo Alpha B. et Diallo Abdoulaye B. (2008), "Algorithms for detecting complete and partial horizontal gene transfers: Theory and practice", in *Data Mining and Mathematical Programming*, P.M. Pardalos and P. Hansen eds., CRM Proceedings and AMS Lecture Notes, 45, p. 159-179.
- Maniloff J. et Ackermann H-W. (1998), "Taxonomy of bacterial viruses: establishment of tailed virus genera and the other Caudovirales", Archives of Virology, 143, p. 2051-2063.
- Markman A.B. et Ross B.H. (2003), "Category use and category learning", Psychological Bulletin, 129, p. 592-613.
- Matte-Tailliez O., Brochier C., Forterre P. et Philippe H. (2002), "Archaeal phylogeny based on ribosomal proteins", Mol. Biol. Evol. 19, p. 631-639.
- Mau B. (1996), "Bayesian phylogenetic inference via Markov chain Monte Carlo methods", Ph.D. Dissertation, University of Wisconsin.
- Mau B. et Newton M.A. (1997), "Phylogenetic inference for binary data on dendrograms using Markov chain Monte Carlo", Journal of Computational and Graphical Statistics, 6, p. 122-131.
- Mayr E. (1997), "This is Biology", Cambridge MA., Harvard University Press.
- McClelland J.L. et Rogers T.T. (2003), "The parallel distributed processing approach to semantic cognition", Nature Reviews Neuroscience, 4, p. 1-13.
- McClure MA (1991), "Evolution of retroposons by acquisition or deletion of retrovirus-like genes", Molecular Biology and Evolution, 8:835-56.
- McDade L. (1995), "Hybridization and phylogenetics", In PC Hoch and AG Stephenson, eds., Experimental and Molecular Approaches to Plant Biosystematics, Monographs in Systematic Botany from the Missouri Botanical Garden. P. 305-331.
- Medin D.L. (1983), "Structural principles in categorization", In T. J. Tighe & B. E. Shepp (Eds.), Perception, cognition, and development: Interactional analyses, Hillsdale, NJ: Erlbaum, p. 203-230.

- Medin D.L. et Schaffer M.M. (1978),"Context theory of classification learning", Psychological Review, 85, 207-238.
- Medin D.L., Aitom M.W., Edelson S.M. et Freko D. (1982), "Correlated symptoms and simulated medical classification. Journal of Experimental Psychology", Learning, Memory, and Cognition, 8, p. 37-50.
- Mervis C.B. et Rosch E. (1981), "Categorization of natural objects", Annual Review of Psychology, 32, p. 89-115.
- Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H. et Teller E. (1953), "Equations of State Calculations by Fast Computing Machines", Journal of Chemical Physics, 21(6), p. 1087-1092.
- Minda J. P. et Ross B.H. (2004), "Learning categories by making predictions: An investigation of indirect category learning", Memory & Cognition, 32 (8), p. 1355-1368.
- Minda J. P. et Smith J. D. (2002), "Comparing prototypebased and exemplar-based accounts of category learning and attentional allocation", Journal of Experimental Psychology: Learning, Memory, & Cognition, 28, p. 275-292.
- Minda J.P. et Smith J. D. (2001), "Prototypes in category learning: The effects of category size, category structure, and stimulus complexity", Journal of Experimental Psychology: Learning, Memory, & Cognition, 27, p. 775-799.
- Mirkin B. et Koonin E. (2003), "A top-down method for building genome classification trees with linear binary hierarchies", DIMACS series in Discrete Mathematics and Theoretical Computer Science.
- Mirkin B.G., Muchnik I. et Smith T.F. (1995), "A Biologically Consistent Model for Comparing Molecular Phylogenies", Journal of Comp. Biol., 2, p. 493-507.
- Mitchell T.M. (1997), "Machine Learning", Carnegie Mellon University, McGraw Hill.
- Mitchell T.M. (2006), "The Discipline of Machine Learning", CMS-ML-06-108.
- Murphy G.L. et Medin D.L. (1985), "The role of theories in conceptual coherence", Psychological Review, 92, p. 289-316.
- Nei M. (1991), "Relative efficiencies of different tree-making methods for molecular data", In Phylogenetic analysis of DNA sequences, M.M. Miyamoto and J. Cracraft, Editors, Oxford Univ. Press.
- Nguyen D., Boc A. et Makarenkov V. (2005), "HGT-Simulator: logiciel pour simuler des transferts horizontaux de genes", proceedings of the SFC2005, Montreal, p. 215-219.
- Nilsson A.S. et Haggard-Ljungquist E. (2001), "Detection of homologous recombination among bacteriophage P2 relatives", Molecular Biology and Evolution, 21, p. 259-69.
- Nosofsky R.M. (1986), "Attention, similarity and the identification-categorization relationship", Journal of Experimental Psychology: General, 115, p. 39–57.
- Nosofsky R.M. (1987), "Attention and learning processes in the identification and categorization of integral stimuli", Journal of Experimental Psychology: Learning, Memory, & Cognition, 13, p. 87-108.
- Nosofsky R.M. et Zaki S.R. (1998), "Dissociations between categorization and recognition memory in amnesic and normal individuals: An exemplar-based interpretation", Psychological Science, 9, p. 247-255.
- Nosofsky R.M., Palmeri, T.J. et McKinley S.K. (1994), "Rule-plus-exception model of classification learning", Psychological Review, 101, p. 53-79.

- Ochman H., Lawrence J. G. et Groisman E.A. (2000), "Lateral gene transfer and the nature of bacterial innovation", Nature 405, p. 299-304.
- Ota S. et Li W.H. (2000), "NJML: a hybrid algorithm for the neighbor-joining and maximum-likelihood methods", Molecular Biology and Evolution, 17, 9, p. 1401-9.
- Pace N.R. (1997), "A molecular view of microbial diversity and the biosphere", Science 276:734-740.
- Pacherie E. (2004), "Catégories et concepts", École Normale Supérieure, France, http://pacherie.free.fr/COURS/Centrale/concepts.html.
- Page R.D.M. et Charleston M.A. (1998), "Trees within trees: phylogeny and historical associations", Trends in Ecology and Evolution, 13, p. 356-359.
- Palmeri T. J. et Nosofsky R.M. (1995), "Recognition memory for exceptions to the category rule", Journal of Experimental Psychology: Learning, Memory, & Cognition, 21, p. 548-568.
- Pauling L. et Zuckerkandl E. (1963), "Chemical paleogenetics, molecular restoration studies of extinct forms of life", Acta chemica Scandinavica, 17, p. 9-16.
- Pearl J. (2000), "Causality: models, reasoning, and inference", Cambridge, UK: Cambridge University Press.
- Posner M.I. et Keele S.W. (1968), "On the genesis of abstract ideas", Journal of Experimental Psychology, 77(3), p. 353-63.
- Rabiner L.R. et Juang B.H. (1986), "An Introduction to Hidden Markov Models, IEEE ASSP magazine, 3, p. 4-16.
- Rannala B. & Yang Z. (1996), "Probability distribution of molecular evolutionary trees : a new method of phylogenetic inference", Journal of Molecular Evolution, 43, p. 304-311.
- Ravin V., Ravin N., Casjens S., Ford M.E., Hatfull G.F. et Hendrix R.W. (2000), "Genomic sequence and analysis of the atypical temperate bacteriophage N15", Journal of Molecular Evolution, 299, p. 53-73.
- Receveur-Bréchot V. et Grob M. (2006), "Les protéines humaines", Magazine La Recherche – Hors Série, Spéciale Biologie, p. 32-37, No. 2.
- Rehder B. (2003), "A causal-model theory of conceptual representation and categorization", Journal ofExperimental Psychology: Learning, Memory, and Cognition, 29, p. 1141-1159.
- Rehder B. et Burnett R. (2005), "Feature inference and the causal structure of categories", Cognitive Psychology, 50, p. 264-314.
- Rehder B. et Hastie R. (2001), "Causal knowledge and categories: The effects of causal beliefs on categorization, induction, and similarity", Journal of Experimental Psychology: General, 130, p. 323-360.
- Rehder B. et Hastie R. (2004), "Category coherence and category-based property induction", Cognition, 91, p. 113-153.
- Ricagno S., Campanacci V., Blangy S., Spinelli S., Tremblay D., Moineau S., Tegoni M. et Cambillau C. (2006), "Crystal structure of the receptor-binding protein head domain from L.lactis phage bIL170", J. of Virology, p. 9331-9335.
- Ripley B.D. (1996), "Pattern Recognition and Neural Networks", Cambridge University Press.

- Rips L.J. (1975), "Inductive judgments about natural categories", Journal of Verbal Learning and Verbal Behavior, 14, p. 665-681.
- Robinson D.F. et Foulds L.R. (1981), "Comparison of phylogenetic trees", Mathematical Biosciences, 53, p. 131-147.
- Robinson D.F. et Foulds L.R. (1981), "Comparison of phylogenetic trees", Mathematical Biosciences, 53, p. 131-147.
- Rodriguez F., Oliver J. L., Marin A. et Medina J.R. (1990), "The general stochastic model of nucleotide substitution", Journal of Theoretical Biology, 142(4), p. 485-501.
- Rohwer F. et Edwards R. (2002), "The phage proteomic tree: a genome-based taxonomy for phage", Journal of Bacteriology, 184, p. 4529-4535.
- Ronquist F. & Huelsenbeck J.P. (2003), "Mrbayes 3 : Bayesian phylogenetic inference under mixed models", Bioinformatics, 19, p. 1572-1574.
- Rosch E. (1973), "Cognitive Reference Points", Cognitive Psychology, 7, p. 532-547.
- Rosch E. (1973), "On the internal structure of perceptual and semantic categories", In T.E. Moore (Ed.), Cognitive Development and the Acquisition of Language, New York : Academic Press.
- Rosch E. (1975), "Universals and Cultural Specifics in Human Categorization", Cross-Cultural Perspectives on Learning, In R. Brislin, S. Bochner, W. Lonner (Eds); New York : Sage-Wiley, p. 177-206.
- Rosch E. et Mervis C. (1975), "Family Ressemblances : Studies in the Internal Structure of Categories", Cognitive Psychology, 7, p. 573-605.
- Rosch E., Mervis C.B., Gray W., Johnson D. et Boyes-Braem P. (1976), "Basic objects in natural categories", Cognitive Psychology, 8, 3, p. 82-439.
- Rosseel Y. (2002), "Mixture models of categorization", Journal of Mathematical of Psychology, 46, p. 178-210.
- Saitou N. et Nei M. (1987), "The Neighbor-Joining method: a new method for reconstructing phylogenetic trees", Molecular Biology and Evolution, 4, p. 406-425.
- Sattah S. et Tversky A. (1977), "Additive similarity trees", Psychometrica, 42, p. 319-45.
- Schaffer A.A., Aravind L., Madden T.L., Shavirin S., Spouge J.L., Wolf Y.I., Koonin E.V. et Altschul S.F. (2001), "Improving the accuracy of PSIBLAST protein database searches with composition-based statistics and other refinements", Nucleic Acids Res. 29, p. 2994-3005.
- Shepard R.N. (1957), "Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space", Psychometrika, 22, p. 325-345.
- Smith E.E. (1994), "Concepts and categorization", In E.E. Smith & D.N. Osherson (Eds.), An invitation to cognitive science, Cambridge, MA: MIT Press, p. 3-33.
- Smith E.E. et Medin D.L. (1981), "Categories and concepts", Cambridge, MA: Harvard University Press.
- Snel B., Huynen M.A. et Dutilh B.E. (2005), "Genome trees and the nature of genome evolution", Annu. Rev. Microbiol., May 9; [Epub ahead of print].
- Soding J. (2004), "Protein homology detection by HMM-HMM comparison", Bioinformatics. 21, 951–960 (Epub 2004 Nov 5).
- Sokal R.R. et Michener C.D. (1958), "A statistical method for evaluating systematic relationships", University of Kansas Science Bulletin, 38, p. 1409-1438.

- Soltis P.S., Soltis D. E. (2003), "Applying the Bootstrap inPhylogeny Reconstruction", Statistical Science, Vol 18, 2, Institute of Mathematical Statistics, p. 256-267.
- Spinelli S., Campanacci V., Blangny S., Moineau S., Tegoni M. et Cambillau C. (2006), "Modular structure of the receptor binding proteins of lactococcus lactis phages: the RBP structure of the temperature phages TP901-1", Journal of Biological Chemistry, 20, p. 4256-14262.
- Spinelli S., Desmyter A., Verrips C.T., de Haard H.J., Moineau S. et Cambillau C. (2005), "Lactococcal bacteriophage p2 receptor-binding protein structure suggests a common ancestor gene with bacterial and mammalian viruses", Nature Structural & Molecular Biology, 13, p. 85-89.
- Studier J.A. et Keppler K.J. (1988), "A note on the neighbor-joining algorithm of Saitou and Nei The neighbor-joining method: a new method for reconstructing phylogenetic trees", Molecular Biology and Evolution 5(6), p. 729-31.
- Swofford D.L., Olsen G.J., Waddell P.J. et Hillis D.M. (1996), "Phylogenetic inference", In Molecular Systematic, In: Hillis, D.M., Moritz, C. and Mable B.K., Editors, Sinauer Associates: Massachusetts.
- Tatusov R.L., Koonin E.V. et Lipman D.J. (1997), "A genomic perspective on protein families". Science 278, p. 631-637
- Thompson J.D., Higgins D.G. et Gibson T.J. (1994), "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice", Nucleic Acids Research 22, p. 4673-4680.
- Thorne J., Kishino H. et Felsenstein J. (1992), "Inching toward reality: an improved likelihood model of sequence evolution", Journal of Molecular Evolution, 34, p. 3-16.
- Torvisk T et Dundas I.D. (1978), "Halophilic phage specific for Halobacterium salinarium", In Energetics and structure of halophilic microorganism, Elsevier, Amsterdam, p. 609-615.
- van Regenmortel, M.H.V., Fauquet C.M., Bishop D.H.L., Carstens E., Estes M.K., Lemon S., Maniloff J., Mayo M.A., McGeoch D.J., Pringle C. R. and Wickner R. (ed.). (2000), "Virus Taxonomy. Classification and Nomenclature of Viruses. Seventh Report of the International Committee on Taxonomy of Viruses", Academic Press, Inc., p. 1162.
- Vapnik V. (1995), "The nature of statistical learning theory", Springer-Verlag, New York.
- Wagner P.L., Waldor M.K. (2002), "Bacteriophage control of bacterial virulence", Infection and Immunity, 70, p. 3985-3993.
- Waldmann M.R., Holyoak, K.J. et Fratianne, A. (1995). Causal models and the acquisition of category structure. Journal of Experimental Psychology: General, 124, p. 181-206.
- Ward J.H. (1963), "Hierarchical grouping to optimize an objective function", Journal of American Statistical Association, 77, p. 841-847.
- Wittgenstein L. (1953), "Philosophical investigations", New York: Macmillan.
- Wolf Y.I., Rogozin I.B., Grishin N.V. et Koonin E.V. (2002), "Genome trees and the tree of life", Trends Genet. 18, 472–479. 2002.

- Wolf Y.I., Rogozin I.B., Grishin N.V., Tatusov R.L. et Koonin E.V. (2001), "Genome trees constructed using five different approaches suggest new major bacterial clades", BMC Evolutionary Biology. 1, 8.
- Wommack K.E. et Colwell R.R. (2000), "Virioplankton: viruses in aquatic ecosystems", Microbiol. Mol. Biol. Rev. 64, p. 69-114.
- Wu C.F.J. (1986), "Jackknife, bootstrap and other resampling plans in regression analysis", Annals of Statistics 14, p.1261-1295.
- Xiong Y. et Eickbush T.H. (1990), "Origin and evolution of retroelements based upon their reverse transcriptase sequences', EMBO J. 10:3353-62.
- Yamauchi T. et Markman A.B. (1998), "Category learning by inference and classification", Journal of Memory & Language. 39, p. 124-148.
- Yang Z. (1996), "Among-site rate variation and its impact on phylogenetic analysis", Trends in Ecology & Evolution, 11, p. 367-372.
- Yang Z. et Rannala B. (1997), "Bayesian phylogenetic inference using DNA sequences : a Markov chain Monte Carlo method", Molecular Biology and Evolution, 14, p. 717-724.
- Zadeh L.A. (1965), "Fuzzy sets", Information and control, 8, p. 338-358.
- Zaretskii K. (1965), "Construction d'un arbre sur la base d'un ensemble de distances entre ses feuilles", Uspekhi Mat. Nauk., 20, p. 90-92.
- Zhu X. (2006), "Semi-supervised learning literature survey", University of Wisconsin – Madison, Last modified on December 9.

ANNEXES

Annexe A – Processus stochastiques

A.1 Chaînes de Markov

En mathématiques, une chaîne de Markov est un processus stochastique possédant la propriété markovienne. La chaîne de Markov est souvent décrite par un graphe orienté où chaque état est représenté par un sommet et chaque transition par un arc. Les arcs sont étiquetés par des probabilités de passage d'un état k à un autre état l. Ces probabilités portent plusieurs noms : probabilités de transitions, matrice de transition ou encore noyau de transition. Elles s'écrivent a_{st} tel que :

$$a_{kl} = P(x_n = l | x_{n-1} = k) \tag{A.1}$$

Une chaîne de Markov en temps discret est une séquence x_1, x_2, x_3,..., x_L de variables aléatoires. L'ensemble de leurs valeurs possibles est appelé l'*espace d'états*, la valeur x_n étant l'état du processus au moment n. Soit un modèle probabiliste de séquence de variables aléatoires x_1, x_2, x_3,..., x_L, on peut écrire la probabilité conditionnelle de la séquence comme :

$$\begin{aligned} P(x) &= P(x_L, x_{L-1}, ..., x_1) \\ &= P(x_L | x_{L-1}, ..., x_1)\, P(x_{L-1} | x_{L-2}, ..., x_1) \ldots P(x_1) \end{aligned} \tag{A.2}$$

La principale propriété d'une chaîne de Markov spécifie que la probabilité de chaque état x_n dépend seulement de la valeur du précédent état x_{n-1}, et non de l'ensemble de la séquence d'états qui précède, c'est-à-dire $P(x_n | x_{n-1}, ..., x_1) = P(x_n | x_{n-1}) = a_{x_{n-1}x_n}$. On parle de chaînes de Markov d'ordre 1. L'équation (A.2) devient alors :

$$\begin{aligned} P(x) &= P(x_L | x_{L-1})\, P(x_{L-1} | x_{L-2}) \ldots P(x_2 | x_1)\, P(x_1) \\ &= P(x_1) \prod_{n=2}^{L} a_{x_{n-1}x_n} \end{aligned} \tag{A.3}$$

A.2 Modèles de Markov Caché

Une des nombreuses extensions aux chaînes de Markov classiques (e.g., chaînes de Markov d'ordre supérieur) est le modèle de Markov caché, ou en anglais *Hidden Markov Models* (HMM). Un modèle HMM a deux types de processus stochastiques, à savoir, le premier sous-jacent qui n'est pas observable (il est caché), si ce n'est seulement à travers un autre ensemble de processus stochastiques qui lui produit de symboles observables [Rabiner & Juang 1986]. Un HMM est donc un graphe d'états connectés où chaque état est en mesure potentiellement d'émettre une série d'observations. Le processus évolue en fonction d'une certaine dimension, souvent mais pas nécessairement, le temps. Pour les séquences biologiques, la dimension temps est remplacée par la position dans la séquence. Le modèle est paramétré avec des probabilités qui dirigent les états aux temps t+1, compte tenu de ce qu'on connaît aux états précédents. Comme les processus évoluent en fonction du temps à travers les états, chaque état peut potentiellement émettre des observations, lesquelles sont vues comme un flux d'observations à travers le temps [Birney 2001].

Formellement, un HMM est défini par la probabilité conjointe d'une séquence observée x et une séquence d'états π :

$$P(x,\pi) = a_{0\pi_1}\prod_{n=1}^{L} e_{\pi_n}(x_n)a_{\pi_n\pi_{n+1}} \tag{A.4}$$

où $a_{0\pi_1}$ est la probabilité d'états initiaux, $a_{\pi_n\pi_{n+1}}$ est la probabilité de *transition* d'états via le chemin (séquence d'états) π à travers le modèle, et $e_{\pi_n}(x_n)$ est la probabilité d'*émission* du caractère x.

Pour un HMM donné, trois types de problèmes peuvent être résolus [Duda et al. 2001] : l'évaluation, le décodage et l'apprentissage. Ils s'expriment formellement par les formules suivantes :

Forward : $$f_l(n+1) = e_l(x_{n+1})\sum_k f_k(n)\, a_{kl} \tag{A.5}$$

Backward : $$b_k(n) = \sum_l a_{kl}e_l(x_{n+1})\, b_l(n+1) \tag{A.6}$$

Viterbi : $$v_l(n+1) = e_l(x_{n+1}) \max_k (v_k(n)\, a_{kl})) \quad \text{(A.7)}$$

Forward-Backward : $$p(\pi_n = k, \pi_{n+1} = l | x, \theta) = \frac{f_k(n)\, a_{kl}\, e_l(x_{n+1})\, b_l(n+1)}{P(x)} \quad \text{(A.8)}$$

L'évaluation s'effectue par la détermination de la probabilité d'une séquence observée particulière sachant un modèle HMM. En observant x_{n-1}, la probabilité associée peut être calculée par l'algorithme *Forward* (A.5). De manière analogue, mais en inversant l'analyse en commençant par la fin de la séquence. La probabilité associée peut être calculée par l'algorithme *Backward* (A.6). Le décodage s'effectue par la détermination de la séquence d'états cachés (ou le chemin) qui aurait le plus probablement généré la séquence observée. La probabilité associée peut être calculée par l'algorithme *Viterbi* (A.7). Enfin, l'apprentissage s'effectue par la génération d'un modèle HMM à partir d'un ensemble de séquences observées. La probabilité associée peut être calculée par l'algorithme *Forward-Backward* (A.5).

Annexe B – Références bioinformatiques en ligne

Le tableau B.1 ci-dessous liste l'ensemble des ressources bioinformatiques en ligne qui nous ont servies dans la présente étude.

Ressources bioinformatiques en ligne
Banques de données génomiques
GenBank (USA): http://www.ncbi.nlm.nih.gov/
• Génomes : http://ncbi.nlm.nih.gov/Genomes/index.html
• Données de gènes (download) : ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/
• Utilitaires de famille *Entrez* : http://eutils.ncbi.nlm.nih.gov/entrez/query/static/advancedentrez.html
o Utilitaire *e-Utilities* (routines pour automatiser les download) : http://eutils.ncbi.nlm.nih.gov/entrez/query/static/advancedentrez.html
EMBL (EU): http://www.ebi.ac.uk/Databases/
DDBJ (Japan): http://www.ddbj.nig.ac.jp/
BD et taxonomies des virus
ICTV : http://www.ncbi.nlm.nih.gov/ICTVdb/index.htm
Taxonomie ICTV : http://www.virustaxonomyonline.com/virtax/lpext.dll?f=templates&fn=main-h.htm
Taxonomie NCBI : http://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi
Génomes viraux : http://www.ncbi.nlm.nih.gov/genomes/VIRUSES/viruses.html
Base VOG : http://www.ncbi.nlm.nih.gov/genomes/VIRUSES/vog.html
• Sommaire des clusters : http://www.ncbi.nlm.nih.gov/genomes/VIRUSES/shagog.cgi?clust=PHOG&fam=all
Base COG : http://www.ncbi.nlm.nih.gov/COG/
Base CDD : http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml
Logiciels pour l'analyse philogénétique
PHYLIP (NJ, SeqBoot) : http://evolution.genetics.washington.edu/phylip.html
• Liste de programme: http://evolution.genetics.washington.edu/phylip/software.html
PAUP* : http://paup.csit.fsu.edu/index.html
• Tutoriel : http://paup.csit.fsu.edu/Quick_start_v1.pdf
T-REX - : http://www.trex.uqam.ca/
ClustalW : http://evolution.genetics.washington.edu/phylip/software.etc1.html#ClustalW
• ClustalX : ftp://ftp.ebi.ac.uk/pub/software/
Treeview : http://taxonomy.zoology.gla.ac.uk/rod/treeview.html
Prédiction de gènes
BLAST (Suite) : http://www.ncbi.nlm.nih.gov/BLAST/
• Documentation : ftp://ftp.ncbi.nih.gov/blast/documents/
GeneMark - Violin : http://opal.biology.gatech.edu/GeneMark/
PSI-BLAST : http://www.ncbi.nlm.nih.gov/BLAST/

Tableau B.1 : Liste de références bioinformatiques en ligne

Annexe C – Représentativité des génomes dans VOG

Le tableau C.2 ci-dessous liste la représentativité des 163 génomes de phages répartis dans 602 regroupements VOG. L'ensemble de ces phages a été utilisé dans la présente étude.

Nom d'espèces		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Long	Court				
Streptococcus pyogenes phage 315.1	3151	39538	56	26	46.43
Streptococcus pyogenes phage 315.2	3152	41072	60	37	61.67
Streptococcus pyogenes phage 315.3	3153	34419	54	34	62.96
Streptococcus pyogenes phage 315.4	3154	41796	64	41	64.06
Streptococcus pyogenes phage 315.5	3155	38206	55	31	56.36
Streptococcus pyogenes phage 315.6	3156	40014	51	29	56.86
Staphylococcus phage 44AHJD	44AHJD	16784	21	3	14.29
Aeromonas phage 44RR2.8t	44RR28t	173591	252	114	45.24
Bacteriophage 933W	933W	61670	90	76	84.44
Bacteriophage A118	A118	40834	72	26	36.11
Lactobacillus casei bacteriophage A2	A2	43411	61	29	47.54
Bacteriophage Aaphi23	Aaphi23	43033	66	14	21.21
Aeromonas phage Aeh1	Aeh1	233234	352	103	29.26
Acidianus filamentous virus 1	Afv1	20869	40	2	5.00
Acyrthosiphon pisum bacteriophage APSE-1	APSE1	36524	54	28	51.85
Bacillus phage B103	B103	18630	17	15	88.24
Bacillus phage Bam35	Bam35	14935	32	1	3.13
Mycobacterium phage Barnyard	Barnyard	70797	109	9	8.26
Burkholderia cenocepacia phage Bcep1	Bcep1	48177	71	44	61.97
Burkholderia cepacia phage Bcep22	Bcep22	63879	78	19	24.36
Burkholderia cepacia phage Bcep43	Bcep43	48024	65	42	64.62
Burkholderia cepacia phage Bcep781	Bcep781	48247	66	41	62.12
Burkholderia cepacia phage BcepNazgul	BcepNazgul	57455	73	6	8.22

Nom d'espèces		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Long	Court				
Bacteriophage bIL285	bIL285	35538	74	41	55.41
Bacteriophage bIL286	bIL286	41834	71	50	70.42
Bacteriophage bIL309	bIL309	36949	67	43	64.18
Bacteriophage bIL310	bIL310	14957	34	10	29.41
Bacteriophage bIL311	bIL311	14510	24	5	20.83
Bacteriophage bIL312	bIL312	15179	30	9	30.00
Bordetella phage BIP-1	BIP1	42638	48	47	97.92
Lactococcus phage BK5-T	BK5T	40003	65	39	60.00
Bordetella phage BMP-1	BMP1	42663	47	47	100.00
Bordetella phage BPP-1	BPP1	42493	49	47	95.92
Mycobacterium phage Bxb1	Bxb1	50550	86	48	55.81
Mycobacterium phage Bxz1	Bxz1	156102	225	12	5.33
Mycobacterium phage Bxz2	Bxz2	50913	86	53	61.63
Lactococcus phage c2	c2	22172	39	12	30.77
Mycobacterium phage Che8	Che8	59471	112	38	33.93
Mycobacterium phage Che9c	Che9c	57050	84	24	28.57
Mycobacterium phage Che9d	Che9d	56276	111	36	32.43
Mycobacterium phage Cjw1	Cjw1	75931	141	28	19.86
Mycobacterium phage Corndog	Corndog	69777	122	28	22.95
Streptococcus phage Cp-1	Cp1	19343	28	10	35.71
Mycobacteria phage D29	D29	49136	79	56	70.89
Pseudomonas phage D3	D3	56425	95	22	23.16
Bacteriophage D3112	D3112	37611	55	3	5.45
Streptococcus thermophilus bacteriophage DT1	DT1	34815	45	39	86.67
Bacteriophage EJ-1	EJ1	42935	73	23	31.51
Enterobacteria phage epsilon15	epsilon15	39671	51	21	41.18
Bacteriophage Felix 01	Felix01	86155	247	28	11.34
Pseudomonas phage gh-1	gh1	37359	42	27	64.29
Enterobacteria phage HK022	HK022	40751	57	42	73.68
Enterobacteria phage HK620	HK620	38297	58	43	74.14

Nom d'espèces		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Long	Court				
Enterobacteria phage HK97	HK97	39732	61	46	75.41
Haemophilus phage HP1	HP1	32355	42	26	61.90
Haemophilus phage HP2	HP2	31508	37	26	70.27
Bacteriophage IN93	IN93	19603	35	1	2.86
Vibrio phage K139	K139	33106	44	28	63.64
Acholeplasma phage L2	L2	11965	14	1	7.14
Bacteriophage L-413C	L413C	30728	40	38	95.00
Mycobacterium phage L5	L5	N/A	85	57	67.06
Enterobacteria phage lambda	lambda	48502	78	35	44.87
Lactobacillus johnsonii prophage Lj928	Lj928	38384	50	19	38.00
Lactobacillus johnsonii prophage Lj965	Lj965	40190	46	18	39.13
Bacteriophage L5	LoL5	N/A	8	1	12.50
Mycoplasma arthritidis bacteriophage MAV1	MAV1	15644	15	2	13.33
Streptococcus pneumoniae bacteriophage MM1	MM1	40248	53	33	62.26
Enterobacteria phage Mu	Mu	36717	55	14	25.45
Bacteriophage Mx8	Mx8	49534	86	10	11.63
Enterobacteria phage N15	N15	46375	60	27	45.00
Vibrio phage nt-1	nt1	244834	381	93	24.41
Streptococcus thermophilus temperate bacteriophage O1205	O1205	43075	57	51	89.47
Mycobacterium phage Omega	Omega	110865	237	36	15.19
Mycoplasma virus P1	P1	11660	11	3	27.27
Enterobacteria phage P2	P2	33593	43	39	90.70
Enterobacteria phage P22	P22	41724	72	32	44.44
Bacteriophage P27	P27	42575	58	37	63.79
Lactococcus phage P335	P335	36596	50	33	66.00
Enterobacteria phage P4	P4	11624	14	4	28.57
Synechococcus phage P60	P60	47872	80	20	25.00
Pseudomonas aeruginosa phage	PaP3	45503	71	6	8.45

Nom d'espèces		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Long	Court				
PaP3					
Sinorhizobium meliloti phage PBC5	PBC5	57416	83	11	13.25
Mycobacteriophage PG1	PG1	68999	100	17	17.00
Enterobacteria phage 186	phage186	30624	46	39	84.78
Listeria phage 2389	phage2389	37618	59	26	44.07
Streptococcus thermophilus bacteriophage 7201	phage7201	35466	46	43	93.48
Bacteriophage 77	phage77	41708	69	50	72.46
Lactococcus phage 936 sensu lato	phage936	31754	64	18	28.13
Bacteriophage phBC6A51	phBC6A51	61395	75	8	10.67
Bacteriophage phBC6A52	phBC6A52	38472	49	11	22.45
Bacteriophage phi1026b	phi1026b	54865	83	29	34.94
Bacteriophage phi-105	phi105	39325	51	22	43.14
Staphylococcus aureus phage phi 11	phi11	43604	53	36	67.92
Staphylococcus aureus phage phi 12	phi12	44970	51	40	78.43
Staphylococcus aureus phage phi 13	phi13	42722	49	45	91.84
Bacillus phage phi29	phi29	19366	62	29	46.77
Bacteriophage phi3626	phi3626	33507	50	20	40.00
Phage phi 4795	phi4795	57930	48	39	81.25
Yersinia pestis phage phiA1122	phiA1122	37555	50	42	84.00
Lactobacillus bacteriophage phi adh	phiadh	43785	63	33	52.38
Streptomyces phage phiBT1	phiBT1	41831	55	14	25.45
Streptomyces phage phiC31	phiC31	41491	53	14	26.42
Virus PhiCh1	PhiCh1	58498	98	12	12.24
Pseudomonas phage phi CTX	phiCTX	35580	47	28	59.57
Bacteriophage phiE125	phiE125	53373	71	29	40.85
Bacteriophage phi ETA	phiETA	43081	66	42	63.64
Bacteriophage phig1e	phig1e	42259	62	22	35.48
Bacteriophage phiKMV	phiKMV	42519	49	11	22.45
Pseudomonas phage phiKZ	phiKZ	280334	306	8	2.61
Bacteriophage phi LC3	phiLC3	32172	52	41	78.85

Nom d'espèces		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Long	Court				
Staphylococcus phage phiN315	phiN315	44082	78	40	51.28
Temperate phage phiNIH1.1	phiNIH1.1	41796	55	40	72.73
Staphylococcus aureus phage phiP68	phiP68	18227	22	3	13.64
Staphylococcus aureus prophage phiPV83	phiPV83	45636	65	54	83.08
Staphylococcus aureus temperate phage phiSLT	phiSLT	42942	61	45	73.77
Bacteriophage phiYeO3-12	phiYeO312	39600	59	48	81.36
Pseudoalteromonas phage PM2	PM2	10079	22	2	9.09
Enterobacteria phage PRD1	PRD1	14927	22	1	4.55
Methanothermobacter wolfeii prophage psiM100	psiM100	28798	35	9	25.71
Methanobacterium phage psiM2	psiM2	26111	32	9	28.13
Pseudomonas phage PsP3	PsP3	30636	42	35	83.33
Staphylococcus aureus bacteriophage PVL	PVL	41401	62	48	77.42
Bacteriophage PY54	PY54	46339	67	19	28.36
Bacteriophage r1t	r1t	33350	50	41	82.00
Enterobacteria phage RB49	RB49	164018	274	115	41.97
Enterobacteria phage RB69	RB69	167560	273	126	46.15
Bacteriophage RM 378	RM378	129908	146	10	6.85
Mycobacterium phage Rosebush	Rosebush	67480	90	14	15.56
Enterobacteria phage Sf6	SSV1	39043	33	20	60.61
Streptococcus thermophilus bacteriophage Sfi11	SSV2	39807	34	23	67.65
Streptococcus thermophilus bacteriophage Sfi19	Sf6	37370	66	45	68.18
Streptococcus thermophilus bacteriophage Sfi21	Sfi11	40739	53	49	92.45
Shigella phage SfV	Sfi19	37074	45	42	93.33
Sulfolobus islandicus filamentous virus	Sfi21	40900	50	45	90.00
Roseobacter phage SIO1	SfV	39898	53	41	77.36

Nom d'espèces		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Long	Court				
Bacteriophage sk1	Sif	28451	72	3	4.17
Streptococcus mitis phage SM1	SIO1	34692	34	7	20.59
Enterobacteria phage SP6	sk1	43769	56	16	28.57
Bacteriophage SPBc2	SSVK1	134416	31	20	64.52
Bacillus phage SPP1	SM1	44010	56	31	55.36
Sulfolobus spindle-shaped virus 1	SP6	15465	52	12	23.08
Sulfolobus spindle-shaped virus 2	SPBc2	14796	194	19	9.79
Sulfolobus spindle-shaped virus Kamchatka-1	SPP1	17385	101	18	17.82
Sulfolobus spindle-shaped virus Ragged Hills	SSVRH	16473	37	21	56.76
Salmonella typhimurium phage ST64B	ST64B	40149	56	38	67.86
Enterobacteria phage ST64T	ST64T	40679	65	42	64.62
Stx1 converting bacteriophage	Stx1	59866	177	152	85.88
Stx2 converting bacteriophage I	Stx2I	61765	176	152	86.36
Stx2 converting bacteriophage II	Stx2II	62706	182	155	85.16
Enterobacteria phage T1	T1	48836	78	19	24.36
Enterobacteria phage T3	T3	38208	55	47	85.45
Enterobacteria phage T4	T4	168903	278	135	48.56
Enterobacteria phage T7	T7	39937	60	40	66.67
Mycobacterium phage TM4	TM4	52797	89	23	25.84
Lactococcus phage TP901-1	TP9011	37667	56	45	80.36
Bacteriophage Tuc2009	Tuc2009	38347	57	52	91.23
Lactococcus phage ul36	ul36	36798	58	44	75.86
Vibrio harveyi bacteriophage VHML	VHML	43198	57	21	36.84
Vibriophage VpV262	VpV262	46012	67	3	4.48
Bacteriophage VT2-Sa	VT2Sa	60942	92	79	85.87
Bacteriophage VWB	VWB	49220	61	7	11.48

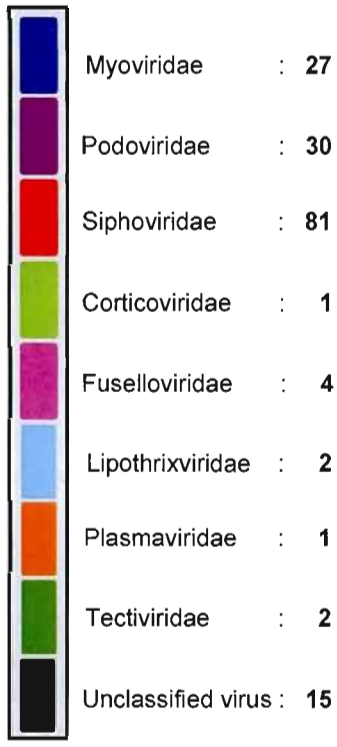
Nom d'espèces		Taille de génome (nucléotide)	Nombre total de protéines encodées	Nombre de protéines assignées aux clusters VOG	Pourcentage de protéines dans clusters VOG (%)
Long	Court				
Enterbacteria phage WPhi	Wphi	32684	44	40	90.91
Xanthomonas campestris phage Xp10	Xp10	44373	60	16	26.67

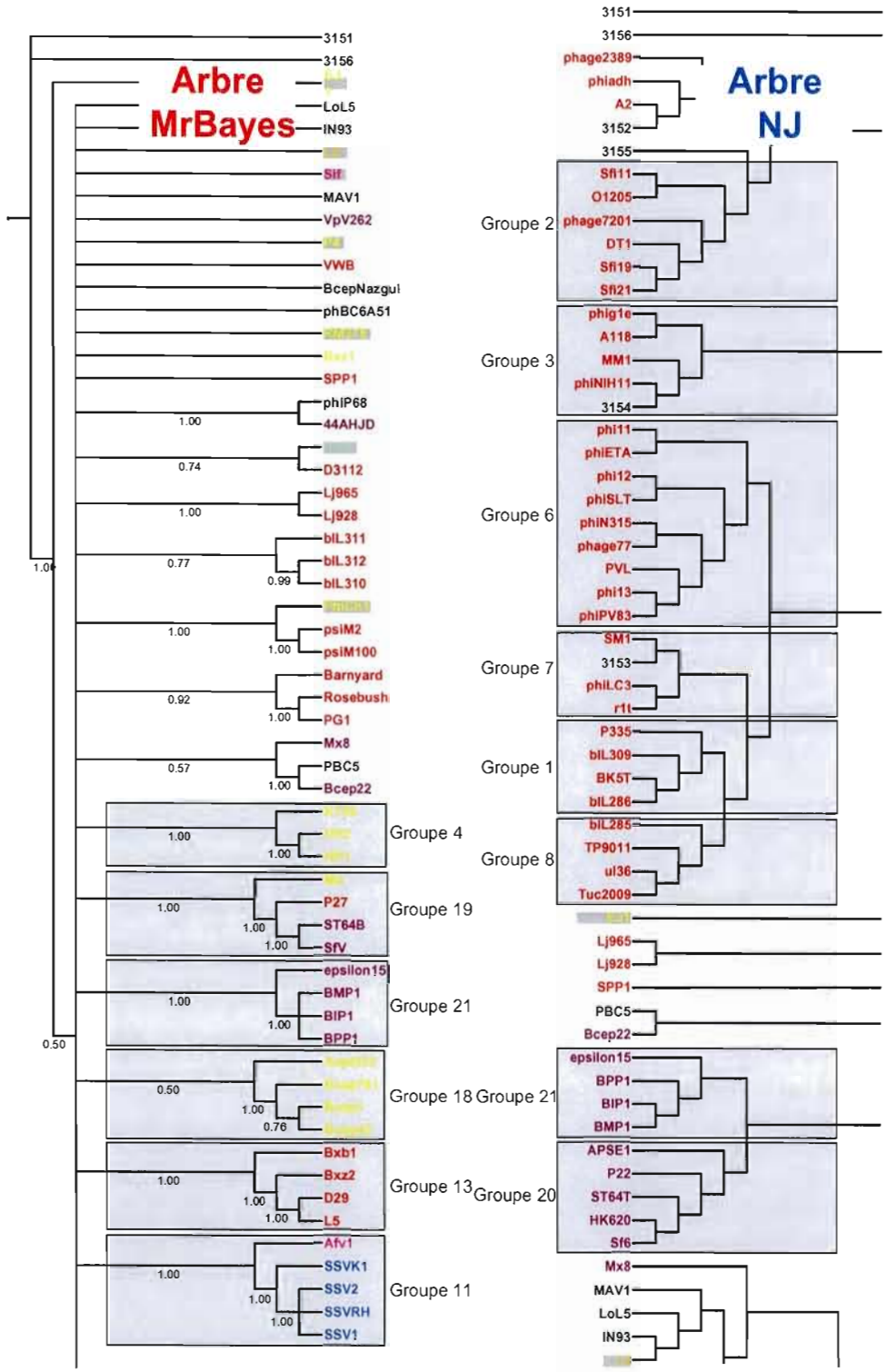
Tableau C.2 : Représentativité des 163 génomes dans 602 VOG

Annexe D – Détails sur l'identification des groupes de clades

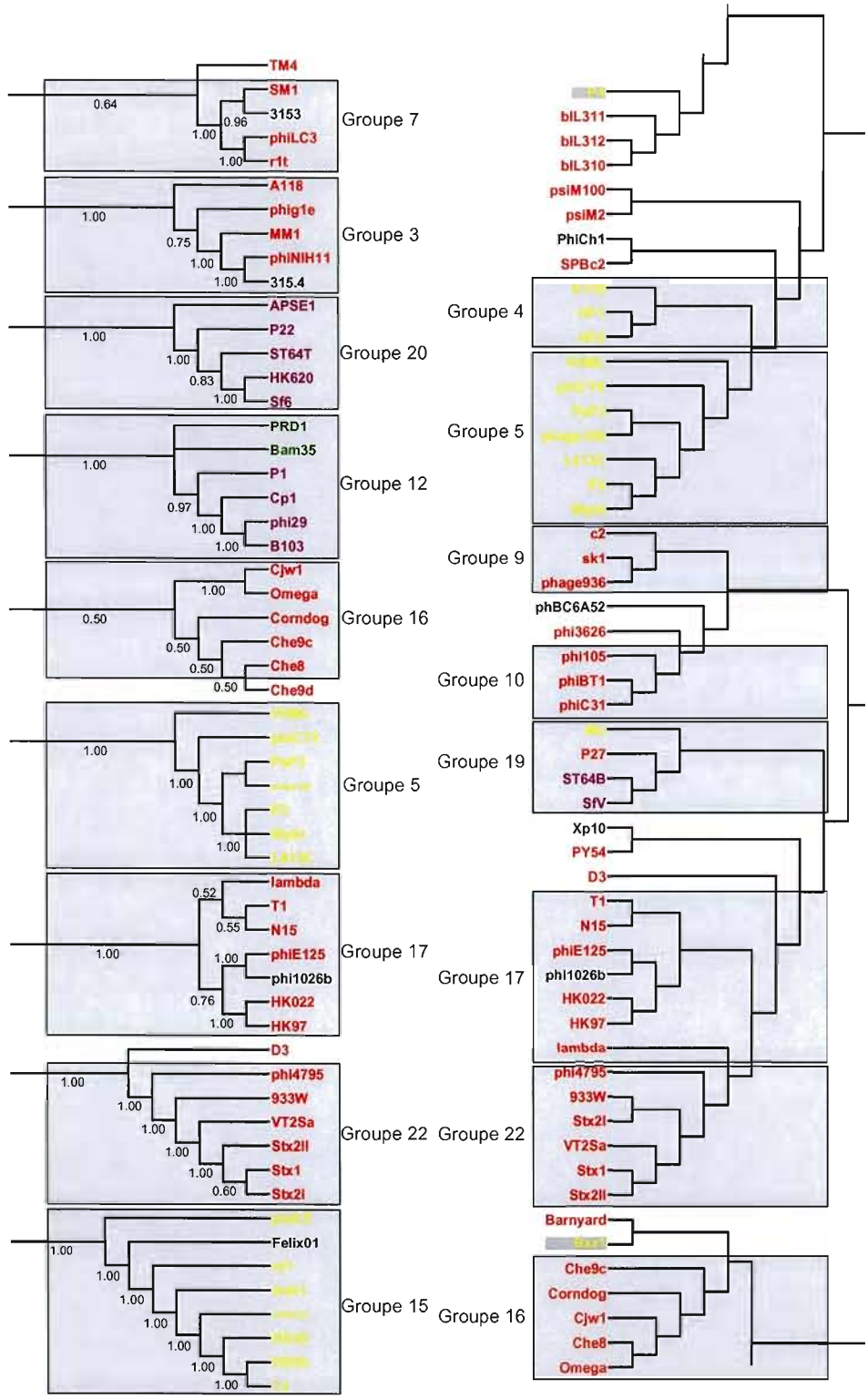
Les 3 partielles suivantes d'une même figure (i.e. figures des trois pages suivantes) montrent la confrontation de deux arbres d'espèces inférés par MrBayes (partie gauche) et NJ (partie droite). L'arbre inféré par MrBayes est présenté avec des scores de robustesse alors que celui inféré par NJ ne l'est pas puisqu'il n'a pas été soumis aux tests de bootstrap. Avec des tests de bootstrap, NJ n'aurait pas donné des clades consistants et donc ne pourrait être confronté à l'arbre généré par MrBayes.

Note : le code de couleurs ci-après correspond aux espèces identifiées dans la figure qui suit. Les rectangles de couleur grise signifient le regroupement d'un certain nombre d'espèces entre elles dans des groupes numérotés de 1 à 22. Les 22 groupes comprennent le même nombre de phages et d'espèces dans chacun des groupes.

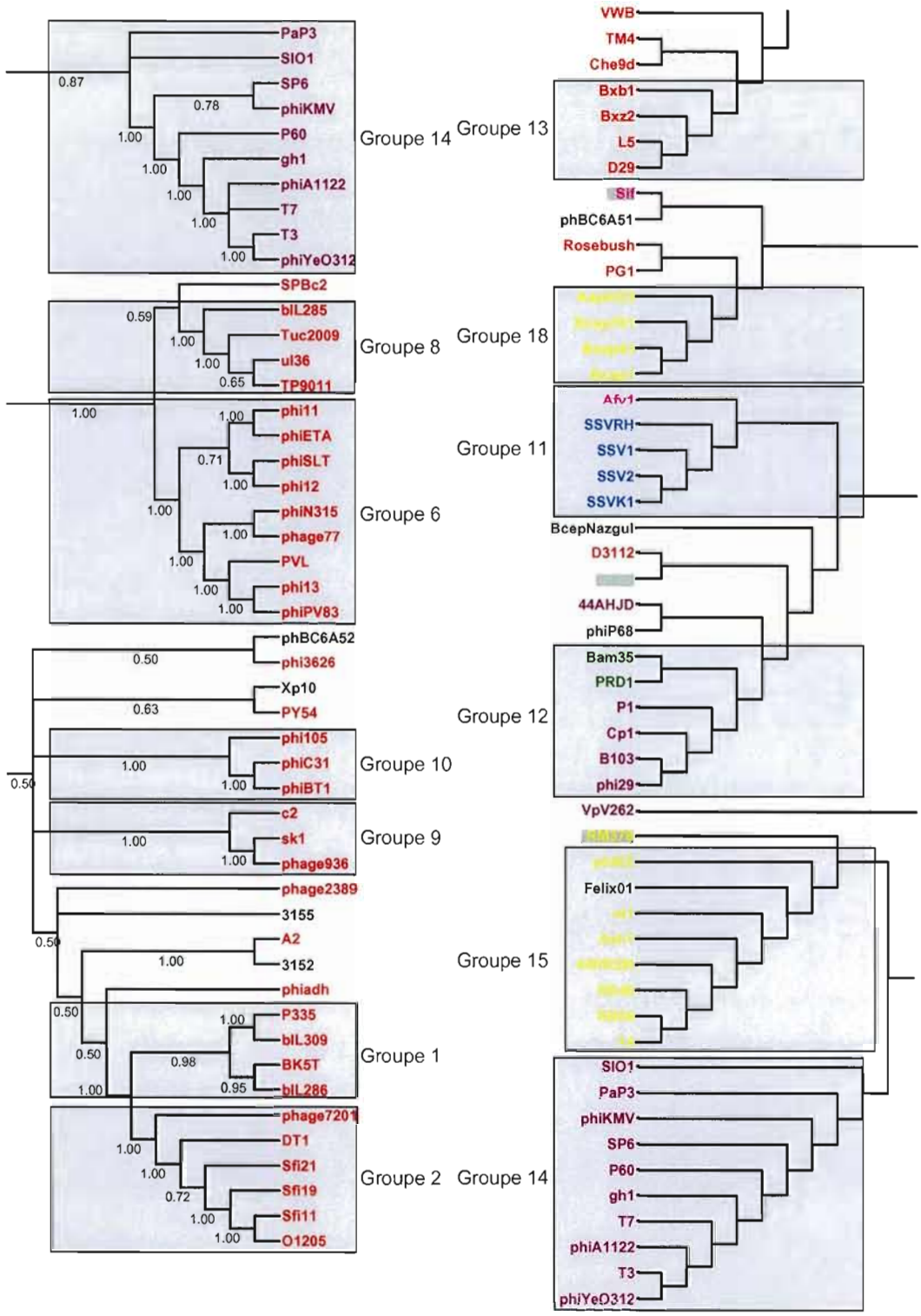




Partielle 1/3



Partielle 2/3



Partielle 3/3

Annexe E – Nœuds internes ACP et séquences de protéines ancestrales

La liste E.3 suivante monte les nœuds internes ACP de l'arbre d'espèces associés aux séquences de protéines ancestrales générées par la procédure de reconstruction ancestrale grâce au programme *Ancestor*.

Nd Int.	Fonction	VOG	Séquence ancestrale
0	[R] Repressor	vogp0367	ISMMTTFERIKELCKKQGISLTKLEDDLGFCNNSIYSWKNNNNPTSYDKNQKVANYFNVST DYLLGCEEAKRTDNPTITYNNNISYYTSYZDDIPQEANTIAAHIDNDSTEEDMKKIZDYIEFI QHKHKKKLNENN
0	[L] Putative helicase	vogp0693	VMAPKQVAEDTWPAEVGKWNHFNKVSLVLGSPEERDDALKTQADNWLVDHYENKWPF NLVVIDELSSFKSNSYTRFKDHYFKPNQHVCZNIYYPWELRDCSQEEIYKNIEDICLSMKAK DN
0	[L] Recombinase	vogp0240	MNLVQPIRDSDQIKPLKDYLKEKSQRNYILFILGFYSGLRISDILPLKVRDVKGRDHIAIIEKK TGKQKRFPINPTLKKELKKYIKDKELKEYZFZSHYNHNRPIGREPAYKILNEAAEQLGLHN MGTHSLRKTFGYHKYMQTQNIAMLMEIFNHSSPDVTLRYIGINQDRMDNSMTHFYI
0	[L] DNA polymerase	vogp0258	MQDQLRLDLETYSQDDLPNCSLDIYSDHESTEILLFSCSFNGGPVQHWEITDGKFPPFHADH ADVSEALVDPNVIKRSWNSQFERIISRRYLNIHTPYKPNWHCTVAQAYVLCLPGSLDKVCD HLGLPQDKVKDATGKKLINLFSKPRNRQTNPYEWHNFLIYNHQDIMAKAAIDKWLAPYPT PDKEWDLZELDQFINNRGIAVDMDFAHSATALEDRSKEELVRQMHAITGGALENPNSPHQ LKPWVKARGYSK
0	[U] Uncharacterized	vogp0260	MKNLNLASKPYPFHYGGTVITNNFILIGNHCSFIPRELNGDHTSDGDDDLIKAVLEVFNTEZ DPKSAMVKALHKVEESQZDLQETKQKSEQAKVTAKATQKEAKSLEVLTNKSHKMVHLQA MHSLTSSTKVYPNIYKGLLELIHPAKQGKNQAYDIFTMVDPKHEEQGNEGNLVLIQVNQPF TYKGQTLKELEGEDKVTIIKYADLVKQDPQN
0	[A] ClpP protease	vogp0261	PNFWAKAQHKMRMMNNINLKGHIISNNYREVYNYLGMEYHNISPKTVNETLDKANDKDII LEINSNGGYVFAGSEIYTTLKDYKGKVTANITGLATSTASFITMAGDNINMSPTAQMMIHK ASSNSMGSNSNDLDQDSKALKSFDKSMVNAYTPKSTGMDEEELLHLMANETWMTAKEAI DKGFADKIMFLNDNKPDFVTSIDSMSMPHDIINFVNSNNKNITEITNKSLQNKKEDILLDZH LQDNMAILFIKDN
0	[A] Holin	vogp0263	KMMINWKLRLKNKVSWLAIISAIFLLAQQFGNFFGYDLLDFPNNITDVINTILILLVFLGIVT DPTTKGIADSNQALNYKEPSDEKEGKLQSSWEHSQDSHSNQESYHDAPKDYEASQHSSYA SPDECSPEYESPLGCGEVNSN
0	[U] Uncharacterized	vogp0289	MMTNMQELFQEZGVYLVYFTTSZWNREGFYYPDYNIIYINRNLSNKDHKKVILHELGHFD RNPKHYKCFREKRNWYESQANRHVIHQLLKDELALCEDMDDFNPNHFMEKYNLTTKNDQ VMVKEEYFNLIQTII
0	[U] Uncharacterized	vogp0301	QIKTSVKRVGIDTSDKNLGKNTSLDDVNKVFZKNTTNNNNKTQNLAPVNTGYSKRSIYFKF THGVYSSYGNSTSPPTTYSTYMEFGTRYMDAQPFIKPAFNKPNMVLVKDLENLFKPRN
0	[A] Portal protein	vogp0315	VSIFFNTFKNSRNLTYTVCCSITSEKLKEVFIKEWAMDSCINYMARTISQSNFQIZENDKVTK NQNNYRFNIKANPNQSSTQFWEKVIYKLIYDNEALIFILHCKDIYVADSFTQDEZTLYDNIFK DVTFKDZTYKKIFTVHQVIYLKYNNNNVTSMVESLCEEYGKLLGHIINNQKKTNQIRIIMKS TAAIGHEHDVEKSQGFTQPKSNKDFLNQAVITLPIEGFIDIHVSASSNNIDKSSNNTSTIKYY VNKN
0	[L] Terminase	vogp0324	MGKVKSKLTPKQKRFTDEYIISGNNATQAAIKAGYSKKTAHSIGPENLTKPNIKDYIDERLE KMDDKNIMDPEEVLEYLTRYIARGKEKEKVFNTYRRFVHRYHNNZIDNGEDVEKNNIANI NLGAKDCIKALELLGKRYHDQLEEDLVQDQIKMFTDKVEMDVYKDITINIGDWDDEESET LDKVHGKDHDGVDNRQSNSTN
0	[U] Uncharacterized	vogp0173	MTLTTDQIHNLLGIDEAFKAPNRLMEILFDKZDREDLFRQFLKYEKDMSYDWFMNYFEEE QAVTVNQKQDFTPKSVSTSSSKIVSCNRYYZAAVGTGSILIQAWQDHRMNSYPFTYRPSNY WYQVZELSDKAIPFLLFNMSIRGINGVVIHGDSLTSQVKNVYFLQNTKDNTLSFSDINVMPH TQDIKQEFDIKEWIGQAIEHIESKLIDWIPL
0	[L] Terminase	vogp0359	SMIVNPYFDSTPQZTTKILKSKIKTSKKIKKACKNHLKNLHYHZKKKNLNZNLKKKKAKKK IKFIKLIPKTKASCNNKLDHWQKYILTKISSRKKKENTCKNINNFKKTFIQMTNKKSKSIZISS ITKYKFSSKDYLETTNHFFITNLKZNQAKNFLNQTKNMTKKSKKLNSNIKKTKKIKKNCRA YKINNFTIYZKIKSSTKNTNPIKSNNPPLTILKKFQTQKKKKJKKIFKSSQVSRNQPLIFIITTTG CDL
0	[S] Minor structural protein 1	vogp0170	MVKANIFPSGTTQTZKFNYLIIHYDNELTTNEVMNYFIZKNQDQLEHFZECKNYYZGHHNV ZNKGKDNKVIDCPNPKDNYKADNRITLNFPKYIIDLNNSYLLGNPIKYNYNKEILDDINYLV NHNNLDDNNNKLVKNLSIZGHAYZLLYLDKDDHKNNLIHFPPKQTFIIYNDTIQHNSLVAIH YYYNKDSTREDNHNIKSYTYYINYNYDYINZDYHSTALKDCTDHSYSEIPIIHYTNNKEHPII FDSIINF
0	[S] Tail protein	vogp0381	???
0	[S] Putative structural protein	vogp0393	IPILLESQNTNFDNDGLGPLEDSFNGNITQNGNCHYKLTFEYPVNRDLRPLIKEGFIFEANAK QGKDLQTFQVYIIDINKSNSYINIYANHVAZDTTSNTMDNFSVDAFAGTAFPYELVGNTYH LRNFCLDFNSGISNIDNFSSTADZDDNVLDTIAGVKGSSNPRGGELVLNNYYINLLKHASKD NDTFFEYSNNLTCFEDKDTTEGLITSMYPFAKDEEHDREEPDTVSDQGFIHHEDSKENSNFN VQDSYYI
0	[U] Uncharacterized	vogp0396	MPTIDNGLDZIKKYLKENCIKKTYLATTYGIHQQDVTNILSGRIKAPKANRFILKVIQDFNLH

Nd Int.	Fonction	VOG	Séquence ancestrale
0	[L] Terminase large subunit	vogp0398	LMTTKHRTKNEMNVDRLRKVFDKHVNNTLSKYSNFAEVGYGCESSGKCHGVFEKVILKV LNPEWKDPTEIMRLRNVRSPIZECVFADIDADLTNCGVLDEZKINGSNINVILPNRSDIFLFE VMDNPKKIKSIEGVSRNVVLEEASHRDQHDYAHZTMRLRVTKHREKQIFSMFNPVSKLNC VFESFFDNAPTTDYKHNIVHQSSYEDZRFLDDVTLESMEDLTNANQAYCEIYPZGDFADLD KLIFPESNNRIL
0	[U] Uncharacterized	vogp0401	MIPKFRAFDKKTKKMYCVDGFKSSERADDEFRSGRLETFHFVEDNLDDYILMQSTGLKDK NGVEIFEGDVVKLQNHYSMTSFLEFFKVNHNIRWTVNFRAGSRRIYNRCRGSYSCSRNKDC EVIGNIHENQDLMEAVQETLEN
0	[S] Structural protein	vogp0500	YKFQNTTKQKHYCTLTFILTCKVSYDVSKSEDLTYDSLTLEVQGRDMYSFQIDKVAGQDG GFITNETZPPRELTINFNLENCNPSSLRQZVYNLKALLIRHEDVPIIFSDDSDCTYZGRFKTAT NVDEKGSSIIFSFNIFCYDPFKHGPNQSIKNNVIKDLGZAVTNPIISFNLLTSAKGFNLLPTDG HNRFKSSQAKNGFFLEIDFHTGNITLNGQDIMDSLDMDSYCNHMRLPANTNLZSSNSTITIQ SRKVF
0	[U] Uncharacterized	vogp0643	MSRDPTLTLDESNLVIGKDRFHYTFTAEKYNPKVRLASKCLGTTHFNQLMIEHGNKATNY VTPMVINRNPTGLFKDLKELDNELTDTNSHLWAKIKLNNQGMLQEYSNSDIKTEIFNSAQG ISTHVSYVNDKZZSZFKHTIKGIHHKFKSNYNQLASSFDSSIZGLKANLSNZAKDLHALFQA SAQGLCQKYNDSQNESSTNFHTTTTGTKVAYVSQTADLWAZLITTTQGICSELDHKJHGLQ ATIQTTSS
0	[U] Uncharacterized	vogp0685	MKLDISIPTHAKLSPSSAHHWLNCPTSINLEADFPYKSSLFTQEGTSAHELSELYLTZKQDGF TQCKFNNAIQDYKGSNHYYNEELREVTEEYIDNVQEYFNNVWSLDNNVKFLLEKHLDFCD FVPEASNTSDVIILSSGVLEIIDLKYGKGIKVSANHNPQLGLYALGAZDZFDLVYDFNTIRMT IIQPRLDHFSTWDMPIYHLLQWGTLDVKPMAHQANKGAGQFKACSQGHLRPPVFSKJEHH CRFCREZIH
0	[U] Uncharacterized	vogp0688	TKVISSKVRLSYPHLLEPNPVQGGREAKYSTSFLIPKSHTATIKDIKQAIKAANEENRGNKNE GMNQASFKNPLRYCYGEWEAKEYHGYPGNLFMSASSKTRPWVIDQNNTPLTSQNNLYSG CYINASINFYAYDTNSNKGISACLNNIQFFHKGDPLGGGSNSEEDFDELNKDEEEDLLVS
0	[U] Uncharacterized	vogp0690	MRESTMEKYLVZEISKMGGLGLKFVSPGTSGVPDRLVILPKGNTFLVEFKAPGEKPRPYQM REHNQLQNSGHQVCVLDSQEWVNIFLAVIGSWFN
0	[L] DNA primase/helicase	vogp0352	MGVNTKTENLIIHPHPWQAWQAIMGNFLFDHYLTFNYPNSDVLDILKPKDFYRPNHHKIFK FIIDLDSNNNPFDFMTMANDLETQDMLAQVCGFTYLAEIANNVPSSTNIKDYAKMVPKYD LQRYFINMYKDCKDFIYTHNIITSZNNIKDVHAIMSFFSHYYNTFESFSLVHFWEVMSDCFD KFKZRFQLSDKFNSISTGIHDLDNITGSKGLDNGYLVILGACPSMGKTTLALNIAKNITINDM PSSNQPVLF
0	[A] Hyaluronidase	vogp0857	MTENIPLRVQFKRMTAEEWARSDVILLEGEIGLETDTGYAKFGDGKNRFSDLKYLTNPDNN AFAQKTDTDGQDTDVKIAKLESNKADKDTVYSKAESKIELDKKLSLTGGIVTGQLRFKPNS GIEYSSSTGGAINIDMSKSKGAAMVMYTNKDTTDGPLMILRSDKDTFDQSAQFVDYRGKT NAVNIVMRQPPTPNFSSALNITSANEGGSAMQIRGVEKALGTLKITHENPSVDKEYDKNAA ALSIDIVKKKKG
0	[A] Amidase	vogp0197	LATMTQIQANERSZTYZGNGFNIDCSZGFQCSDLANTCIKYLFGINLWGWGINKVPSNNDQ DFEVLATVYKNTPSFMAQFGDFAVFNPCSHTSIGHVDLVSNTTRDNNINLESNWZGSGRAI NDDCHVSGPDHATFRTHSYNDRTLFIWPNFPYQSYTNSKTTSTQPPTKDNLNQKAKDSNIK FEVAHINSWHNGDFPCCASZNCEGVSSPKYAITHTYYSFDYGNGYYCMSSLANSCMCIYLA CCDTYHDCDHN
0	[L] Integrase	vogp0015	???
0	[R] Anti-repressor	vogp0072	MKELQDFNFNVRSAFHVHTIVVGKEPLFFGKDIAKVLGYTKLHNGMDKHIRDKFKCVSHF TTPSGISAGRDVTFISEPSLYKLVFKSNKPRNTAEEFKDWVAKKVLPTIRKHGAYMTNSKLE EALLNPDTFINLATQLKEEQEEDFGLQSENSKLGSENEIMKPKAYYFDSMFESSNIISLTQIA KNYGFCHNKLTRWLVETHFRYKVKKKRVMPSRQKDKDFFEFKDSKDVRSTWQEVNTLIT PWNQKASKFLL
0	[U] Uncharacterized	vogp1128	MDLTLKNKDLNTLYSVLDKIKVTNMRANRGRAKLLAKVVDKFKEYAKDEGDLIDLYAQK DEDGKFVIDEHENIKLADPAKLDELNDLLNELADEDIVIKGGEYSKRFIDFLEYLAESEDEFT SDEIIIIDNILEQFEESKKGEKN
0	[U] Uncharacterized	vogp1155	MNKLELFLLATTIILAIIARVQYEVIKKHNSPENKRRIFREVALENSKGWSEKRSRGEVVS
0	[A] Putative protease	vogp0773	MKKMFLILVLAPTIFLAAGCYYVPAGYVGIKVHSLGDNSGVNFEVKGPGRYFNGPNMDIFI FPTFNQSNTWDKANDKSFTFQTVEGLSIDTNMGVSYTIPHKKATKVFQNYRRGVDDITDTY LRAIIRDALNWAAANMAIZEVLGCGKADLQQQVZKDVQANTAKVGIZVVZVSFVNQMG WPQKVMNSINGKITANQIAQQKENEVKAAKAEANKQLAKAKGQAEALEVKAKAZAETTH IZEQTAIQNWDGMZZE
0	[S] Major capsid protein	vogp0745	MAITVKKSPKSSNATTEFFNTINEGATQEEQDKSFNDMVNQMRNEIMAQAKEEVERMFNL HNSAHALTSNEFKFFKDVNKNIGCKDKAKLLPQETMDHVFEDLTTKHPLLTDINFKNTNSR LKVLTAETSGTAFWGNICGEIKGQLDKAFNEKNTTQYKLTAFILIPKDMLDLGPTWLEQFIR TQLEEAMALALEAAIFKGTGKDQPIGLMRDLHKATSGKHTVHTTTTTZKDTTTATTASTVM DPLTHVZZN
0	[U] Uncharacterized	vogp0714	MNKRIKKKRKLETAVVLLIAENAMQAEAIKNQNKQIAELRAIVQQNAQATNRELATVKAA TLDNQSVIKSIGDGVDYIKKNYKRKWGK
0	[U] Uncharacterized	vogp0702	KAKQLIFSFGSFILPIITSYYFFKAKSIKQDTHITSLQZEVEHLKMFNHQNTSRLEEHDEQNKT LMALTEQIKNLSQEVRELKDLMQGKTN
0	[U] Uncharacterized	vogp0695	KDTKDTQDNVNDQPSANQQPFNNFSSHSKMDTFVHKNLAKAMDTARNNWEKEQNEKAQ KAKDNAEMSPEERAQQELEKREDAFTEREAEVTZREMKSEAHTQLITDGLPIZLAZMFHYV SANEDNMTNTYNDAFHKAFHKAIEKQVEESFNPPTPKGDTSTNSSTKELITELADNTQNVK FIFKNIS
0	[S] Major tail protein	vogp0115	MAAZTAKVGKDIIZLFRFFNAATKEAASKLAFQTEHSIEKTRDYNTTDTKDGTISASGSIEYS LSVTSIATNGDPRLDEMEEAFNDGKMIEVWEINTAKKGSDGNNSGKYKAKYFRAYLTSFZ YQANAKDTVELSFEFGVIGKPQKGQATSPNKEQVKVIQYIFKDTIPGHTMGGDHSTATPPP K
0	[L] Putative primase	vogp0156	NGWDVISHQDKPIPDRINFSELLDSFGYIEGNHFSCCHQVPSHDFIGHVIQFNYHDQAYALH NFNDCNCWNVVNPSNRFGTDIGSSSTANDVTDLDAICCHFDVZSSACRNFANTCHFIEKLSI YVGQGPTNVVMAGFETDNLLPHWEIKDNRCVPCTRHTVIPSMQATSDEVDAZATDVHNIR GHZVVNDTQRKGAKMDRGVYNLAHVFRVHISYNNKGQPIMVNCNDSCGGPFSINDLTECF NEGGFNYRDGH

Nd Int.	Fonction	VOG	Séquence ancestrale
0	[U] Uncharacterized	vogp1134	MLTYDEFKQAIDDGYIVGDTVAIVRKNGQIFDYVLPHEKVRNGEVVTEENVEEVVVELDKN
1	[U] Uncharacterized	vogp0505	MTFFPEIDEKKTKKNSKRKLREYPRWRHIANDSDNQKVTPAYSFMPRRPCSGPTKPVENLAVRRVDAZKELEAIEQAVNGLZDPDYRRILIEKYLASHPKPYNWDIYKELGFYESSYZEMLDNALLAFAELYREGKLVVEKGVLSCNCN
1	[U] Uncharacterized	vogp0946	RDZVNNCLSFSHLKAFQECQAQALAFMAGHWAQQSPSTIKAMLVGSYVHAYLESNKAPEEFKLEHDSDMFAKKGNNKGCLKSDFQIAEKMVHTLKHDPTFKTIYQGNGHKELDIFGKIEGIEFQGYLDCLNLERDNNIDIKTIKGSIRDKEWSEMENKVNVYWIZAYSYLWQMAIYQEILEQSDNKKMGKNISPIIYAVTKETLPDSNTITIHNEEWLDDSLDKLGQSINKZMDNINGRKDPNPCDHCNYCKATK
1	[R] Cro repressor	vogp0539	KKTTPKKTLKDLRTNYNLTPKEVAHKFKIHHQTILKLEKDSTNIPISLLFKLAHFYNFNIDYIFLGKKYEFNHFLDNNZRCILN
1	[L] Terminase large subunit	vogp0526	ITMVMSQPFAQNQLEVLSCFRNYSQIQDQEVFISHSTKHPGKTFFVNLAFIVZLPTFHDNRKFMAVDKAIRFFISTSZASMHVNIZYEGYNNYGCHSNDVNYSSFVFRGNKFFLFDSKHKCSQRLAHGFTSSGFFINEASLTRQSFVKEVITRRSIRGTRMFLDPNPNNPNHWLKNNYINKNKNKYIYYCLFNLDHSTFLCKDIIDSFNATNPNWDFYNRNIZGLWILAKGVIYNDFYKNIHVVNQLPNKDRNFC
1	[R] Repressor	vogp0366	FZDKFIMKENTSKRLKQIMKERNLHQVDILDKSKPFQKKLGIKLSKSTLSQYINCNPSPDQDNLFLLSKTLGVSEAWLMGYDVPMVRVNPSKMENDSETIZETITVMKKLEEPRQKIVLDTANIQLKEQEEQKSKVEYIEDZRLSHDYSZDHISKDSDZDWDQWISHDEDFFKDZVKNTLKEDIRKGDLGKCMGPMLKQGTYAFIKKEDSIKDGTIALVVLDGPQEVSLIKRVDICEDCYINLVSLNPN
1	[U] Uncharacterized	vogp0356	NNKCDKLIMVSDYEVIKPENMDNVNKPSHYNCCKNESIDFIHNFIGKZPSKSACSRGNAIKYLSRFQKKNNGEDLKKARYYLDWLIEZMEZENQNI
1	[U] Uncharacterized	vogp0331	MNIKALIKKYEEZWNEHSPFYEPVPYTSMVELFLKELKQLDQPKKVKVPHFVADWIEEAKKDCKDFVZLFEFDFTNQEVRNFZCKWFMQESPFHLLARAWLDGYEVEEEKRYSVTLNNPQPLVKSQSCSTQNIFFFLYFSQDINARNYKGTHKELEEANFGWVFHCQGMEIEEVE
1	[U] Uncharacterized	vogp0264	EIIMTTSLKLANQFIIFSILFSVCLVIYHTDYIQVTKETKPIVIYNADNICVKMLGKVTDKGIVGKLYTLTIRAYGKFLVTKEQYKSIKVGDKIPSYLKGRGQZN
1	[L] DNA replication protein DnaC	vogp0083	MDSTIQZTANGIPQSHQHNMIKYGQICQKHSNTIFHIZVHHASFCHYPQKNYKQPKHZIMDDNNYNTQZSYKNIYZFTNYSFIDFTFKDANLDNFKPNNHKSAQNLNFACHFSQNCSNQMNTNNFILTGYPGTGKSHLAFSIAKDFNTNNHSNTIFFSNITSINYLFNISCCKDSQQTTDELINQZSSVQLLVMDELGIQNNSYCZYTILYDMVNHRNKSIITTNLISNEIFDNYWAFFISNFLDRTNNGNINNF
1	[U] Uncharacterized	vogp0544	TWQKVKDLLMKKNMTQKALSKKAGLNHNTLRNFKNEHNKNTFKNMCKIADALGISLDELRN
1	[L] Putative recombinase	vogp0222	MTKNNQLTQKQITYHVSSRIREMQNQGNLKIPQNYSPSNALYSAYLALKNSSNICTQDSIYNALLDMVTQGLYPAKNQCYFIPYGNKVKLTRSYFGTMKVVKQLPEVKRNYAEVIYKGDKFQFKGNKGRRKVFVSHENDLVNTNAENTIGGYYSIIYKEDCQNFMTVMTNKEIDKSWAQAQTNNDQNNFHQDKANRTFINQTAKQTFINSDHKDFFNDTVNRTTVNDYDKEPQENQAEPVSEAVVNLDDILKAN
1	[U] Uncharacterized	vogp0949	MPMANEENLIPNTERTKSEPREITYKGGMASGKARRKKADLKKAFETVLHSDVTYPKVKKQLEDMGLDTTNQTALALVVIQQAMKGNLRAFEQISKLTTNAKDSLDNQEQKERIKASKLGNEKLRKHIEASKVNFEADEVIKSWAGVVKDTWEDLEN
104	[S] Putative baseplate protein	vogp0414	TNTMSNIPELARTIHNLIRSGTITEVDNARRVRVPTGDLQTTWLNWATTRAGRYRKWKAPALGEQVLLLSPSGQLANGFVLTGLFSDDNPPPSSCSDSRHIVFHDGSVIEYDPETHHLSVRGIETNPIVLVMADTTMTIDTPEVISVNNFTNPSFEVQKGSQIVTDKVSISGNFISNGIPLVGHTHGGLHRGGSSTDDTKK
104	[S] Baseplate or base of tail fibre	vogp0416	TTNFATIDLSQLPPPHVVEQSDFETILAERKATSISLYPDIARALALESEPLAKSLEENAYRELIWRQRVNEAARANMLASAKGTDLDHMGANYNVKRLVIQPGDHSAVPPIPDVMESDDSYRZRIQMALEGLSTAGSRDSYIFHARSADGRVADASAISPSPACVVVTVLSRQGNGTAPEDLLAIVRNYLNDADRSLVADRFTVQSAQVMEYHVDATLYLSPGSPNEPIRDAAQANLTAYVNHQHN
104	[S] Tail protein	vogp0417	STPSLLPPNSSHLERSAAQALAQIKNIPIPLRHLWNPYTCPFTLLPWLAWAFSVDRWDYNWPEHIKRQVIHDSYLVHCHKGTFCAFHRVLEPLGSLTNISEWWEPTHPRVPGTFNMAICVSENITEEMFLEMERLMDNAKPVSRHLTSLDVILAAHILACPGSVTLYSDIMDISPCKQPIN
104	[S] Similar to P2 tail sheath protein	vogp0418	AADQYHHGAQVLEINDGTRPIRTIPTAIIGLVCTASDADATAFPLNTPVLITNFQSAVAKAGTRGTLPPSLQAISDQANAVTVVVRVKEGMDYNAPTTSTIIGGTSDNGKYTGIKALLAAKAIIGVQPRILGVPGLDTQEVATALVAISQKSRALAYLSAWGCKTNAEAIAYRENFSPRQSMVIWPYFLAWDTVTNTTVTPPPTAHASGLRANVHQEVGWRKTLSNLGVKGVTGISPSVFWDLQAPTTHANLLND
104	[S] Similar to tail tube protein	vogp0419	ALPRELKNMNLFNDGZSYQGVVKSVTSPKLTRKTEDFRGGGMNGPFEVDLGLDDEGFNFENGFSDLAFLEQYGSVSANAVTLRFAGSFQQDNEIISVEMVMRGRRKEMDMGDGKPGEDAQSKITTICSYYKSTNNGKEVVZIDIFNMVEKVNGMDLLEAHRKAIGLL
104	[S] Putative tail protein	vogp0420	QNAPDNVIPLDQPIKRGAQSIEYLTLRKPTSGDLRGLHLLDLFQFEVAATIKILPRISQPTFTEQEATGMDPADLLACGQEVAGFLFQKRAKATNN
104	[R] Putative regulator of late gene expression	vogp0423	LAPHFRLTLDCQDITCYISHRLISLTLTDNRGFEADQLAITLDDTDGLLAIPPRGAVLTLRLGWQDSGLVYKGNFTVDETEHSGPPDILTIRARSADLRKGLKTRREGSWHNAKNGDIVSAIAWGNNLKPSVAPSLAGIPIPHMDQAKESDSNFLTRLAKDYDAIATVKAGNLLFSPASWGKTASGKALPHITLTRSDGNHHHFVLADRCSYNGVTAYWZDTDDAKHQKQKVSLKTKPKEKHLKTN
104	[U] Uncharacterized	vogp0601	MMLALGMFVFSLHTLAYQAFQRQTDCRRASYSRIGNQPARQFLGRGEDTITLPGVLYPELAGCALSLDALRQMADTGCAWPLVEGTGRIZGLWVIDRVSETRTLFFSDGTPRKIDFSLELKRIDDCCTDLLGSVLN
104	[S] Tail protein	vogp0411	TKVLTKQGDTLEAICCRHYGRTDVTZTVLEANPGLAEZGPILPPGSTVKLPDIPTSPPNPQTVNLWD
105	[S] Putative phage tail completion protein	vogp0413	TYZLKALEDRSAGLIASLSPSPRRHLAAZLAKNLRPSQQQRIMAQQAPDGTPYAPRKQQSVRCKKGRIKRNMFPKLRTSRFLQAKGSPZAATVEFTGNVPRMARVHQYGLKDRPNHDSHEVHYPARPLLGFTCYDVQTIQDVILAHLDNNN
105	[A] LysC	vogp0538	MNMKPFASGITSLCLMLWAGCTSAPPSPAPLIVISGCPRVSSCPMPGSDPKTNGDLSADVRR

Nd Int.	Fonction	VOG	Séquence ancestrale
			LEGALAHCASQVKTIKHCQDETDAQAHN
105	[S] Putative phage tail protein	vogp0412	MZKPDSLRGALMDAVPQLHNNPDMLRIFVDNGRIPSTSATSLSFEKAYTLKVIFTDFTGHL DSFFFPFLAWLRENQPDIMTTYAGHKKGFTFZADINDHSSLDISISLLLTERTLVKEVEGALH VIYIPEPPQNEPVTRPVELYVNGELVSKWDQPN
105	[A] LysB	vogp0052	MSKLMKVLVSSSSLAVSGLFLLDHENASSRASLDQANKVASEQQTTITMLKNQLPVSLTPA HNNELAQVALRQQLEKAAKGEAHREQTITRLLNENEAFRRWYGPPLPDAVRRLHQRPACT YASDCRQRLPQCEPLPHAGQGN
106	[U] Uncharacterized	vogp0768	LSTPSKTRNNTLDZLFQEAKKEEHEDPASAFSIHLEALAIHITKADMSGTEAAELLSQEATNF QNKSQELRN
106	[U] Uncharacterized	vogp0764	MTAEEKSVLSLFVIGVLIVVGKVLAGGEPITPRLFIGRMLLGGFVSMVAGVVLVQFPDLSLP AVCGIGSMLGIAGYQVVEIAIQRRFKGRGEQN
110	[S] Putative DNA circulation protein	vogp0068	LPYZPSZSSZTIRCYSTTWRDSSREASFRSVPFSVKEEQARAGRRFQIHEYPNRDKPCTKDLG KDTRCPTITAYLVSNNCSDQADPLMDALDTPGPGTLVHPHYGKMQVCVDCRVCISNSNTZ GRMACISFKFLZSGESSSPTSSAATAQISAHSCSFSDYFVYDLFSVFSMASVSNFIQNDVMDD VTTMLGNISDAIKMVDSAHTICLLQGYLSVMLTPPSSSINLLNTLQNTWSAGKCLVCYTSDS VTTIKN
110	[S] Tail sheath protein	vogp0074	TTISFNAIPSDTLVPLFYTEMDNSASNTPPPSRPSLLIGQSSNKASIALNSSVLMPSAYQASQI CGPGSQLACMVDAYCQTNPFGELWFIPLPKATGTAATVNFTFTGKARESGTLVIYIGRPRL QVPVTNGDNATALATSMEDAIKGLPTLPFTASSHAGFVTLTTHHKGLZCNKFPVSLNTVSF CGGEISPSGIQIAFKAGTAGTGTPDLTASIAAMGDZPFNYIGLPFNDTPSFNSMVTEMKDSSG NN
110	[S] Tail protein	vogp0168	SQFYPPTLPQLISMIHHNLLTRLHGSCPLARMDTEVYTNVHAAALHTFYGYIYYLARNMLP YTCDEDWLSHHSRIKGCPRKEATAASGSMPWDGFSGTPKLTASSRIQPDDLVHSTATTTVIS SGGFSHIPITSSAAGTAGNTDDSTSLTLVTPVTCLPSTGFTDTGNGGSDIEDSENWWSRFVER YYYTPQGGTHFDYVIWAKEVPGITRAWTFPHWKGTGTVGVTIATDNPTNPIPQESTINAVP HHISPH
110	[S] Tail protein	vogp0227	MNNTVTLRVNGRZWSGWTSVCISPCIECMARYFNVGITLQRPSDQGIIENGNKVKVLIGDQ LVITGWVEASPFHYNAHSISTGIAGRSKTADLIDCSAFPNQRNSHTLVQIARALAPPFGIKVR NDRAPSAAFPDVQPDHCETVFEALNRFSGQHRALAYYNPRGHLVLGSPGTKATTALVLGE NILSWDTEESIRERFSYYQVTGYHPGKDDDFGEATIASFHRTTKDSCVTRYRPMIIHQSCKA TTNN
110	[S] Tail protein	vogp0462	VAKDHYIHSLSHLLPPGPAWSNSDPFVEGSAPSLTRVHQRAHQLMRELDPCTTTZLIDHWE RLWGLPDYCSPDNTLRQRQQRSDAKVNLAGSINQHFYLDQSAALGCTDATINRFQNSNYW QVNMPATTNTPWTTCTDTCNSPLRIWGDTVLECFLNKLCPSHTFFIFIYPEGEENA
110	[S] Tail protein	vogp0472	METMLNVNGSSNCASASLDFLTRAVIISLFTWRRAZHDDNTPHPLGWWGDTWPAVQNDR NGSRLWLLQRSKSTNKTLQTARGYTQEALEWMIDDGVVSRIDIDTHRTGTDWLGLSITLCR RDGPIIIFIFDAICSNIHN
110	[S] Tail protein	vogp0461	KRSLNTMKRPLRNLIARAVINAINTARKCQTLDFNLMAGEQKDEVEHLEPYGFTSPPQPGA EAFISFPGGHRSQGIAVIVSDRRYRSKGLEPGEVAFYHHQGQSITLTHDGIFINSCSKPIVFPN APKARFKTPIEFTGHVKDQWDSGGKTMSQMHSTYNGHNHIENGKGTNINNPPSHPGATT
12	[U] Uncharacterized	vogp0348	MRWKVKDELVGIYPPRHYEHISVLEETQEFSHWFWDNHQDMEFISTKLGISTKKLNRILTL EQLPDEELLKDMVELCKIKN
12	[U] Uncharacterized	vogp0265	TKRYEYAGLTKELHQRLTLEFDALKEKHRRTLTKYIMETKQCDRSEARKYCQRFDNVVKE RSKLSPSTLDDMREYLTDGLVNDLQEYLSEHYSAPRGSCKPDTDKTNAGLTEELFLQYREE IQELRAAHPNCFADYIMEVKGCSNQQANTIRTAINTVYTEMGILTPRKVVQLEGLLSRELFG KIAKYVFNKYEWPESLDSEVDRIYLEYRTKGDLGLDKESVKRALYKAISMGL
12	[U] Uncharacterized	vogp0518	MFAKLFAINIVNNNYTFKRVPKVLKPKVKELIADMVNDEELLAKLTQE
12	[U] Uncharacterized	vogp0230	MRYKVIVYYDNMEDDVEIYENKDEAINRLHHLRGVKYRNSRLYKVEMVEVE
13	[U] Uncharacterized	vogp0195	MEIQYLEINQEHEPNENISDYIKDFSEAATVIDVQCNAIPVHFEKVGEDYWTDEDYGIKVVA FIKYEDNKEATPEKKQWLKEFFDKHLDKETKLFIDDPFDNRVTKGDEVYVDTLRNCLFQQ ASMAYTDKQIVSIYKEWLQNH
13	[U] Uncharacterized	vogp0182	MKKEIMTKAWEIAKEAVKKFGGKAIEYIAEALKMAWSDAKGSNTSLAKFQAVEEKMRKA GKYSMIQVLDFAKEVKFNEVMHKEGAYYGIEVVADGDSIGTYYIAEKVWEVA
2	[L] Putative endodeoxyribonuclease	vogp0188	MNEEFVNIAGYDGYZVTNKGIVNSMKTDQILHTFTIHHNGYLIVYTYHNNKTKTLSVHRLV AQAFIHNNDDZTMVNHMDGNHLNNHLENLEWCTHSENNYHSVNNNNGTHTRFFERITES VKHSSKVFTFISPNGNTEEFNSMAKFSKAYSLYNSPFCPHINTFTNCKRQKDPGYKFFNHHS YTEESYRVHN
2	[U] Uncharacterized	vogp0542	MYKALQDTTPCIQTAYNQSPNELVRALMAQGWSQEQIEDTTDIYQPSICRIYSGNHKDPRY SIMENLDHLVLNLDNFHLFTSN
2	[R] Repressor	vogp0486	MAVDLLRVKAERIARGYTQDZMAKRLGWSDRARYAKRENGIVSFDADELAKLAYVLGIS NGIFFTN
2	[U] Uncharacterized	vogp0543	MSNLRKYRESLNVSQTALAEEVGCTQGAIGHWESGRRLPDLKTCRPLVDCLNYLGAKVPL DDVFPPEHKAA
2	[U] Uncharacterized	vogp0484	MPTLFRKEYPRQSRSTEFLFLILFMILMTPISPLIFVWAVGKMVELVIEWYNYVVWASFNTL HNKINPYKEN
2	[U] Uncharacterized	vogp0536	KMMMLMDAMAKTARMVRIHDKPYRFSEFEMELIENVKNHGITPGMVSKRVKDGWELHE AMDAPZGMHLSEYREKKTIESLEQARLERKLERQRKKEAELRHKKPHLFNIPQKHSRDPY WLDITYNQMFKKWQEAKN
2	[U] Uncharacterized	vogp0482	LHKSFQQFIKSRYKDNSQTISVMEIKDFSWVIWQASRSAIEIELDKPKNDPZQGTYHMGYDE GHNCTINSCIKAIRAAGIKVKEN
2	[U] Uncharacterized	vogp0478	MEMIMACSTFNPLTLQKYQPDPEDLCSLCGGNHGKAAMIECKDKIHICLNCVDVLVDIKNE REDKKRSEAVRALDSWMRDGYSAAQIYDLAISKGEIPGVRIE
2	[U] Uncharacterized	vogp0468	MTKSWSVPFPESETEHDGMPVFWRFQATVEEDGIKIFALQYIAFHQTEHYAWLVPAHWIV NFKPAPNQWLQEWKQRRNRYAIKKVAKNAERSFAFPTKKLAIESLLRRKKYHLMRIKQDL AVVSTLVDGMKNIDTSTPDIEYNFGHNQETENWVFY
2	[T] RNA polymerase	vogp0452	VPHGPPYTAYGGHTQALAYSZSALEKEPPSGDMRRFZASZGRRVSASFDRGTAWTAHLLT YLAHPTAAATZSYEEEZEAKTGPCPTALALLRSVENEVATYSTISSVMDVSGTRTHSPTFQY VGRVVGRPIENQMHFSTLTYIAYKSFSKGKKPQKYCRYNSSCNPTNVPRGGKKVGAZRRA SCHCRTGWPWTQPCTRLQIGTTLSELQKTSVSSSTRPLSRHTMRTCGGRTLHHCRDSDSVR

Nd Int.	Fonction	VOG	Séquence ancestrale
			ZSLSPSKEPIAW
2	[L] DNA A-methylase	vogp0449	KAKGIIPYTGNKRSLAHQIIPHFPKHNCYVESFCGGSSFSLNIQGAILENINHLHFILVNLYMV MRHHPEEFIYQIMQALNSSQDYKWSHQTFPKTLNDTHNPFCLYFLHKRCFCNKIGFHNKGT FTTPFSSCNLNNIYEELCITFNHLNKAHIKNFDSVNCMELHNCHGTLIYLDPPYLITSLDYNK FFSLEEHKELSKFLDSIKGRGILSCNYHLLHHQGLANCHFZDFPNQYNINCCKKNERHELIRS NWEN
2	[S] Structural protein	vogp0485	QSLAEEEDNETSKEPSKTETYTYEEEQQZQYTDNDYMVKNLQKRISKEQAQKNETKTQLE QALDHIEELEKGCYKSVKEKSEEEKATKIQKAZEKQIASLKAQIKISNITYQANEVLKESGM TLSAAELGLLVNIDEEKTYSNVKTFLNLLDKHRYQWEKSIYTGPTPKIVPSNTKIHLZEQEA AZCZAE
2	[L] Helicase	vogp0491	IDIRYARFZRMNNDYTYZTAKAMFIQSEQITAKSSKFIKIFLMDHNDLLDZYGKAQINFWKA RNRAMKDDDILRDDLDSISFPHVINCGIPLRDHTTZIMRVGVZFRRIAAAREHFVVKYHYG VDFCFLFVYVNNIAYQNEVDAFNDLEFVCYWKRHVDRVARSFSSFYVGAREFRVAHDNT DVIFVGDISVVFNRLQMFDCAADLDHVKVSRTAAFLIRQLVDVZTNHIZRFKDZYSSVLHF DTRNITHNQTSV
2	[U] Uncharacterized	vogp0496	MPENTPVNWCDPSRDPGZGCTGVSQGCDNCDAEKSNFHRYSCTNDKWSPGTAHSKITSAN SHLQRHFALDYAKRFRVNSNSNRVFTASLRHNZDZQLIRHRRTGRFYVIFITNTLDSWLTTD HIANLADVLPWDRWYSFWPYNENIGGSRTTQGRAYRGKYLRVAGTARDZMSRTALEGSR LDPSDSLSDVQZDASTSNYDCKDGRNLDSFYPPDVDRPAPVLGLPWFVNWTYSSEWAYHS QSDGARAISKDSTK
2	[L] Replication protein	vogp0498	SHHKTLEGIWKATKICVGNASKKVVLVKLAYYPDDQGDCCPSYQHISHQLGLSKSSVSNLI SGLGDYSLVTKESRQEGKGNSSNFYRLNSDRSEESSGTSGNNSSPFGEDCQPSSWDTPGCSK GDYPSPSQSKNEYSPEFEQAWSAYPKRAGGNSKSAAFKAWKARIKEGIKPETMLEGIKRYA AWVZATGNTSTQFVKQAATFFGPDCHFEESWAVPAVPSPRLEDPPFKNSNN
2	[R] Regulatory protein CII	vogp0533	MEQTSYSKLSQREIDRAETDLLINLSTLTQRGLAKMIGCHESKISRTDWRFIASVLCAFGMA SDISPISRAFKYALDEITKKKSPAATEDFKQIDMQF
2	[U] Uncharacterized	vogp0516	MKTIIIEQWENEHYPLGRIKKQKLAEKTEHEIIFILNRMAQMPAIARFGEASEIS
2	[U] Uncharacterized	vogp0517	MKCEKCGNEIDCDCMGCHECHPEYTCETCGFCHIDGWEAGACWSLANDPDYDPFDI
2	[A] Amidase	vogp0522	TTNFSMYSGHCDKGTGAEDILTKIKEADKVLVTTSKYFKASGHNVKTFIFSTSTNLFSNLNK IVHWDNANPADFHISIHFNAGPRGTGFEVWYYASDZKNQKLAHDIFNKMAKDLGLKDRG AKATNLCFLNNTKVTAVLTEVCFIDNKEDANSFKKKNLLDKLAMANAEGLTGNSVSNKNP NRHSRTVEDSIPMFSKSPIKIYDASYSLLVYDHNKYWYNAYKKDKLCYVYKSFCISNGKKD NKGHLKVR
2	[L] Putative methyltransferase	vogp0523	SHDRKJEINPDJVGDFRYMPFEANSFHLVVFDPPHLRYAGENSWLAKKYGNLDERTWQDDI WQGFSECMRVLKPHGTLIFKWNEDQIKLSEVLDAIGFQPLFGDKRCKTHWLVFMKEDZTA NN
2	[U] Uncharacterized	vogp0524	MGVHVNTNNLMQDLKKMGFDIDKEAHVTFKAGAQTLAHZLKSNTPZSDSKKHTSDHITIS INLSYFNTCLNMVTIGYGKDVSRRTHFPESGTSKQDPQPFIKKTQKKGKNMVIETMLYTLK DSSK
2	[R] Putative translation initiation factor	vogp0532	MADYEEHLKELQKPFHPDRIFWRIQQSGYTQQGKAWAMVLAYIANRAIMERLDEVFGMA GWHNEYNVPNSGTLKCGISIHIGYEWVTKWDAAENNQVEAVKGGSSGSMKRAAVQWGI GRYLYNLPKCFAQTSLDKTNFKDGTDKTSFDKNKMQSVENYQAIN
2	[U] Uncharacterized	vogp0445	MTNHMYKTASIHFNITTHHINGRSTNIQFSTQDSGSSNLIFTFTKDNNPLPLSTVNTKTFVVF ZAGSNKYESFYNKYLTIIDNVEGTFEYFLSNEELKHYGPVQAQIYFYYTSDNNQALATSFFT FSIEKALVDHNIFPTADYYINDFENLEKIINHMVEDMYQTIEELQNQVTQLEAIMNKEGAQK KAHATQVKHKAYIKKHTNNKDKHIPATENKAWNTKENPSGTQNKVDSHSNYHEKHINSA DHTDWDSTE
2	[A] Portal protein	vogp0391	IFYMNLNRSLPTDVYNLWKNHZSFSSFNGSLFSKYLGISAFHNSDIFTAVSMISCYFSLIPLVF TDSATGQIIHSDHIDYSINTNPTHISYAYIFKLAMFVNTLLTGNSYTCIFRDHNTKNMNLKFL NPSQIQFKSSDHNHIFYFYTPFNSPCNNIQDNCIFQDIIHIKFFSYDSSIGFCPLYSLAHKIDFQK GGNNFFMDFFKGGSHASRILKIKEAFLSNKARHRTRQEFQKAQAGTNPGCPMVFDPTVDY SP
2	[U] Uncharacterized	vogp0493	NSDVPVTDFSNNNEENFRDLNYAYLRETWNZNQVWKLDFNVYHTNHNTNGYNLLPFRAS ILYHGHDFSIQHSSHNGSAHAWCYDIAATHFYYTCQKHNDCFFQPTINSHLTLEECLSQIFN DHTYPFSRDIINPSHFLDNVQQENVGGNHYLALFDNFVYNYGLNIFHINNHLCFGDPDDFGS KTDNFVSYEDNTDKSSININTLYFKTNLKGYSKDYCNDNHDFSTAKSNAPNADNCCIHZQV TSSDKCYTVLG
2	[R] Repressor	vogp0361	MKENKRMIEIDMGPVLKRIRYERGLTLQKLSRLTDDKVLPSNISRIESAGAGPTLKTLTTLA NALGTSPSDILRNAEGGDKVITKPQQVLYVPFLSWVQAGTWTESPEQPTDGDYKEWVEAP PSAPSRFFPFKFKGICMKAPIGKSLPGRMLHFFLTQPNKQAITAHSVLLDWQTZESTTFKZLII DGPHKYSKPLNPSYPPPQVNSEVHVSSVVLAWGEGYTVNGIT
2	[S] Putative capsid completion protein	vogp0394	TFIIPNTPFQDTIINNNPFWPNINSEHFQDZLGIHSSIDPARLEDAIITAITSVNSELSDWHLEKI AAGYTYLAQVNSHKVNDKNZMVLZYZCAVFAPATAELSQHYRSFYTTTEGNHKAQDIHT TMNDLWADPHRAIRNLLDKPRTTVZLI
2	[L] ninG protein	vogp0378	NMPIDZPNPKKCKTCSVQFIPTYSNQRVCSPKRAIELACKHAQEAHKNAEKNNZKERKAQE QKQKDQSKIAKLAEKARCHWFRLSQRAVNTFIRLRDKGQPCISCGTPWKTCNQACHFFTK GPSPQLRFHEHNIHSRCIVCNLRISGCTNHGYRPGLINHIGSEAVEGLEANQHPHKWTIEELK AIIAKYHAKLKDLENSNSKA
2	[L] DNA replication protein	vogp0377	TNTSDILNFSRPHLEVLEHNVADLDDGYTRMANTLMEAFSRADLTKRQFKVLLAILRKTYG CNKPMDCMTNAQFSDMTNMPLDHCYKAKSQLVHMNILIQEGGMFGPNNNISEWSIHENK GGSPLTSNIPQTZGKSPKTHKATQSKTGDCFPYTKNDNNDSIPKDKCKDHSSPKYCKSTHL DYEDFSVVHTNIASQSCSHSVTANDVNTAQLMIYMAKTTTPTJNNFRISVWACMAQSMPY RFGDNHLDMGVIFD
2	[S] Tail fiber protein	vogp0376	MGRCZCRGPSQVIIIITANQTSQEEETGKDWLDFSNILFPISYNCQAKLDDLTKKVTYQQLA TTTHLGSRAQVDHHRYEEACKPVQDHFIHSASNYNNVRQAQMAAQAPAHIHTTTRNQASP ANHHMASMGVIPASGSCQGVNASQDFTTTFDSAWATDTSHQKVPYKALALRAYFVNTGZ GSPSQAASHPGCSSGGCGSAZTPSQGTNSLQQQSHNWMWQGN

Nd Int.	Fonction	VOG	Séquence ancestrale
2	[S] Tail length tape measure protein	vogp0375	MTSKPLGNLVMDLTANATSFNNRMDRAKHPSQDRDKDVQKDASMVNQSLTRZDTAFQD ASISVSKSKAAMFISTTHYTNVSRMDHLASVSHAWAILLQQWGQANESFGSLIAKLDNLFE DITYPIGDSTSKAFATSPWAZARYEGNSRFTAFKNTLVSSGNQLGLTSHSMLDFTPTWSDTA ZTLDPTSTYLSDLVPALDTSDDHISPFSQAIGTFSTPSDDMVIKFASAFGDLANALNQGTTA MDNQFDTVSTSK
2	[S] Minor tail subunit	vogp0374	MPMINANPNSLADPKAFWNTIMAHRHKDEEERYKPPLIRLWDGDLGLHGLVASEHSPNFE WIZNDTGTAYIQLPLDHHLAKWVYNHKGRAENGKKRNVHITIDNRGARWSGRMDNYRIK HREDGDFFMVLDCSHNYEHTKHIYVWSNPFLPPEFQFPQSWLVSGPTNWILLLTLFINLIRZ HNPLLTLPNNPLDLTSWLDLAFNLSNWHIIVKPFPFLAANSPSVFSRFQTFHDTADNILEDTQ LTIN
2	[A] Portal protein	vogp0382	MTKQZTHQDANNSTHSNSLTDAFTFPEPMPTFSCTTIZDYLDCNHNDNCZEPPISLKGLAKF SNSNVYHGSFLKAQANMVACTFMPRRCLSRYDMNPFCLDYLGLGTASLVKIRSCLGKVIP LEPSPAVYMRNGKDGDFFQLVHNCKYNQDFKAEDVILIPQYDPKQQIYGLPDYLGSIQSSL LNQYATLFRRHYYYNGSHTGFILYTTDPNLTEEDEEALNENMTTSKGLGNFRSMFFNIPNG KAKGIKJIPLG
2	[R] Repressor	vogp0364	TIMTYDYSKLKGRJMEKYGSQHDFAKAIGLSENTISFKLNNKASWKQSEIDEAIELLGIPKE DIGNYFFNZKVQKIKQNZ
2	[S] Host-specificity protein	vogp0383	MMYFFIIDNSLHKIGITTDDRGGTINIIDNSDNHSIPSSTSSYTFTILVNPKESHGAWSZVNHS AAHGTYSSFNDKGNNALFNTIMETSGEDCACNVSFYCKDSGFDLINZTVSPYKATQATATT NYFSNHQCFTNYSSFQIGLNCSTWDIPNYNPTFECGGKGDTNLTRILSLEGHFDNADLDFTF DISSSTVVPFFINIYNHNGANGYIAINCDHDNNKLINSEYINDFNTSVDPTSSSTNANHHSANI NQGP
2	[L] Terminase subunit	vogp0358	GGSICAMNATVKIPIRDNRRQAKFLYWMGWPLCDIADHLGZKDKTLHSWKDRYGWDRTY SVZSIGGALEARLVQLILKDSKTGRDYKEMDLLHRQLERQARIQPYQGGSTETDSNPKSAK RNEGPSNKPKVNNISZELTQKLVEAFLDCCFDYQKDWYRANQHRTRIILKSRQIGATFYFSR EALMDALETGHNQIFLSASKAQAHIFKAYIQALAREALGVELKGNPIILPNGAELHFLGTNA N
2	[U] Uncharacterized	vogp0357	TTYWSKRNZAWQEDHIKKGKEQFKQELEELYKSQYHLHKELDAYRQKFANSQGLSISEAK NKADYLDVKAFETKAKKYVKDKDFSPQANQELQDYNLSMSVSRQELLMQELELELLSLSE DZDKYTYDYLTNGSZSEVERQSWLZRTFPPSKTZDYZVKSAFNAYFKGDGNSENIWQYQE QLKEIVKKZVTRSLICGNNCZTLAPQLNKYMDDYGKZADTZAIQQQSVFQTSALQSIMQEK CFKQFKLKPEN
2	[R] Anti-repressor	vogp0355	TSQLIQIFNCNITNZNAZSVNARDLHSFLGVCNRFSSWIQQRIZEYGFVENQDYIALSQKNNP GCGRPYKDFHLSLDTAKETSMVERNEKGNQARHYFIEREKESNYTQPTLSZPQQLFSHEDL ASZCYMLLWMDQDQESSKHLAPTTKESNNNYTCKSYQVNLQTIYLEPKDHPVSQPNEDRL YHHSLVTHKALPKLNSLLTKQFDF
2	[T] Single strand DNA binding protein	vogp0344	MDMCMMNKVILVGRLTKDPELHYTTTNSTTVANFTLAINRNYKDHTDNGQYEANFIHCII WGKPNEIIANYLNKGPQIGITGYIHTCSYEDQEGHQGYTTEVIVNNINFLEPNNNNQDCNNN HHHHYNTNHYSNYQRGSSFQNGNSYGQQGSFVEGNTNHSGSNTYYNHNHFSNNNNPIDIN EDDLPFFTN
2	[A] Bor protein precursor	vogp0548	MKKMLLATALALLITGCAQQTFTVQNKQTAVAPKETITHHFFVSGIGQKKTVDAAKICGG AENVVKTETQQTFVNGLLGFITLGIYTPLEARVYCSQ
2	[S] Tail sheath protein	vogp0908	MTLLSRGIDVKEINLYVVIIYLVTVRIVSINKFNSNVTYKMSKVIVKINLGHIVNDTKNSIVN SFFLNRNFVNYGNHLNLGYIFNKINVNNLCDLDSNLSYTIINSGINCEIDYIINIRSNNDIVNV DTKVIIINIYNDDRILIVDFHITNIVTKTDVINKYFFFNNIEEIEVSRYFFDFNQAMILNFDIIVF DVNQICIIFNIVZVSNTNSIFIYILKNVNFHSFVARYIVNVDNVVKVCIIIYIECYNFNNVA
2	[R] Repressor	vogp0365	MSSNMMSKGNSQZEDHVEKKAAEKEHNSFTDLARPIGMPSPGVSHLLNTKAMPNLEANZS LAKDLNTPIIWVZNSTNANNRHRIPVLGTTLTGANFDWNPKKGPAGNNZGCLNIPSQDHNS FGLRVQGDSMLPRIQHGEFILMEPNSKFIPGEEVLVRTNKGQDTVKTLFYHHDDNHHLLSIN HNHHPIILAFHKIDKIHYVTGITKHSTLZDNIKAN
2	[R] DNA-binding protein Roi and antirepressor	vogp0392	MTMNKMIYINKVTIASIEITKMVSSHHNNFZHSIKRLVKFNIIRQPPMEVSKKINNLGFFNVQ YKHYFFEGKQGKRDSIIKMAQFSPVFTARZVDRRRKLEEEVFKIPKAFPNNZAMATDTVZQ KKKMEKELEKDTPHAEKPKLIREDSITIIGNFTZVSKLGHIMRFKSMDNRHNLVFSCYIHNN TWDDHTDNCYRTHVKKTTVKAVFGIHFSFINKITGHSPHWYTPKVAYNGMLNFTNEAANE EVKSF
2	[S] Putative baseplate protein	vogp0415	DPEMHMDWNNKDVSRSLGLRSIKNSLLGIITTPKGSRPFDPEFGCHLSDQLFENMTPSTAYT VERNIQAAVRNZEPRIHKLSVYVIPLYDDYTLIVEIRFSVIDNPDHIZQIKLQLASSNRV
2	[L] Putative recombinase	vogp0410	MSNEDLKKAMSHKQGAIHQNPPTPLDNFKGYSEAMTLKIKDVSPQDMDANRLSRIALTFIH TNPNLLDCNPDSFMGAIMESAQLGLEPGSNLGQAYIIPYNDNAQFILGYQGLLELARHSGQ VSTIYARTVYNNDNFYYNYGLDHTLEHFPYQTSNDPGKSISSYTIAKLKDGGFHFKVMSNQ DIKKNRSTSKNCNSREDHFDHMAKKTVLNHLFKZLPISFKQLKGVITDEHTDSGLHQYNAK DLPSANPMDN
2	[U] Uncharacterized	vogp0409	MSTYZESMKELPYFDINVANILPDNZZGFCWNKFIFINPNLSDNEKRCILVEEVGHHKFTHG NITNZSKVNNIKQENZARRRGYKSLVPLQDMVEAYYZGFSNYYEVAKYLKVTKEFFLEAIK YYKKNYGISYNYSEYIINFEPSNVFKYKEIFN
2	[S] Scaffolding protein	vogp0408	LKVMERKDLEGLGLDDKQTTKVMNZYNKGIKAIKQRLADTKSELDSLKQQVTNRDEQIEY LGEQASNSEELNKQIAELQENNKNZESESKASLTKVETDYAVQMALRNAKAIMAFMDMD TVKLGDDGQLMGLDEQIKNLQESHDYLFDKGEDNGSNHTVKITPCGNPSSSGDGKTKLSD MTLAERGQLYRKDRQKWEELAKQAQ
2	[U] Uncharacterized	vogp0403	MSNCRFIIHNNFSNYFNTCLVNPPVMTISKHNVTLIQVSGVNSSFIKDNWDYNNITHQFNFF FLYPTIHFNRFIRNFNPWLTNNDCDNLVFPYYFDPFCNCNHVAFISHSSSSINTLIEDKGHFTIN LALDPFQYNKDGINYMNFTKTSPLYNPGNYZSZPLIKIYGNGNITLTINNIKFQFKDINDAIII DSEKWVVYKYMKDNSESNNILITRNFPILDPGKNNISWKGNISNLEIHPRWHYNII
2	[T] Similar to transcriptional regulator	vogp0402	MKVLLSIKPKYVYKIMKGNKRFEFRKTILKHKINNVYIYSTKPIGKIVGSFTIKQVIZDDPQTI WEMTPKESGLTKGDFYNYFRGMNKAFAIQINELIPYKTPLDLSZIDISDLKAPDFYTHYN
2	[S] Minor structural protein	vogp0379	MSTILYMSKTKNFNKSSDITTFIECCLLKZISTNGSDLKDITFSISYTZHESIKDANYWAIYDP KDLYFYFDFFLCTHDKDNSMCFZRINFNLNHLDFFINNVMRKVNNNLSCINNQSLSHAGFD WVSEFNKPSNIVTTTFSYSYMCNALKALQEFGAQFNYCFKSTSNKNPNSILZFHNHMGKINI THIQYGKNFQKIILQLDCASMVTNFFPYAHGENTCDRYGRNSEYSDKYRNKCNGYAZDNA

Nd Int.	Fonction	VOG	Séquence ancestrale
			QHLNWIED
2	[L] Endodeoxyribonuclease RUS	vogp0397	MKLEFSLSRNTTNNIKSHKAITNSNNRIHKNHKHNLAQAIRRITYKQICNPLNZEEVFHCFPS SNPCHVTFTIYTPNKSKNPPNLHPTFKAIMDGLTEAGIWNDDNHKVIKSSSFHYSGLSDEKD TYSLELDIKEN
2	[U] Uncharacterized	vogp0433	MQMSRTQLFEAQHHCYSAIILDDMPNLDQVKTVSGSSYQFZLDLNVADHSKACPRRGRWF FTZSNDLHPWTGQAPDCMSNLFHDZFILSSCCNQHNZLHNYSISDANLMVDLVLDCMFQC HIPFKEGFKEZHNEQEYYLFRCINHRFCIICCNPHADFHHFEVIGPGLNSNHVNHSNHHVLA LCRIHHNEKQDMGAKAFSDKYHFPVSCIKLDKQTNKRFSCKAKVPHSTDRVKGPLVKEVLI N
2	[S] Major head protein	vogp0390	MKMDKKLKELKEKSNDMNTKMHNLNDQINNKDKTKEQDKEWKKAMKDKEDMDKGIT HKVEVQEQVHAFKEZDKDYQGQNFDPKNDYDQDENHAHVFNKWMHHSTSNAZRZLQA QDDTQDNKGGYTIPQAFLDKVKEKMKSYNAFDNVTHIITTSTTCDIKSVTNIATIDTFVLLA ENNETRQEEAHPGTSLMASFNMKYNIITTVTHHLLKDSTINMTAYLAYRIAKKIVNTKDTH FLGGTGTCTPKHTMAZATN
2	[A] Portal protein	vogp0389	MEDPKHTGZLCSFTMMFFSRQFWHDSDHQDTSTHDSYSAFFSSYSSYSGKSIAPKTTLHLST IRSCIRLLAKSIANLPLNLYQNNGNSSEKVAIDHPLYHLMHYQPNRYMTPLEFWELMMARL CLZGNAYTYIDRNSHGQIIYLFPFNPHNIVLMMDSHSLPSNPYPNGQDDNLSQEEILHFKGF TLDGRLIGLSPIHYTRKAMGLAMATQEHTTHFFPNGTTPSGILHTNHEANLAQEAQDHFKE DLKEIYSGF
2	[S] Minor structural protein	vogp0388	KLTIIWDNINQAFHQPIQITQGNINSHTLZINIIHNSZDMDLTGHSFKLTYQYNNNYNSSFIML TPNNTSKGEFTLVIPKZMTKPGFIZSSLMSLNKDKNLVLSNNLNFISKDSSFTNLSREVKDNI YDFSKZLLYNMHQVTTSKZKYZQAQNKYDKSNMKWKSNLADSTSZQSAMTDLNNEVKA FRMSPENSLTIRSLSHPISSSSSZSZVLKLINSFKILNSNISLMSNGNYSPKANQTCWQSLQFT N
2	[S] Major capsid subunit precursor	vogp0387	EQELATFTTQLKDIGHQINYSVEEVKAQATHTGELNKETQAKVDELLTAQGELQARLLDA QQKLASREGNYAGEEAPKTSGHEVAESZEKQGMDSSSRNSLRIPVARNAITSVNATSSYGN LVIPNHLPGIVTPPQRRLTIRDLVAPGPTSSDSIEYVLETGFTNDAAPVSEGAQKPHSYLKFN LKNPPIHTIAHLFKASRQILDDAPTLQSYINNRAIYGLKLKEEGQILYGNGTGANLQGIIPQA PAYAPPLY
2	[L] Putative HNH endonuclease	vogp0385	GGMPMPIMKLCRHPSCHTLIIHPTAYCNKHNHZDTNYHHNCHKYCHHPSTSNPNQDYNNR IRYHNHNNPQYYKFYHSKEWQSIRNLVLQHDNYLCQYCWAHSDDKHTVANIVHHIIPIKE DWDKTLDMNNLQSLCNSCHNKKTNKDQNNYGTNZSNPZKNINSITNIKGLANLM
2	[A] Lysin	vogp0384	LIKKVKKSSLVVDVSSLQGSDFAGISDQGANSTTVNDTZGSDYSNPKFSAQVNRANHNGT MVCCYHYARFAANATVADQEAHNPVCFSLEVVPAQVNYLVSDYETGSSNNTSGDNQANT NAILRFMHIIAAAGYKPMYYSYKPFTQDNVNYQQILAQFPNSLWIAAYPSNDSVGTPNFNY FPSMDGIPIWQFSSNRFDKNIDSNVNLMDFKPNTSCTRKQDSKGPRTPPNTSSNDKIGGIWC YLDSKGYCSTCHW
2	[U] Uncharacterized	vogp0400	MQKNEYAIYKGENFIAMGTKHELAKRLGITPTTINCWSSPSHSKRNPDNRNIHFIWIN
2	[U] Uncharacterized	vogp0746	KIMYKVKAHFKDLQDNRHKYKVGDFYPRKGYEKERFESLSSSNNNQNKHVIKVSKDKNK VEELKELAKELGIKVSSNMKKAEIIKSLEAHDN
2	[S] Portal vertex protein of head	vogp0909	MKPIFLKFLKPYVKDDZNDFQFYSYDEIDSLTIPNTSESTVDMYHHFDATTPNSCILHPVLFY CFIPNIENTHKLIDYYYFFTKZHEIDNIIDEIVNNAVVKDDNDEIVYSDLDETNRZKCIEDIVZ DEFYVVCDSFZFHYEDZRDSYYWYIDSVIYFRKIINPVYLEQGVKELYVRDPIDVDDRRDII KNNNDGNEIYKSIEEFFLYNTAEYYYRNRFRKFNICNDRIRIHKKJIIYLYSGKVDFNDQGPY NHI
2	[U] Uncharacterized	vogp0305	MRYSDKVILKYDYTPYHYDPDLGHRVGREHCTKTSSCNITGICSDLQAELCDZLNNDSMV MRFMPSFNHDFDSIEYDSNKWDPITFRSPSEGNNFIZHEEVLN
2	[L] DNA polymerase	vogp0868	IKIFHYRVKYVKINIFKFYINNNNLKZIYKYYYKVILFNNRINVMEIKHYDIYCKPFICKKFFII GKIFQLVKYMKEINFEAFFIYHFIFFYILNIYKKQIRFIKNKLSVVKIYIYIVFLQKFHKLFEVK FFIKVLIYZNRLKKKCZFFKLIDNKKSHFRRZNFKSNNCILHDEIENHIIYVILMIEDKFLVDY VHFLEHNIFILFRNRKIYILDIFYIIKIINNIFVNKIVKLZLLFNNIKINYVKNNLNNRIN
2	[L] DexA exonuclease A	vogp0863	IIDFIIDLETLGNTPKATVINLAVIVFNPTFKELVPHCIKIKFNLKSQKGNRFFTKSTIEWWKN QSPENNLTPSNEDVSTMEGMDKFMDYINNHNVNHWKSQVWCRGNSFDFPILVDLIRHFHR PHSNNQRDIRTPIEAISLVRNMTMCPLPKGTLEGFITHDSIHDCAKDILMLIYSQRYALGLED IPTDQN
2	[U] Uncharacterized	vogp0851	MNAIDALGEAMEQALEECPANEVLAFLTGAFVGLIAELARRQGPDASQDIKIDGGKNRDITI HASKN
2	[U] Uncharacterized	vogp0818	MLRTTTATVFAAAALALGIPAVADAAPKHCDNHGTGHGMIYKHACATGPGGAGADWTP VMNPDGTYKTVTEDGKTRKVYKCVRHCGGGRHHAETTDPW
2	[U] Uncharacterized	vogp0810	MTVTSHLTALPSDHNVPSTTSPGZHHLSPDQFDRFLCPQTSANLSNFFASISEFLFRDLPEILA SIZDLFTGFQCFNSYSPAKJCAGVVQAFYDTFPMLVSHIPCLGHFFSYLHDRFFSPVFHASSF DSVLPTNKILFTVVSTFLNSLHCLIGTRPISVSNLZGNNSFADPTLPHVSVIVSSICVFDASFSN SAGYZCSAEVITFASVPAACDWEAVSTLNVVTZAQAFTAIDFTYHEVYICTTIPIDFPVIPFF
2	[U] Uncharacterized	vogp0803	YWWGRCDLNNDYVHYETIARIPRTTIVYAKWYDQPKVIQRAZWREGSRERFLHVSKGLH DLLSQADIVVGHHIDNADIPWLNGNLHLDAGLPPLPPFZTIDTLKVVRWEFNSCAPFKGSDS FCQVLGSSLPPNTDSYNGGAMZRAVKGKSLEDRERSVSYYTCDVVAAQGLYHYLRPHVK NNIHPGLFFDGKDNLLVCDRCGSKTIVIPCRNFYVADFZTYTMRRCTNCGCSHAEFRNSEQ CTVDSZN
2	[U] Uncharacterized	vogp0797	LAIATSTYSVNNMLNSSLRGVASTLPSSIYFSLHTADPGNTGANKVSTPTWPAYVPZQASPG GTISSSCAPPTNKTSTNSIEIVCLAQNGGPTITITHLTFRDTSTRGNFSVHIPLTTNYTLNPTDL SIIH
2	[U] Uncharacterized	vogp0790	MYRSILNSZQLSPLSSSTINSCQYTQKASLNNMYNPTSWQRSNZLEESRGQHCQLRIPGVCN CNPZTKILAPCHSARNWGTGMKPHNZISARACSDCHAWIDHNTLNSHNQDNFZYRZEGFK DSQAILSKEGEKDPCN
2	[U] Uncharacterized	vogp0784	MRIRSIKPEFWRSEDIANLPLSARLTFMGLWSYVDDNGVGEDNLVSIVAYLFPHQFARDPN DMEQLDSGGLVRRYKADHKGSLNDLLFITQWKHHQRVNHPCLGHHYPWPPATMVNTAA YLLSSSAYPHETLTPDPGNQGTCEAZQASPNGDEEVPWPPEPPPGPYASPPKVLDTZPLPME WVHKHSKPQPSSPYKTIVRQELGSTSYPHDTFDRLAVQVDKLNCEGRPDALIREAFQEPQR

Nd Int.	Fonction	VOG	Séquence ancestrale
			PPNC
2	[A] Rz1 protein precursor	vogp0779	MRELKTKLSVLMLPLVVSACGSTPPVQVKPPAPPAWIMQPAPDWQTPLNCIISSSEVGN
2	[A] Prohead core scaffold protein and protease	vogp0911	MNKPQLLIETWGQPGEIIDGVPMLESHDGKDSGLKPGLYIEGIFMQADVVNRDERSYPKRV VEKAVARYIKEQVATKRALGESNHPPRNNVDPMRAAVIIEDVWWKGDDVYGRARIIEGD HGPGDELAANIRAGWIPGVCSRGLGNEGFELTVGVDAVRGPSAPDARVAAEQISESESALEI TENSADEAFKASAESLKASNRLLSN
2	[U] Uncharacterized	vogp0772	MSIRIEIGDKZVITSNQYQFILHEKKVIKAGKKAGQEWLAIIGYYPKFNQLVSGLIHHHILTA PANSLPDMNEQIEELSMSCSEAFNSFSKN
2	[U] Uncharacterized	vogp0952	TITYLLDQVEVISEANIDLSHNLAQVKPKMGPHPPNRKKLTECEVKDIRQAYLGGMKQKDL AENYGVNPATISRTVRGIYHRN
2	[A] Amidase	vogp0712	SMARRNHDVSPIVITRAEASADPLEDAVDILGIVDKDAVWGIAVWRPASDNIAYGVTQQDF CYPHNFNCTTPQFKGFCDZFDVRGNGFAWHANPGSSGDFWSDIACMRDADNGDSDAYW YANGRHAHVTNVRHHRDAGPNHLYEHCADTAGGTFNFSSDPVWFEDVFPPPLGDHLGTW PGCAIGMGGILEDIGSVMRDHASNSDDNAESVDVVCPVLGRHFAPVHITPAGVVRSLAFGH GTQARLGSYPGFAAHD
2	[S] Major tail subunit	vogp0707	MASTSALHNKAQGTKVEVSNMASTKEDTAYSAFLDLSSTIKQIQWQGGQTEEIDVTTLASN QQEYIZGLPNPSKFSFQGNYQGNYLAQDISQAANGKKYRFQVTFPYSTSFTFMGTVRQYT WASSINGFISSTFSIRFSGSPTLVPPPSAHTNN
2	[U] Uncharacterized	vogp0691	TYKEKHIDHENGQFWVLKKYKSYPVLISGFTHSTSDSTYEESCLAKSRSDYLAHKSN
2	[L] Exonuclease	vogp0677	IIINTNDYIIDIQDLFTRIFVNYSLVFNDDHSLIYDVICVYVDKLIIGDLELFDYNSNERFLIDY DNVCIHEGVCIMLEDLYNHIANVDVHFYKKEFDNDYRMZZNFNKNNANVRDZVDZYYFW DMRRHIFFNNNDDIYQNDNYHDMTLIYRCNDNDRYLZINNFQIIFHVYLHAYVIDIIVDYKD VNWIIIFFRKRRFQIVITCYYNTNRLKERLVYVVSFNDWYNINKDMFYVSTIDNDEVZGYDL IZYRNGI
2	[L] Terminase small subunit	vogp0674	TAKKPKASNPSNYTQDVANDICYLLSYGEPLFQVCKHQRMPDWSTVFHWLDDHQDFPDN ZDKATQACSDSISEEALEIANNPLPGEEKNAQANLQVDTRZRALANRHPTKYGDKASKELA GNDGGPIQIKTSHMSTLFGNIN
2	[U] Uncharacterized	vogp0667	LSSEADKQZNPSPSPNSPAAVKAGAPTAAPQLEATAPKPPATQQEDYSQKNPTATSDETTA TPAHAPPAKQSGAAYTPKSKAENPAQSQAPKQPQDPAPRYWPPNFQDQWDKLPQZAQKE GTHPELEVQSKLCEANQPHNKPQHFEQVLSPCKRAMQGKNN
2	[U] Uncharacterized	vogp0654	IIDIFRFYCVYDDNVDZELNMVTNCILIILNIMSFAPDNVZYRYIRTVDQAYZISYPSNFCEVS VKEIITAKSIZRLFDSNVNIFINDVVDZYVASIFNNGSSSVKWYIIIPZGYYZKSYDIIIAZIFZP LTIHINFYMDSAIRIFRCILFLMDIHDGYMYLDYTLDZFISIIFKFYNINKRYRIZIYDFIKMYF KIFNNQRFSFVPKFLRSDFYYFIILVFYKSIDYPYIYFZYHNSDHLIITMILDIFYCNNV
2	[L] Reverse transcriptase	vogp0651	GKSQGCSFDQISTWENLLYASYYSSQGKTKAWGSLEFZENLDDZLAZHEQSKAGNFQNGP ZSHFZFYYPKPRLISAPQFKDRLVQHALCNIIDPLFEESZLPYSYACRNNKGSPAWLCLIQEE ZSNPCNAHSLKSDFSKFFPSIDHQALZSMLEDKJHCQATLDLLFFILPNYPCGIPLGYLTSQLF ANVYWRELDRZSQHDLGHNHWARYMDDFAILRHDQEELHQWFZDLQDFSPZYLGLKTY HWN
2	[U] Uncharacterized	vogp0648	EDZDQAFDQLIGHEGGYSNHPKDQGGHTKWGITEAKARDHGYQGPMPDLPRDSAKAIYQ AHYWTPLRWNQZPPELAFQLFDSAVNPGTGQAAKWLQASLSSSQHCRKRPSTLNAAAGM NPTTLDSLZDSSPKMCYSNLPTWPPFGKGWAIPFSGNLKZATEDLEN
2	[S] Major tail protein	vogp0637	MMAKVDNITTSNSEIDGAFYSAPLGTTLPTDANTTLDTAFKPLGYISEDGFTNKNYHKSNNI KAWGGDTVSTIQTEZEDTFSFTLFEALNLEVLKDVFGPDNVTATLTTGQHSNTINANYDEL PHHPFFVZVIVNNGIMKRLVIPKGKVTEIGEITZIHSESMGYEFTFTGFPDTEGNSHYDYTKD TNVTNKPENSFET
2	[S] Major head protein	vogp0632	TDTLSTKGTLFNPQLVTNLIHKVKGQSSMAHLSPQKPIPFNGQKDFTFTMDSHIDVVAENG KKTHGSVTLAPLTMVPIKVQYSPRFSQEFMZASEEEKIDILQAFDEGFSKKLARGMDLMAI HGINPRTGTASTIIGTNHLDSKVTQKVKATHNANPNSAMEDAVGLLTGFDSDITGVTLDPSF PSTLAKQKDSQGNKLYPELDWGTTPDSINGLTVDINNTVSYCTTATPNDTAIIGNFNTSFKW GYAKEVPLE
2	[U] Uncharacterized	vogp0631	TSEFKVINTQEELDTVIKERLARQRAKFKDYNQLKAQATELDNINQAZKSDLQEAKEHTDS SEKEMDNSKNQISGYKTATFQGILALQSGLPLDLADHLPGNDQDAWKAHTKPLASFIKPPP QQAPSKATEPNINNNTDSNNAYPDLVHGLYTEGRS
2	[U] Uncharacterized	vogp0776	MTIYITELITGSLVIAGLFIWGRGERG
2	[U] Uncharacterized	vogp1108	TNTIGIVARTTRLDQAQNLANTVRADYLSIDNGTLGCEDNHRKVVHRLAHHDTDWSVVL EDDAIPCNNFRDQLHPALAAAPSPIVSLYLGRQRPRHYQHRIEEAITTPNHWLTSPRLLHAV AIAIHTNLIPHMLNHLPPKNPIDQAITTWARHQGHTIAYTNPSLVDHSDSPPVITIPHPPRPRS EGRIAWHFGTRHRWAHTTLHLPYPV
2	[A] Polynucleotide 5-kinase and 3-phosphatase	vogp0600	MTMLZVMLTIGCSCSGKSTRASAIVNKAPGMYLNINRNNTCESLLAVNKWGEYKFTKDKE DLVSATQKAITTVAINTNKSSHNIIISDTNLAPQYRMEWEELAIEZGFQYCEQCFNLVSWDE LVAHNTHRGPQEVTLQHVKSRYQHFVENSZIPDYSVNIQLYEPTSSLPKAIMFDMDGTLAH MPVGVRCPFKWDKVKEDMPHHKVVKLVKVLHHCGYPIINFSGREGICKEDTTQCCQKDSI WRZPHNHIPCDL
2	[L] Terminase large subunit	vogp0604	IMCVGEKRRMMKCLKVDVTDGNDDLKESDNIRSILHGLSNADZNFVDATIVFDADVAVNH FDZHVEFLLSDDDRVLYGRRASGGNAVSVLLGTVQYVDYADYNSLILCRRYPELNZGGLI DMANNGVGRRHAERNDEKKTFAVPSGARVRFGRFEQEKDRYGYHGSRYRDITFYEFTKFF ESRYFTVFFALRZEAKDPFPLSVPAASRPGGISHDGVDAFRVTGDETVVHSSCCDNPDLNG DRDEDAFDIVDHVT
2	[S] Putative major head subunit precursor	vogp0607	MKVSHNFDQVKTQSPITNDSFLVHHPIISNVGVHAYZAPQVGCZEEGIZDVSRTPTELFNQD TLKSFEGTPLTISHLTDTPHNTNKEFVGCFVRNVNSFGNDLKAHLSIYDDHAINTVQAQEVG QLSCGYNCVTIMYPGQGSNQASDRIMGDDIGNDVASPHGWNNILANHVTTVIGSNASIPNP TLDAYNVMPSHKNKQPTKEDDQVKKAKAGZANPEGTKDATNHMDNMTAPMKEMEDKA DKVEAENKSSKE
2	[L] Recombination endonuclease subunit	vogp0611	TKILFLGDFRVRIRERNNZIQEILFDDLREIFDZSKNQSIDAFVZSGDWFDRHEVVAYDTIKF NQTWLNPIVNEYGVTGHSVVSDHNMHVKDRVFPRTSNZILSZLPNVNVFDZPTTFCFDGIC VYLIPWICKENTSDILDYIENAYSHDCVGHWEFDGFCFYZGAKSTCGZSSDRFKKYZEVZS

Nd Int.	Fonction	VOG	Séquence ancestrale
			GHVZTLCECGCVDYVGTPCTVAKGYNNDAYRFRLFDTETQKRRFIPNPNIDDKNFRYFDZ NRVNMNYSYN
2	[U] Uncharacterized	vogp0162	MCMIDLSINSPDIDVNDHRRDGNARGCKIVPFRVDGDEIKYRSDKWMAAKKTRALFAMYS NEGEFNKEDZDZVEAFFQNGYNSASVFVVFGRRGGGGGTCVHEELVYASAFWISPNFKKZ VNRDFDDVINGFKHNTIAGSVEEGYDDDILPZFSRIYLKDLDSYDLDHRNQLDSSDGKGLS VCRFAKAVFVALVDEATRHLQIFFDLSSAYVLRAFVAKEIQPFITTFAADYRERYFRFGVFD IPPFQPZFDTKY
2	[U] Uncharacterized	vogp0164	MTKFEEEVKRPDNGPITELNNNMHEVFSQYRKMRQYADYLEYELKQKDEHILKVENENSF MRDERTTFDSNGMAVEPKLQQQALPVVPNIKYNYFVAEWIEILKTKGLKPLKNPETYGET GFTEEKSQNIVFWISERQEDYMRAWLDGYTVEKPQLFYMRDELTGQFLAKNNQFENEDRY FFWNGLTHSIGTAWKLTFTQQEIDSMQTGSYELVPVEEGEEK
2	[U] Uncharacterized	vogp0166	TSTIPSSSNSITPSFLHCHZKQGCLCMASIKITTFSGKMPRLVPCLLPETSAISAKNFHLNFGG LTPNHKPYNQVTTITTNPKKTIYYHQDSWLAWPNVVNAIPCPIAQDNLYFTGDGSPKVTIGS ITYTLEVPHPTSSLTSAISGTSTGHICSRNRFGEESDPCPTSNKVNWQAGQTVTLSGLEATPG RPSIAVQCIYPSQTFQSSTGSYFMPKPSTSAGNFTHNISFGHQEEPLPSSEWNAPPYYYTGZIS LN
2	[U] Uncharacterized	vogp0169	TLADFTEIVAQSGVPIHQNHIDLVLEDNVTPLESFIADDSEMSPFCGWIAAIAIMPLECIIHHL VAGEVLPNVFFETANGGALEMRAWAAGFSLKASISANGVVIFTKDDNVDDITIEAGTVVQT HHIEDHISNLIVTNETVILQGGWSDVVPSIAAQAGPNFDLAAGYFCVLPQVISGVITVIDEAD CVTDAGANEEPNEEVDEGFRTGLAPVGERDIHSVYRDMMAEVATFSSHNIYFNNNNHDDP STANPYE
2	[R] Late gene activator	vogp0174	NLVTRFRCPWCQGSAHTRTSRCLSPEINERYHQCHNPNCGCTFITLQTFYRFIVHPGTPTNLP EAFHKIHPSPPHPFHTQQHQLCLZDN
2	[L] DNA replication protein	vogp0177	MKNIAAQMVNFDREQMRRIANNMPEQYDEKPQVQQVAQIINGVFSQLLATFPASLANRDQ NELNEIRRQWVLAFRENGITSMEQVNAGMRVARRQNRPFLPSPGQFVAWCREEASVIAGL PNVSELVDMVYEYCRKRGLYPDAESYPWKSNAHYWLVTNLYQNMRANALTDAELRRKA ADELTRMTARINRGETIPEPVKQLPVMGGRPLNRVQALAKIAEIKAKFGLKGASV
2	[U] Uncharacterized	vogp0181	VTHALIPFTYHNASFDGAGGFRIEIDVNSDFHNDFIGASSAFSIKKQTSVEAIGVRANDSYVQ VDDEVTYGDARZZGKRHTLNVDRDEVRVDQMSHAVSSCSNKSSGRTVDGVGRISQHRFE DNWAKFGGDSICIYFDGSHGLHEDVNEAAARADFAADAIEAPAAAHIECGRIATADASTNA HDRKSFGVVDIALVQTETVDAYRAEIADILPLIIDGEHYLWKSDRLHDDDLRDDDPGSGSD VQKRPSTA
2	[U] Uncharacterized	vogp0620	RKDGKAPKDQKPFYKRTWFWIVVVILVAVIGCALGCGCKGKSCSSSYNSYYSYSKAQNQK YSTNPKSSGQMNKDZVSNGTSANDKHKGKSSEFQGZVSZSTANFQGADANFEAGHFKDN RYQDTZDZAKTYQKEANQSTSGLNYHLQSKCGGKLTPEGWLZSLDYYYRVN
2	[A] Protease	vogp0343	MTLTRPVRPVDLTDHPDHRSHPMYRMEVHEDSDDNTLVLEGYASTFDHZDMYGGPANGN SLIEQIDPSALENTLRQNNNLHLFVDHTGTPLGCTKSSYFRGTLDLSVDNKGLKVIAHLDNS DLDDQSLDFNRNRSDMDDMSFAFRVKDQKWEATDDFLKDHQALRTITEVSLHKGDVSVD DLGANPTTAIDLPYIPPASCLSSQDNZDELGZVRSDEDLVKNTVEKZACKGSVHIKHIHVAY GEAGZAPZEPEES
2	[U] Uncharacterized	vogp1147	MINNSZQAKELAFFLSVSQSKACHIIRELNKELEDEGYIAIPGRIPVQZSHKKFPCN
2	[L] Putative endonuclease	vogp0951	MANILATIKDMPHNEWLAIRTKGLGGSDAANILGVSPYCTPLEVYLDKTGPFTHEVAESKQ NNWGTRMEPVITNEFSKDTSNQGZPPMEVMHLHFLYIHPHFDFMITNMDPVVLHNENRNG ILEIKTVSSNLAKEWGEDNAKEVPTHYMVQVQHHLEVTDKELSFIAALIGGNNFKPYYIKR DEEFIPHIIEREINFWDSIINGNPPQATSTSDTYTKFKKDLYPHSZNSDYKYLAEGFKFYQEK KKLN
2	[U] Uncharacterized	vogp1105	MVIPFRHRRRDPRRLDNLNRFTKHWGIYSCEVMVSHDCPSCYSQFNHSAAYNRGLTQTDP DILVFTESDSMVPYPQIHQAIKMASYSSGMVFPFSRFMAISNEDSIPVHNRNVEPLGSSATPI RNHHNSIGAINIFSCQTYKAMGSZDEEFEGSWYYEDTTKMAFHVTSGPARWVQGPAYHLY HLSGGCGQDRATTNPNHRRFDLYQQACAPQQIRKLTASK
2	[U] Uncharacterized	vogp1097	MARIHAKTDWHSRALCTQTDPELFFPEKGRTTSNAKKVCGGCHVRDECLAWALADPQNV GVWGGLSYHERRHLLEDHRAAANN
2	[U] Uncharacterized	vogp1095	VPQAMTPSFNPNDCLDMVSLLIFTFPSTIPALSTCCFRVLTLRRQPKGPEPTRQINPKIKEIHY QVSNTHNTNMRDDLDHIGDSVRECNKQIHQDIRRLHEDLHTERRERIEGDHRRDINCK
2	[U] Uncharacterized	vogp1064	MSTLAQLVHANLQEEWVRRYRYCYSSSLPSDELYKHSPKPHSYPRDRVLNRLFQLNNQFH NYTIFHSN
2	[U] Uncharacterized	vogp1050	MQRLSRVMVLSKQLTGCRYQNRRRRLTVANLQLAVKGEYFDAMIRGEKTEEYRLFNDYW NKRIMFREYDRLIITKGYPKRDDSSRRIDVPYDGYEIKTITHPHFGDKPVKVFAIKVNIGNE
2	[U] Uncharacterized	vogp1045	TMALZATLNTGLALLGYFYIMFCSGHWLSSZFLKQWNKRRKLZHNQKAIDALFEAFGLDR VZQGDPAKAISKGGLVILVCRPEENNNQDN
2	[U] Uncharacterized	vogp0993	MSPITGNHQEVHRWQQHRHYYNHSKELKTPRFHAHFYQPQATTYLEDLPHVQPDHSVCT AHDLFGSSSKRNQVRNNFYSLTLYQRPTWCMPAGLTHKESNLPYWHFNEEYCHWVHPWL EDLNKLSYLFECDPTILSQFKPQQYLH
2	[A] L-alanoyl-D-glutamate peptidase	vogp0982	MMSGKFRFSRRSEKNLEGVKPQLVAVVRRALELTEVDFGITEGLRTKERQKQLVAEGKSQ TMNSRHLTGDAVDVVAYIGSQVSWDWPLYEKIAQAFKQAAANAIEWGGDWKTLKDGPH FQLKWYDYPDHNPGKNN
2	[U] Uncharacterized	vogp0976	MTRRVLITGSRNWKDRTTVREALDTESHRSRYCRMVIVHGGNRRADDIADRWAWCTRQS GCHVFPDSHQADWEQHGKLAGILRNQEMVHLGADICLAFPLGRSIGTRHCMRQAKKAGIP VINFGYQN
2	[S] Capsid component	vogp0967	VTVRKLSEVTSGHPKQSSALGGSGLEGASCFACDTLRWNPSFYSPDAVVDPVKHITRARSR HIARNNSFTNSAVGYQHDSVVGTQYRLRCRPNINVMPGATEEWADEYQTVVEADLELYTZ ALACYIDRAAVSTFAGLVRLGVRSYVETSEVLATAKWDCPANRPYATCFZMVSTDGFANP HQRLDTPALRRGIQYNRHCPPZGYWIHVADAGDWYQMAPDMHEWEFIERSDAGGCLQVI HRLDPLEPDZSGELN
2	[U] Uncharacterized	vogp0963	MPQVCNIVNNDHPYFDVYIRRGTLWGNFNAGNDGTRSEVIELYRERLRNQFQDDEFKKQL MPLQGKILGCPCKPKACHGDVISDLVDHLSSPLNNFVGN
2	[L] UvsX RecA-like recombination protein	vogp0955	MSZLEDTSTSKMSSVSAYSKFFDDKDHTLTHIPTLNMALSGELDSGLTPGLTILAGPSKHFK SNSALVLVSAYLNKYPDAICIFFDTEFCFAPSYLESQGVDPDRFVHIPFKYNELKLEMVNQL EDIDHRNIIFIDSIGNLASKKEMEDAINEKSVSYMTRSKHLKSRLRIITPYSTIKDIPCITVNHT

Nd Int.	Fonction	VOG	Séquence ancestrale
			YDTQGEMYSKTVMSGSACIIYSSDTVVIIGNQQEKEGKEFMGYHFILNVEKSHTVKEKSKF KFN
2	[U] Uncharacterized	vogp0549	MITMGINMTTIPGFNNIYAATEGAEAHDHLSHAGKNASQAFISHLYHCSRNSQRIANKVDH HTARADNDTSNAHSCADTSQDASNYIHDFNDZDRRASTLAYCFDYSHGASLRAANSATYA YRQYQDRNRAASHSAGGVSNAGFRASPVSIAADGQAAAARAVGGFAGAHGZLVTGTSGC TAFAVGGGAVITAVKGASLVADDLTHAVHNSHAADLALDGAYVDLATIARPAATAIKDL ADVSTASSEAFCSADL
2	[U] Uncharacterized	vogp0623	MFGFKTKEEZSMLADHNNTMRDLEDTMDSMDHRYNSLHARAQMISTLQQZZEEAYSKZE QPHNKYIIRHKKDKFNQNLAMN
2	[A] Putative holin	vogp0307	MEEQAWREVLERLARIETKLDNYETVRDKAERALLIAQSNAKLIEKMEANNKWAWGFML TLAVTVIGYIITKILN
2	[U] Uncharacterized	vogp0127	TGFKEZSKAHSIDEFNEMQKNDQDLANIKDQSKNDDIKNALGTKKEIKQLHNDFKENEATI FTTFYFDPRNNLIISFGSFSTFWSDIDMTHVMTFRVFPFGNMKQKGRDKNLTMAGAIFAGG FGTMEGKNAGENTMDFVVHMTDGSNFKTKSMTINFTITSIADELKKFKELLDSGTITQEEL DIQKPKLLKTIN
2	[R] Regulatory protein CII	vogp0129	SMLFTTKREKANNINSTILNRMAILGQDKVAQALGINKSQISRWKHYFIPEFSMLLAVLEW GVEDKEMAQLAKKVATILTKEKAPKNSEFFEAKQMEWN
2	[L] Excisionase	vogp0130	MGKIMESHSLTLQEACNFLKISHPTSTNWIHTGILQATRKDPTKPKSPYLTTHQACIAALQSP LHTFQVSANDITQEZKCHSSTEVKFGTPVSHRRTDKDLCSLLAQRTKGHPQSYTTSKN
2	[R] NinB protein	vogp0131	ITSIKPSGHRHHQLLGILNNGTGEQAMKQTFLLRSQALQQNAINAILQIPTDNNKPLVVHIQE PKRSLDQNAKLRAMLADISRQVQWRGQWLDPKDWKDFFTSSRSKTNKLEDQAVPGLEGG FIILGQSTSQMSIACMAZLMEFISALGTEHGVHWSDHSRLDCEWN
2	[R] NinE protein	vogp0132	MRRQRRSITDIICENCKYLPTKRSRNKRKPIPK
2	[R] NinF protein	vogp0133	MLAISQGZHYQKESVZHALTCANCSQKLHILEVHFCSYCCSKLMADPNGRMZEEEGE
2	[R] NinH protein	vogp0134	MTFTVKTIPDMLVEAYGNQTEVARILNCSRGTVRKYIDDKEGKNHAIVNGVLMVHRGWC EGDDSLRKN
2	[A] Holin	vogp0135	TKKMPEZQDLLPSLRASNDQGIGAILAFVMAYLRGRYHGGAFKHTLLDASMCAILAWFIR YLLEFSGLSSNLSYIASVFIGYMGIDSIGCLIKRFTGKKAGFYEAN
2	[A] Endopeptidase Rz	vogp0136	MNLSPMSRVTAIISALVICIIVCLSWAVNHYRDNAITYKDQRDKNARELKLANAAITDMQK RQRDVAALDAKYTKELADAKAENDALRDDVAAGRRRLHIEEVSPSTAREATTASGNGNA ASPRLAAAAERDYFGLREGIITIQTQLDGSQEYIHAQCRK
2	[U] Uncharacterized	vogp0139	NQFQQTKMSKYNAKKVEYKGIVFDSKVECNYYLZLEYNKNVTKYDHFDIQAKFELMPKL DNQRKITYMADFSLCKEAYKMIEVIDVKGMATQVAKMKAKIFLHKYSNIQLNWISKAPKF NGK
2	[L] DNA methylase	vogp0140	MTIETHMTIFSLFSGFGSLDLAFQHATGHNKVVSHAKIHKASRQVLAYHCPDVPNLGNVHN ITRYDLRRASFNGHVNVLTGGFPCQDISTAGCNAGLAPGNRSSLWFEIVHTIHDFRPQRFFL ENVPSLLPIRANIRSTLGRMGGDRGIGTVLGNLADFGYDTDWAVLPASDVGAPHHCQRIFI VAHACDHHRVAAGNFLHRHDFSGASRTDRDAGVDAGRRPCASPHGTLCHCNPSSYQGAN SFCTDFDARRTV
2	[U] Uncharacterized	vogp0142	MINTNFNTSKKLYSLAHNDNFNVHKELSVKISGHTKRNHELSQLYLDICNKYNHSKQMKW GQLYKILEELTKDKQINLN
2	[U] Uncharacterized	vogp0152	LKRMSCYIPHKYZTLIGLVPDLSLIFSFKTCZIWLYTFLIFHNVDLNNZIZLQLYTNANFLFLII FFZTVZVLYHLIIKFFIWLMEVNNN
2	[U] Uncharacterized	vogp0160	ENLNLINDFGSMACDDYRPQZNENYQSSERFTSGFDDRVEYHQKAKTFHDFAKRIYYZSDI LTCCLZDFDNDLKYRFNNLVVAPNNNSAKEIDQARVNINGHTFDTLKDDLYHZYSATQISA EEIIKDFDTNKNEFZAVEYHLKPDEQGFVFITELAPLTNTVVQSFRVDRHAGIIYMTQAHNN NHYMLSRLDPNSQFIYCSLIMSSCHGTHNSYRYVKGELWIYSFILRCNNDDTLVRFQYDTD VKISHGEYRD
2	[S] Minor tail subunit	vogp0044	SFLMIDDAMIKLEGFNDEZFNLTDYNEGNHVLLSVNRVNDLZDPPVKVVYKGSNNYPSTY YLNRHVLNCDVVVNVCVFDNNDKDSDSWLALDSECDKRWSVNRDCKLYIATHNAGIDYL KLRLFESPEVHMDANPDSNEMHFAVVFCIIDDPFWYEYYDIFSVVTKKDTCFDANPWSREE SSEDTLNDMVDHNEDDDDSIIPYNFLEVDWPFSDYRMZNRLSYIPPIVZFAYNDSAFILFIIPG HSLEYKDQINH
2	[U] Uncharacterized	vogp0158	IZMFIYCNYYQCDGFSVVTLVRDSIDPFFSHTZDEFMVHDQFDDIQDGNLTADIRLKAZIHF VMDINLDSNVHCYANFRNWDDFPVVPNTLRPVAKTSSRDISFEKFEVLYIIHDFTSMDGTD DNAQIDNYFTVTPSYDRNGPDHCAADDSVTDGYMTZDPSSVIKVWZEVMPNRFZYGASSI TSNCIGTAEIFKSFDSGAWDGDGDGGRDGISYCFRGSLIVDDPHIDITRHFAEGTRPSPQDPE GAKEFN
2	[U] Uncharacterized	vogp0013	MTGPSDHLEGFFGDGPFQVGSGDWHYGQDFTQEAIRDLFNWPTITMLNAVDLLEEHLFKL PFEALKILNPLIPDLLEDDFANLTSSVSNIIDTLTDGPAALLFAKFHQWFDSPFSSLATVVRH VWEILASLSIPPFNPTNQAVEARRDQLTDAFGINTDTDLTNHIKDSFNKVLGLTSTRRTPND APKAMWTIFKKVRTSTRQWQDELAAETPEQLSRCTFIVPLTHYPNAPLPDVFDLTYSGTGT GYRDDVDIF
2	[L] Endonuclease VII	vogp0017	MSTTTKCCPCYHMAAAKPQPRRCVYCATTCITANCTPPHPRAHHPEKRTVRKDTAWEKRL WESYDVTANQYWQIYEAQGCRCYICHKGRSLVNKLAVDHYHHTCHLQGPNNTPCNFNLL GSNGQNPQALQRGINYSENPPTFAFISKRITPIZVPNLNRNQGHKKGYA
2	[A] Putative portal protein	vogp0018	TTPQTNTPANTLDSNKTLRLHQRLLRFYCRELDCQHDNRHETAMDZDYYDNIQWPQEDFQ VLKDRGQAPTVYNFISPTVNWVMGTEKHTRTDFEVLPHHKEDDKATWPKAILSKYSADFN QTNFHNSRNFAEPTKAGLGWMEVRYQGDHZGQPFZACSESWHNVLWDSTSRQSDLDDCR YMFRSKWVDTDMAWAIFPKRTQLIDYSIDNWDTFVSSDIYGDQATDLPZSGKDYQN
2	[S] Major structural protein	vogp0023	LTRYSFNRNDIPVGRLYVADVNKVYRGMKNSADTIKDYLDALHTQLTYSGNNIRNASDNT ISGDETFTSDVKVDGCFNITSNFRSRQLASSIFLSMSNQGTTIKYNIGGDSIIRNDSFYDLRSS KTSIEISTSILADTDLVDFCFANNIRAGNPCSVVANIVGSSDAFHIIIDCSGFDFRHWFSPTTNS STDCGTCTSRIHLDNLPNDKNNDLDIIPKTDDDFFIDNKTYAGDTEFNDSSNSTSMSPSDN
2	[A] Lysin	vogp0154	MNFPQSLVNWLLDHHDLLTYSMYCSIDRSTSDCSCSMSQALKEAGIPIQGSPSTITSGQRLA KNSFYHISRNENHIVLMSWGANTASSGGSGGHGIMMDSVNFIRCYYSTQGTAGQAINTYP CNRNYQANKPAYIKVWHYSQSAPQTKNRANTTVTPQQKAYYZANEVKYVNSIRQIKRNY LSPICFDHLENCIPSSAVHWVYEDGQDLPYGPDQDLEAGMHLSFSSDETNILDTGNGGYZR

Nd Int.	Fonction	VOG	Séquence ancestrale
			GYYY
2	[A] Putative portal protein	vogp0033	TNTLTTAFZVHKESMDHQQLGNRFLHHFKNHIZDHQCFTSYYKDNHGHHAIRVSLPRQAH NLEARLSLPCZNVNALAHGHILEGFHPARNGFDKATYPSWDMWQPNNLVIEDDLDHGVPL VHINSYNTITTTYPNTDFCIDPPAPLNRIHHTTATYADRYPHTCTHYHSMDSVDHKDVNDIL SAALYFPTNTSTZPLGEVTWVZATHGHINLQIVTVLAFPHFTGSGDFCSSSHISPKVLYLND GATHNLMPIVS
2	[L] Terminase small subunit	vogp0157	MMASRPKKLLSNYNKNYTKEEIIEKEGQEAQLNKFSKIDTEPPHYLDEIAKQEYZRILPHMQ ELPISNLDKANILHVTYLHQLAQYCSFYSNFVKASLILDREDLIFEDDKGNQMVNPSFNIKE KAGIHLHQTANTLGLTIDSRLRIMVPKEKEDDDPYMEZVCDNNHN
2	[U] Uncharacterized	vogp0045	HKIPTQENRNPNKPQEAFAWALCDLPSISGGRSVTHPGFLQNWSKHLWEWGFPNSPEQHIK FQAPSRSPPSHYNPPAHWVPKDAPDPKPINDLHNLTQQEHQAQSDLYQQLGLIHDDPPRHH N
2	[U] Uncharacterized	vogp0048	MAGISSCLEGIRNSSAISWNTVPILKVSLGNDQVWPAFDPVLTPLTAVGNYTYNIPAQAEFI DVILLGSGGGGQGTGSATAWGQGGLGCSWVTATLCLGIDIPWAVTQITGVIGTGGTAGPG YIFGQTSAGGKGSYTTPTFSCCSTLIASGGAGGNSRKLDFGGKSPNPADMVYRDRTZEGGA HQSTPSGISYAPGGGGASATVPFGITGLANGARGQAWFLAYS
2	[L] Terminase	vogp0053	MTAASTVDAHPPHFTKDELDHCWNDPIWHIKYCNZFQSLINCYYLPDKYGNYLKLNPNHR VHQLFIHMKHHHMILTSCHLGYTRSSYIILSDHKLFNANTQWAIITQDLEAAZKICRDEIDFS YDNZPPWIHIDMGVVNYHSRNNEEYMLDISNSYRLASSTFHCGTINSTVVAMFGTICVNFP TEDEEIVTASZGAVSKGCMDRIKSTTEGWPDFLYPSIVQSNDIDQAGKHFSAQDYYFPFSPW GRGTZYSMD
2	[A] Shiga toxin 2 subunit A	vogp0342	MKCILFKWFSCLLLGFSSVSYSQEFTLDFSTHQSYVSSLNSIRSAISTPLEHISQGATSFSVIDH APPGSYFSVDVRGLDLZQGRFDHLRLIVERNNLYVAGFVNTTTNAFYRFSDFSHVSFPGVT AVSLTADSSYTTLQRVAALYRSGMQISRHSLVSSYLALMDLSGNSLTRYASRAVLRFVTVT AEALRFRQIQREFRTALSETSPRFAMTPEDVDLTLNWGRFSYVLPEYRGQDCVRVGRISFD YISAILG
2	[L] DNA polymerase	vogp0054	AIKNHVNDLCDFAFDDILSIRIYTNEZAFNSFTHFTCDRYNATGFDTTSLSNDFHTNYSTIDR MQPHSSYSDRVFSNHCNRYFNIDSASHIRHYSIHIIFNNTLFDIFAINYSITMLDKYVYPNIVN TVVFARLITSRDSQPYIITRTLDTLAVHRMNFPKVKCDADHITMLTKYZNPSETHVCDYTHI ZYPIYDPYGTVDIIFTCDLHSVIDTLNIQYARNHLSRAYKGRSADDGSYILYNRHMLRFVEL EDFT
2	[R] Ren protein	vogp0055	MTGKEAIIHYLGTHNSFCAPDVAALTGATVTSINQAAAKMARAGLLVIEGKVWRTVYYRF ATREEREGKVSTNLIFKECRQSAAMKRVLAVYGVKR
2	[S] Putative capsid protein	vogp0058	TKQFINFPHLAYQVFSAPQSSNPHIIHSIITALDPHLMGEVSAASISDMAPQNPQPHSSSVAN DGMAIIPVSGILVPRTSHINPPCTPNTSYEGIRSHFNQALTDPSVZDIFLGNDSSGGTASGCLE LAHYIFASPCVKPPNALVNYSTFSAGYFMASSSQVIVSHTSGVGSIGVILDHLDFSKSZEQK GFKVTSIYPGDHKNDCSPHQPLSDEASAYLQSLIDNSYKTFTNSVAKYRGLSTQAVKDTQA SIFF
2	[U] Uncharacterized	vogp0071	MKKMSITTIFAASALLGLSACSZADKVSCNLSKDSDNFKVHRRVVFINTITDKIMFEIQGFISI NTDTDSKKLDIICKISKDQYKKHIMGLSNNISYFMEDIDGANVSTYKYQVNCKPQSIVPVK MMSKN
2	[A] Shiga toxin 2 subunit B	vogp0125	MNKMFTAFLFSLASFNSAAPDCAKGKIEFSKYNEDDTFTVKVGGKEYWTNRWNLQPLLQS AQLTGMTVTIKSNTCHSGSGFSEVKFNN
2	[L] Putative resolvase	vogp0086	KDKZMEIEIKLVMLPTAPQQDSHHSFHNCGNYDNTNSTANYNQNSKSMKNWHKDGLYLD NPSKLDVAFYFSAKEFFSNNPNIFKTRSNTKKSKRZSPRNVNKPDLDNLIKAILDSLTDSCNT DDSQIIKLNSHKLYAKKTQIQVNIIEIGFP
2	[R] Antitermination protein Q	vogp0084	MNLESLPKFHSPKSPKMSDSPPATASDCLSSTDVMAALGMAQSQAPLGLASFWGKLELHN HQKAIKCLMQFAQELSSNSLTKLEGDTKSQVLQFLATFAYADYCRSAASQGNNCRDCQGK GIALDMNNTQLRGTLVYKECGRCKGIGYSRLPTSSAYHPVHKLIPNLTQPHWSKAFKPLYD NLVIQCRQEESFAEDLLNKVTR
2	[S] Tail fiber protein	vogp0081	LFTAVGASDLAAASLTCDKWLTVTVLIFEDGGGNAVPIPDAGQTKLIHQDWCNALNKIVV YNHNTNFMVAELVVPTVVGGFWMCKMGLZDDNGTFVAFTNRATPYKPYVDYGSSHTQT FRMLIIFSGMASVKLTIDASTVKATQDYVNAQMAEDRKGRDHPNKALAGKGLVWRGTITH AHTITLATADVAAGASYSLTDSAZAIIHTASNPKTVVNYGTATDSDSYALVATADADKMSF KQATNQSAANKDN
2	[U] Putative C4-type zinc finger protein	vogp0080	MADIMDZAPEIEQEHREVSLSNHSSNSQALSPSHCWECGDPIPKDRRQAVQGCHYCISCQE VLELMSKHYSCKN
2	[U] Uncharacterized	vogp0092	MPTTFFSNYINELFHNKGFKNKDLKLPKLLYSLDLCFVFHTGTHQFZEELGNPVAPNGYPN YZYKGAFPTWTYCPIMKYLIYNKNPFDKFSDAEITFZGNQKIKITKFMNNVTDMLEAISTLE LVDRSRNNTAWKDGPDYSSSCPMAKEFTTKEPKELFKNNDGSN
2	[U] Uncharacterized	vogp0078	MHKASPVELRTSIEMAHSLAQIGVRFVPIPVETDEEFHTLAASLSQKLEMMVAKAEADERD QV
2	[U] Uncharacterized	vogp0093	MKDPAYGFSMNKVIMLDZNTIHALKLFVLRQMTPSZYLQFZFTAGQRYFTDTYSILNLVST PVVAYRSFLIRRLFKQLEKDMVYRKLDHQKLNTMNKHFNKANIYINTLTNKLVNTKMHA KVDKCSTSNYNIZNNNIDKSHNKSSNTTDSSIPYYEIMDYLNKKAGKSFKHNAKKTKDLIK ARWNHGFKLEDFKKVMHNKVTEWLSTESNGNLRPETLFGNKLQGYLNQDVHTKCNEQR KDNNYKDSSRY
2	[L] dUTPase	vogp0070	ITRGFKKLSENANIPERNTEHSAGYDISASETVTIQPQEIEMVSTGLAIQLREGZVSZLYHRSS IPIKSRIALINSMGVIDSHYYPNQEFKGLFMNISKEDNNIYKGHRIIQLVFIKYMTTIDLNEFD NATDKGTSGFGSSGVEWE
2	[R] Thioredoxin	vogp0065	RPHICTLFMLIIISTRPNCNPCKLIKNNLTQAGIQFKAVHFTQDPEASNYFNDMGYTITLLVL VTCTNNPSLGFQPNQLDELMKDFTTSN
2	[U] Uncharacterized	vogp0064	TISILIPEDKILVNKVZYQELKDKDLDSFVGMEDLIKKSNRSITNIIKVWLNQLSIENSGPVYY PNGNGNNWSFNSKEMSDFINKDFYQMYKGSSNW
2	[U] Uncharacterized	vogp0063	MSKSILEEAQDLINGQRNKNYGHPKENFKDIATLFCAYLEGMQITPVDVAVLMILLKITRFK GNGYHQDSFTDMAGNAGFLENIYEEPVEHYPDDPCQWDTLSDIPADIKTVTSATDTLWIHY PNDIHHSQGRRDGFISDNTSISEGPMLDVFGEVAEED

Nd Int.	Fonction	VOG	Séquence ancestrale
2	[U] Uncharacterized	vogp0062	MSERMVIISNTQIPFHDRKQLKAFVGFIGHTQPNQVVHIGDLMDYPSPSCWTKGTKEELAQ RIZNDSKQGKSRFLDPLRQVYDGPVCFHKGNHHPHPLYYLRQYAPALVEYNSSFNFQNLL DFDGFGLDFLPEFYKMAPGWISAHGHPGNFTQNSSYTPYNATKKFSTSIIMGHTHHRGIKPH TLGYGGGNQKVFRCMEVGNLMNMKZAHYLKGGTDNWQSGFCLLTLEGQHVEPKLVPIF GGRFSN
2	[T] Putative ribonucleotide reductase	vogp0059	MNNYQDVSTTAEVVRYRTYNRPLNDEVLETWHQTVNRIIYHQRWLWECHZKQSAELTEL LQLMLDRKATTSGHALWLGGTNVTZKNNASQCNCSFGNFASLHDFIHAQHLLLQGCGISF YTSISILYRFTTPVZVLVIGSKKQKSNPQGTKKNVHAFYHZDGEWVWEVFIGDSAZGRANA LGKLVDINZRLNIIHFNYIYSYFCQIRPTGIHLKGYGCISSSHSHLSVAQTAIRGILNNRTGQL LHHIDILDMID
2	[R] host-nuclease inhibitor protein Gam	vogp0079	MDINTETEIKQKHSLTPFPVFLISPAFRGRYFHSYFRSSAMNAYYIQDRLEAQSWTRHYQQI AREEKEAELADDMGKGLPQHLFESLCIDHLQRHGASKKAITRAFDDDVEFQERMAEHIRY MVETIAHHQVDIDSEV
2	[A] Putative fiber assembly protein	vogp0116	TSMAINLTNQPCNIMIYNFLVKDQDHFFKHNNFLCFSTDMPEDCSKATPNFLPADFIAIYNH NDATWPFAKDCYSNTVYYISSPEALSVFNFRPIPKNFTCFNPGSWYFKRNRVNRPTHTKDQ QLFQAKDKKATLMALMQEASDVISPLQDAIDLGNATKEETHZLEAWKHYSVLLNRINTSN PPDIDWPKKPEVDKYE
2	[S] Putative minor structural protein	vogp0020	VKFWSNNYRCYHIHLWVDQVSQDIANNTSQVRFKSALLNTTTTFTQYSCSAYIDLKGRRL DRSGSPSILSCNQTIPLIDRTVTINHNANGTKTFGFCASFNGSGGWSPSTSTIGNRTFTSTTIPR SSSISVSSGNNGSPITINJNRQSSSFTHNLCYQWGNFKRGNIANNVNTSFTWTIPMDLAYDIP NSTSGSGAIYVDTYNNTTLIGTQPNPNFFTVPNNSMKPTLSCISLTDTNTVTRNIISSTNYFIQ N
2	[L] RP thymidylate synthase like protein	vogp0057	MNAKTTAHISHKHPHLSDIFNPSFSLCAESVSVYCCHSSNSDNTLAPHNHSLINYLIEHGHW SPLEHAYTTFQIKAPSRSISPQLFHHRCLSFHEVSQRYVDPDPGDFNLPHSRRQATENNHNN NDNLASDMVADFHYDIQDFYDSINFLYQRILAYSIAHEGARTFLPZNTPTKLYMSGTFRSWI HFCQLRQHKDTQDEIRLMADQFZNZLHDISPNSYQAIFKSPYRSHGK
2	[A] Host-killing protein Kil	vogp0122	ISQESQRMDQTLMAIQTKFTIATFIGDEKMFREAVDAYKKWILMLKLRSSN
2	[L] Exonuclease	vogp0121	SLNFKALEQGYQDWQESRLGIITSSEVHNFMAKPKSREKWPDRNLSYLHTLLAEVCTGVDP EVCNKSSAWGQQZEPDARTLFWLTSGIKLTQSHLVCINZSLGTACSPDGLGCDSSCLELKS HFPYKZFSKFISGSNHSDHTAQVQCCLWVSHKECRYFSGYCPCMHREZVLVQRDENYMAY LYEMVQNFFEKMEEALEKIWCVFGKQ
2	[U] Uncharacterized	vogp0120	MAMKHPHDNIRVGAITFVYSVTKRGWVFPGLSVIRNPLKAQRLAEEINNKRGAVCTKHLL LS
2	[L] Essential recombination protein	vogp0089	MTKMLZAKLTHKVKLFKAPKNKYNYFCKYNYTTLKDIMDAIKHAFKNLDFSISKDVLINF FNIDNNYINVETIIIHNYNSSFTFNTNRVDVNKNKVNVEDAQVTSSSFSYTHHYSLSGIFGITS DKDNDTKDHKPLKDPKNFKKPKSZNYSIYSNRIKDVSYNZVHNDKDNIKDSNDZTKKIIID QINHINHHWKLKKKQNHSNNKKTKEMKTSINZKKQVAKEDTYKDHEIESN
2	[U] Uncharacterized	vogp0118	MTNITZQGLQEKAEKANKGFYTLAHTZDDHHANISWVLICLZGCLDNHRILLANNHSSSC MALDQQPHATAEFIAITNFTTNLAILDNQERNHSYDKRDNQENQANALALGYLGVKSDAP DTNKPISESHFSDLAESERVKHSLLEASDKRNTKTGYNNAIASDPYLELVLWNKTNVILFSD STLEALDTISHIFGLAWIHDTMLIPTKSSDESEKDAQAYFNHKNAFPZYZLDKFHHWIWVQT QAPQAASIRN
2	[R] Regulatory protein CIII	vogp0123	MIYAIAGGARTGASRSNESLLERITRKLRDGWKRLVDILNQPGVPCNGSSTYN
2	[S] Outer membrane protein Lom precursor	vogp0113	KSIASLVVCAFSGLACINSSAHAEEGQRTFSLGYSQFQPPSLQYLFRDATALYNPQGININYR YEITDYFGVMASLAWTPSZNNNTNIKSQATSPYKYLRANYWSLLAGPSWQIYQQVSSYAM AGMGLSKWSAYSKIZDNINSSPGFSKEYSTNHTSLAWAAGTQFNPNDSVTLDIAYEGSGSG DWRTSGFTAGIGYKF
2	[S] Tail fiber	vogp0112	IDDNIVATNGYDISNDGSISIEYQEIFDGMTRNNVZDLISEEHVIGLNRDWHHLYWKSSSFLN PDNFFNLEIFAIDFSISAZNZNDVWCFGVFLLKVNFSIHLTFDPTQVCHTNEDZLTTINFRFTL PSZDDRNTLDPNLSSYYFICFINLSVDSSSRTRVLANFVQGKITFESIDFEIVNQADKSHIFLIR INEITDDSNSAVTDNTIKVTKVIDIVNDQMCYAMTAVMRFVFNFDSFTCYVFRYFNNHFINS F
2	[L] DNA N-6-adenine-methyltransferase	vogp0106	LYTHSCQDLSMLMNNKSNTPAQDKDCWQTPSWLFHGLDIZFGNFCLDSSASDNNTLCPHC FTQEDNSLNSDWINHRPJWKNPPYSNPPNCVKKAAQQRVQQGQTFVMLVPADTSEGWFSK ALDSVNZVLCSKAYZTQDWYSIAFITDCPINFINPYTGKETKGNSKGSMLSILRPFISPRRKFT TFSKAPLMANREGLHPATSQQHEQN
2	[L] Holiday-junction resolvase	vogp0105	MNTYRFILPCPPSFNRYWRHSSGHHYISDWGQQYRREVREIIHQRCPQLDMNITPLIRITIFP APPDNRNRDLDNFPKALLDALTHAGFWZDDCQIDYMRIKRCQAIEGGTLGFAITKIZAISPII TZSLN
2	[T] Ribonucleotide reductase	vogp0102	MIKTIINSDSTLEASNPEKSNRZVDZASNDNNNKPINWSYVVMDSMNDVFYSDSTQKLHLK MINDCFDKKTZDYYYMASYLSVYNFDKEFYSRFNYIPTMYTFYDDIDKDIYDLKVKLDRS QEDIDYIEKYINHDEDFYYAYSIFKQFKGKYCFNDLETSHVDQTHHVVYMFMAMNLFDEE PNDDYRKHVVNLCKSLSNDDITSPTPNLHSLGATVYSISSSCLINAGDTSDSFDIASHFTDSTI SQWASVGIYM
2	[R] CI repressor	vogp0100	MSYDMGIHWZFLASKDVLYRIMEAYGFTKQIQLADHLGIPHSSFSTWDNRGSISYELVVRC PLETGAHLEWLANGEEDZNNYQPPVCNQDLYDHLKSNDTQPIKGFTLKSYILKNNQZLSA YLHIFTITISNWNHZTYGTAFRIEEDNPTYFVINDYGTLVHGZCLVNIEGSHSIRHITILPGCK LHIAGGKFSFECWLYDIEVLGQVIFILRKVNGK
2	[U] Uncharacterized	vogp0119	MHFSGSRLHILCAYACRHGTCSMTPQQENALRSIARQANSEIKKARQQFPDN
2	[S] Major structural protein	vogp0282	KNPYQSLFDELFKRFQELGYTTYDYLPTNEVGYPFVVMGNTMTIHKSTKTHIKGTFSLTFH VWGZHNNRKEVSDMACQILHQVFNITTTDGYSLALNFQASNIMGDTTTYTPLQHSFINLDF NLR
2	[S] Putative head-tail adaptor	vogp0248	MQAGKLRHRITFQDPVNVQNPTGPVINTWGDHATFPAEVSPLSGREFIAAQATQGETTTRIS IRYRPVENCVSHKRRILCDGRVYNITGVLPDPKSGNYLTLPCSEGFNNGN
2	[U] Uncharacterized	vogp0306	MELTINGKQYIFIFGFHFIQELDKNYDVTESGISLSSGLDNAFINLZSGNIDTLFDMLQTANTT ENPKLSQKGIEECIEEGNGMDTLFDZVLEELNKSDFTKKNTLNFEKEVSNN

Nd Int.	Fonction	VOG	Séquence ancestrale
2	[S] Major head subunit	vogp0001	MAFNPFPTNTMAGLGFHPITATLHSDVTAFIPVDHSQDFFAEVKKGSTVLHFAQNVPMGTT DNHFPLLASGVCASWVGEAESNPTSKATLAKHNLDPKKMAIIIPTSKEVINYTPTNFLTTMH AKIAQAFCKKLDQAVIPGIDNPSSWNTSTSTTTTVISVEDTDITSADAINNAMGLMVAAGW EPNSILSVYIPHPKFNTTKDSNGNPIFNNSTSTGLDNFFGWGSAWAPHTTFDDKVISCSVSNC IRGYIGIL
2	[U] Uncharacterized	vogp0304	TSDKDKFNKKLCVYSNNNZEVLZKISLERLLYLLGSEIKEVPPKLDFIFZDCSFKRFNRIGQE GTQSZSLEGQSMSYYDYDYEZYPYENKSRRZYSDSGRAKEGEVLFLN
2	[U] Uncharacterized	vogp0302	KQMSLIEMDGFLKGKCIPRDLKVNETNAEYLVRKFGELENECASMGNSIDDLQTKSAPASF GIIFLNVSNNDCLARRWALWHEKCREHPDKWRRVAQEGINLAFNAPHLRLPSNYFKAGFR K
2	[U] Uncharacterized	vogp0300	ADTVZFSFTGLDSLLGELEAFSKANNNKACRFASREAAKVIREEASYSAPKVNDPLTKDAIS NNMVSCCSCKLFRRTGNSGFRIGIMGNSTSDKGNPPSNTFYWRFLELGTQHMPAHPFMRPA LDTINDEVTYIFFTQMDKAIDRAIRWATKN
2	[L] DNA replication protein	vogp0288	IMAPTZENNIYFCLNLDANYHSZYFYMNIEIKKAGYSASGETLVLIYRRNNYHSKSNNGVFF FMGSLPNLSEELSWTLDNDIEKIQMKLAFFSKYRILQSZDDQNSYMLGVHAFANLQTNCAH FZPHQRSCQNLDNVHSISNTCTNDIKSDTNHEKIRNIDTNSLLSKFLDSFINFSSKNISKRAMA PAEFMKLPSFQKEQAVIGAKNYIQCYKNEHPDDKAGKYSVNPYNFLSNTTFMDYQEEVKA NAGYZEDL
2	[U] Uncharacterized	vogp0308	MCFCIQYLSVFQRMTIQEYZLRZKAYQLRSLDKEEFIYQQAWANWQVQATKQQGKKNFY PTFKKFFNKKKLDNEILGMETPNSKFKKDNKLIYLMKKANKN
2	[U] Uncharacterized	vogp0284	MNELQDCELENFNQYNRFKVTDMDSPNWVFKKLDAIKAQEKEIKELANTEMERIKERKDK EVEKLQYGMDYLQCLVIEYYPIQKEQDPNKTLNTPYNGSKVIQVSNKQELLQQLEQSGLD KFIKVTKKLAQSDNKKAFNEFAENGTLMDSNGQVLEGASILDKTTSFTVKVSE
2	[L] DNA primase-helicase subunit	vogp0312	MQMEKAVFPNLVENQEYFTPVVPDLKKEYFQSTTDQHIFMMIKNHSNKYKSAPTSEVFEV CLDAFAGISHDYYAZIQQTINEFDPHPSRHFGRLMDMTDEFSVEQRVFHALSKSLAFQENA AKPLDQQNKHINALGAIPEFVCDACNVCFNTTISHDYFEDWEPDYNSYIKKAAQIPFKMNIL NKIAQSSMECTALNFVSAGSNVGKSWALCHLATDDLWQSYRVLYISMEMSQAAVSKRIZT NFMDISMDGIDT
2	[A] Lysozyme	vogp0274	GVVEDRTTDPLKLGRIQVRMMGIHPTQKAMEDFQGIPTEELFWVHPMQPITCATLSIISKCP NGMVEGNQVFGNFLDKCCQDAITSGVLPSIFKDRPNCSEGFYDPNGQYPLNMEYNITNMD RGAHASNNESHNHSTQDNNHAPDINPDLTPCDPKLPEAKHNITLEKGFARDPEFHKNWSLD SAGYPTITGQSMISHSAQNTATSDHEFSHVFWRDATFNYTHITTSEVTKLVGHDLENMQFZ AAN
2	[R] Antitermination protein N	vogp0268	TTPKTFGKDSLARKTEINCHASRRHPPRQPNMQSQNAISSIKFQAPNTSREVZECNRNPDTA AMVFTCTTZTQEYEGSICLPNVALYAAGYRKSKQLTAR
2	[R] Anti-repressor	vogp0262	MNQLITITQNENKDQVVSCRZLHQFLGVKTRYNDWFEDMVKYGFTENVDFIGFTQKRVKP RGGRPSVDHASKLDTAKEISTIRRNEKGEQARQYFIEVZKELKQQLLPQTPEQQIALLAQGN VNLNKKVEQIENSVLDLTDRFGLPSNKAKVLQKKVASKVYMFTGGKYSNAHKKLGAKVF REFYKDLNNRFDVVKYSDIPLSRYDEATEYLDMWQPSFNTTLEIRGLNSQTSFDFEE
2	[L] DNA polymerase	vogp0257	MEFDFETENFVLDITIIHGEVVKYSYTKQYHDFHDANCFTRLADWEALVVNPFIDSFVLRH NDIQZNSSTSYECVNHDSNNDVRFPPYHEVDTFLFSCFIYPNSZDHDFSAYHPSFLPCZKRG VHRSHAWCYRFEZKKAKFDYRSKSLVNRSAFPNFDQDMMDDGFQZVEVSVTRVDHFVYE IHNPYQSMDYKNZGNREYFIDARETFSZEDHADWTKDQRVRDGQLLDGWDIEDIYHKLAS DDSDLVRPTEHDF
2	[R] Antitermination protein Q	vogp0256	LSWFKSNWIKNLMRDIQQVLERWGTWAPNNSTDIYWSPIAAGFKGLLPLRVKVRPICSDED GLIISHCMNPLKQINPKLFTLLVSYYIQGYSKRAMARHHGISHTPIWKRLQKAEGFIEGCLS MLDVHLDMDPKVHLYSPHKPTQSISKVRKYFDNVLHN
2	[L] Terminase	vogp0252	MNPHNPTMSQVPRHFFTPHRIISNSWAPEGTTCCNRAGLSLDPWQEDLWKSILAKHHDCV GLCAAHKFGFSIPRQNGKTYFLRAFVLALWIMTHDSSVIWTAHHISTSPKTFHNMQDLAKR SDYIAPHLEDNNVLRGNGKEAIZFNPSSRIIFCARHRCCGRCFTGINILIFEEAQMFTHHAVY DMLPTSTTSHNPLIILSGTPPNPTHTCHVFTNYRZDALAGKVNHLGYVEFSAPEEAYPDDHC HWYNTNPSY
2	[L] Similar to terminase	vogp0032	MMAVLHHPARDPAPPPIHTQAWDQFKDRFCFLPHAGRAIHRGILYRSRDYGKSPFGDRGN DFAFICTLEHAGFDVWDDAVETKGRNSDRKSLFRTAIGCDNYTVTSNLALSEMCRPVTFST DFDKDGIPIVISRLRDZISITTPSANFVKNAFACFTFLVPTHWKREFSVAIRYAIISRNSACNLD GPITQPPCVFTPRHSFVVQNIAQDWRISPGGRPRARDTLAENREGHGAMDLGAQALSISDA RSPCDDTV
2	[L] Putative endonuclease	vogp0285	KCHQSKQCDGZSQMISMGMNKVLTHHKFEQLLSYHTDTGFLNWKVSRKGTNGIGSVAGC NHGDGYFHVKVHGKLYQAHRLAWFLSKGNWPSNQVDHINCIHTDNRIHNLRZCTHAENN RNRGKCSNNTSGVPGVCWHKQRSKWGAQVKHNGKNIHLGWFDTMEEAAEALNAANEK LHRFNTIHGEAH
2	[U] Uncharacterized	vogp0328	KSVFEQLNSINVYSKVEQKKTGKTYLSYLSWSWAWAEFKKVCPKATYEIKKFDYGEGKZV PYLYDYSSGFMVFTSVTVDDITHEMWLPVMDGANKAMKFESYNYKTKFNEKTVEPASMF DVNKTIMRCLVKNLAMFGLGLYIYSGEDLPEZPQQHQESEAEKLHZRZMQPPENQPEQLEF LKTKTKQEIADIMKVWLARHETE
2	[A] Lysozyme	vogp0341	AILIKITNKILCSQEGIIILSSEAAIIDHFYTQHKGHLAIKALALGRMHISHKGLVLIKRHZEGL RLKAYQCPTGLRTMGYGHTRKVHSHNFNIGEIITQEQAQDFLEEDIHQVTASFNTRIKVPLT QNQQZDTLCSLVFNIGMTTFTTSTFLKKLNFGNYSGTSNDFMZWNKANVNGKRTPSPSLIN RRQAEKALFZSTIYPDWCMDZDWSQN
2	[L] Putative helicase	vogp0338	MIQKIPYQYHAVAPVRLHCESATIHDMRLFPTVATDAGKFFLNACFANWNGRPCFLAVHR LVSLDQIGEIIKDNDNKVHTFWIDTVVTDSDDYYSNVHVDCLFNZDIYNGPDRIHPHCYHL GAVVHVLVNRGRLFNFDKZGGHTIIHSFVSGTTKDCLIEDNSAAIFKNNVPAIAFTKDMFHD IAGDYRLYSSGKINHMVGDGYQVDNDHNDAAVVDWARLTEVNMDATZSLDNAMKLDD VYHLAIDITTSCIHS
2	[U] Uncharacterized	vogp0337	MNQQELMGYLLVEMEKWGIHICSFECIPQHAMVNINSRIMIYNPNZATPFKIAHELIHVINK DNRRCGECDTLNPQEIRANQEAILLLWEIFKANGGSYEYFNLFFDITDSPFESAKSIIKNNKV SINKQEMHDYIIDYISYFDIIKRNNIYQFLDLYRLSHNFFNMAQKEFQQLLGFDLV
2	[L] Bet protein	vogp0336	LAKQIGMYYANPQESTTTZNQSNCRGQASDDQLATLLAISNQQSLNPCTKEVYSFQYRQN GTLPVVCLDCWSNITYQNHQFDGMEFGLDNZYWACLLYHKEGNLHTWVQEWVGDWAR EPFKTTEGPEITGPCQYHPKHMZNRQANSQWTNZACSLSGKFDKYQAZRMAZPTDYSANY

Nd Int.	Fonction	VOG	Séquence ancestrale
			RQEEYPNPDKEPAKRDKDPZLTHLDYSWEENZSPLCSQELKEKAQPHZELTQEKAEEAZGF SKQEST
2	[U] Uncharacterized	vogp0335	MRGLAYNPGILPAEMIIRHRVKPMPSREELLKRNSFPSVNQNKYLNAMLRKGEKQ
2	[A] Putative holin	vogp0334	MNEVZFNFTVSQTSRIFMFASEVKSLYFMLFLIFINFDIITGISKAIKNKDLRSNKALQGFIKK LLIFLVIIFANIFDTFLDLKGALIMITIFFYIANZGLSIMENLAQMGIZIPQLITDNLQVIKNDNK QZDDKDDQEEDRCN
2	[S] Major structural protein	vogp0333	MAKGTTKTANLVNPEVLAPMIHAQLDKALRFAPLAQVDTTLQGQPGNTLTFPAFAYIGDA QDVAEGEEIPTDKLGTKKNSVTIKKAAKGTZITDEAILSGYGDPMGQSAKQLGLALANKVD DDFLEASTSTTQTVYANATVDGSNAALDIFNDEDSQATVLIINPTDASKLRKDATTKCIGTT EVGANTFINGTYGEVLGAQIVRSNKLAEGTAZMFALKLILKRDILVETDRDITTKTTLITSDK HYAAYLYN
2	[U] Uncharacterized	vogp0624	TKEVINYYMKEFNVNFEPAKIYFPNCEEFKEQVNSITQAYSNHLVTVESYEEDKKARTQLN NLYKTLNNKHMDINKEYNQPFNEFKVRFKKSETVINRAIAQIYTGIKDFKDKEKESNMKDI HZFLGKLATKCKVASRVLDNFKEKYDKFSLKWYFNSNKINLKKEAMDYINSLVZGEKDK MEEYKAIKQTVYDHSFEYNLPTTPYIHMLDYSNTLEILNLMNKDLESEKKHKEQEEKHKK AEMQHLEELENQHQ
2	[U] Similar to myosin heavy chain	vogp0329	MMHVLDFNDQIIYFLHTKDPCZVKALREIKINNNSZKLESLISSHHASNLQQHHLVFMQDS NKQWLHFIINNVQEDSEGYSZIERFSSYWAHITKAKPFNPRKFEKKTTSKSLKDVLSDSSWQ VSNTEYYCLHTTSWTYYQTPZKFLKQLCTSYKMVLHFYIELSSYPIKGCYVVLKNKNSLFK GKEINZGKDLVGLTRKMHSSDISTALIAFGPEKDKGKVITEYKAQSQLNLPTNYIWGIWKP QSNDNN
2	[L] Putative deoxycytidinylate deaminase	vogp0245	KTHPNWDZYFLDIANSVTQESNCKRZQVGAIVIKDHRVISTGYNGTPSGAPSCSDRPHCTCC SVPGVKNYYHSSWSHCIDIHAEANAILYSARKGINGATLYITPSPCHNCSKLITPSGIKNLVY PKDYYSSHN
2	[U] Uncharacterized	vogp0326	TGEYVGLGLGNSNQEIRRIQEVMATKFTCYASHZAHSPNSEQMTAAVDEIRRRSNAZRKLH DCWSFPGMVNTDTKDDLGSLDIPPPPDTRPILFTVSGTGVPWWVGPDADIARHLEDSVYFW QPIGPNPHYTAPAFPMGPSITAGIAQAPRLLEHKDPSNRHRIERYGLALVGYSQGAIVTSEL WEYHIKPSTGCLHWVKDHVLKAVTLGNPMREKGKVWPDPGGSPMPSSNSQGISDNCLEF MVDTPYWWRNYA
2	[S] Putative tail component protein	vogp0325	MTFSITFNGLDLSNLVYGFTAVDRNFGSTWTNTLRPNRVTRYGQEFINNTFYAKTITITFIKD GNPNDLISISKELASVLDVNKPSPLIFSDEPNKVWZAFPDGTPTLSEDISSLLATGRNTFMVP KGLSNSAYTNILNSNNSGGLKGSITNNSDSSVDVAINNQGTIPTYPTFKFTHNSSDGYIGIAG SHNSSLELGNHEEADSNTNTNSNILFDSKSYSNFSDFSHATGHSRPYKASTDTNNTICSQDY TSSS
2	[U] Uncharacterized	vogp0323	KMDYRNICEIVKEEMIRQESKHHFDLKHMLTIPKSJADSLDKVFYSYPITQITZIMSHFTNYN DIHERYIQYYNKNCNIAFAYLAGKALGVDLVKVGEGN
2	[U] Uncharacterized	vogp0320	MKLMCKLFGHKWEPLPFTMTRDYCERCFIQKENPRGCITGRDKNFNRSDLDESENVFPEK WLDKHMDN
2	[U] Uncharacterized	vogp0319	QDQVSAZYCKYQKIVNNGNMLHVZLGMSCYMICPNCGKETISRYTTDDSLCGNCTSNFSN TDKLSVEKLQEQLNTAKKYLEQINSANKRKGHLGTIQTDWTNNDSEKALAAIGGDHKL
2	[U] Uncharacterized	vogp0318	MTVESLLKVIZEGMTVILKTWKNRIIVQFECGNDIEAFSCCFLYRKJKIIKJKNRSELIAILEDT KND
2	[U] Uncharacterized	vogp0317	MGYYDTRNEARRISKLASQNISSEQTKKEFELDRQSKFNQEMQAEFHERIKKLGEKNGS
2	[A] Lysin	vogp0332	TPVLMPVINTHQNIAAFLDMLAFSEGTANHPLTKNRGYYVIVTGLDGKPNIFTDYSDHPFA HGRPAKVFNRRGEKSTASGRYQQLYLFWPHYRKQLALPDSTNPLSQDRLAIQLIRERGALD DIRAGRIERAISRCRNIWASLPGAGYGQREHSLEKLVTVWRTAGGVPAKSQINE
2	[A] Capsid scaffolding protein	vogp0191	MTKFRSKFFRIAVEGTTSDGRNISAQWIQEMAETYDPNAYSARINPEHFHWAWPACKFKA YGNAEEMDNDGKZKGKLALFAKLAPNQDLLELNKDGQKVYTSMEINPNFTNTGKAYLVG LAITDSPASLGTKDLYFSRINGNYPFCKSNPYAEFSTTEDGPLELELYELPDTFCTSLSTKRT DMFKGKEASNQAQFDQVSEAVKAMAEHFHDZWSDLDEHKEQKQKSDSQLDKEAKEFAD KKSNSFSTFNQS
2	[U] Similar to EH domain binding protein epsin 2	vogp0210	EFLZVFNPYMQGEKQDNSNMZEMNNYFDISLASISSWIKKGKYPZKDNPHYKHAISNNDK TKESTEHKNQEEKDFPEKVYYFYHKGNLMGTGTTKELSQLLQVSKHNVYSWIQKGKSDFH QNNTLKHAIFNETETKKRFPWLDTYCNSZFNETKEKERRKHETKEERRLRRNIRAQMAIEN SRKEELGLK
2	[U] Uncharacterized	vogp0211	TTTEEIVQNYQVKLLKIIFKEIDSLMKKKEKADINAHKLAENGNTVRTSAYWKSVGNAEFY IKEIYQKLSALAEIDRLFHWSDRLHQEQLKFVSKYPKVMEKYRQTNN
2	[R] Repressor	vogp0212	VVKNQEKTINHLGQVVYQESVEFYKEKLSVYSKHFLKNSLIPQLYEWSNAYKAAVELTKN
2	[U] Uncharacterized	vogp0217	MNFISQGKAITVFPQAATGCYPPISDTTKZYGVSTHQKEARYMSQGCLHSDGFTRFAQDFN YSPNGLLAPFGNCFTNQTSQQYAGRSTYSHTTRILKPKAIANWVYTNCMNSNNKDSGDGW KYRSHGFIQLNGHKFYHFWYLSVKLEDVTTHSLSAAKILRDSAGLWYWSAIASLQAICDFF TLAKFTNGGSNGLWDPQVLHMTGYCMSATKYN
2	[L] Putative replication initiation protein	vogp0190	ARFALZRRHDDVSIIDNKSANFITLCHHPIKDSLSDIZFVSSNPFCFRHFNSKTNSZYDEKQF WKIFNPHSNHSZQCHHHYKLYDSHSNLHHYNDFNESVNFFLYELSKHFNYIFTRYILNIHSS HNCEHYNDNVKZSPLIYDVISLLKFDDFKTQNKVRSLVISEYZKLAEFHINSSHIVLCFHTSY ACQKKNSKHZISIDKFETKISZLISNDZWERQSNYLHNISQNDLNIIVGLVYKKKFHYTYZQR VNKF
2	[L] Putative methylase	vogp0251	MAMTPINKHCTYNLYHGNCLEFIHNLPDHSVHLVITDPPYCNTNPDWDSFLRLHHCWAHY RRLFKPGGSLFLFTNPPFDSAFEISNFLCFNCYWLWDKGNPSGFSNGYNNHWCHHKHISAF RFCTTCNSHRPYFPKNHHPKASKGRTLNIPITSHYTSCFCNHDSSWNNTSKLSTYSICWLNN YKDCFNFHPTQKPISLKZHLIHASSSQGATVLDHCTGSPKTAKARLQAGCSFFGLDNKDAY FKMDTNPMRQS
2	[U] Uncharacterized	vogp0189	MNKEMZKLANDYKEIINKTSDLALKQNNGYIRKAHKWLKEQLFYTSDSSTNKSIELSIDNIL DYQDVAFNKSGKKVZKRZEKQNN
2	[S] Major capsid protein	vogp0192	MCQETRQPFDAYLAQLAKQNAMGFNFSAQSFAVKPPVEPKLGETIQESSEFLKQINIIPVDQ MKGQKIGIGVSGTVTCRKDTSRNTDHMAHDNNHSYKWAQTNSGTTITWAMLDSWANQN EFLAQLRNAFFQSQALDIMMIGNTSAEAAANHSTNPLLQDVNKGWFQLFQNQDPAHVMN EGTNTGSKVIFNGTNADYNNLDALAFDIIHNLIHPRHHRDPCLVVILGCKLVNDTZFTLDNK

Nd Int.	Fonction	VOG	Séquence ancestrale
			DQTPTEKIATRKLD
2	[U] Uncharacterized	vogp0203	TTTATEIIGRASTQLLDQDHIRWPLPELIDWINEAVRAIILARPSACSISAAIQLVTGTHQALP DSIIQLLDFICITDSYSLAPANAIRPFARQVLDAQNPNWHNAPIKNVVKHFISNEYSPRIFYVY PPTDNGIFVEAFLSYLPTAIKVSTNTFDLEEAYINPLVDCISFRCCRKDATCGTDSGPATPHY QSFTTQL
2	[A] Holin	vogp0200	MKNMDTGTMVRTILLILAWVNQFLAIKDISPIPIDEDTISPVITTKNNLFTHVFKKGTQKTKZ FKNNNZSPKGTNQDHVZENNTFTKN
2	[T] Activator of late transcription	vogp0196	MADKSDRIICDYVNGRSEARIKSIESRYLYKQKVDNLGIRTAYSGGSEPESHVLNKEALEKD EELISLQDSMYQFCFWYKPLIKAEKEIIKLKHZGYNGFTWYRVTMELDTQGIEIPKKKAKII YYRFREDIYPLIGN
2	[L] Excisionase	vogp0204	IMTTVIQIAPNEWVTZYVLMAITGLKPGTILRARNESWMEGREYNHISPEGNPKHYSECLY NFKAINAWIENQKQPRPDIN
2	[L] Small terminase subunit	vogp0193	MSFSPAQRHMLAVSATPAAPANAHPAATEHSDSYHLLLVKLEQDHHALKGFLSNAEKVD HKRDNLPKYZPLVNKYLDEGKGYQNHIFTNFMFWLLDIGDLDSALELAHKAMEZNLPTPQ NNRSTSPYIFAEEVSKWAZQPLPTGHSIEPYFSHVFKKMTNQWKLPEKVEAKWYKFAGYS LLKANKGKAEPSRIRHSERLDQANGLRKDTHQLNHNIKVKTKIHHKDTPMNPINKRKNLPP TSCRHVTKASZKSC
2	[U] Uncharacterized	vogp0161	LMNYQKQAPPSLKLNQQHLQNPGESPFSIGVCRTSTTIKFWKDFKVNPPLYPEGDYREDTF TSSTPPKGKLYFKTTCEAVPEGNKAFSNGSKEGTCSYKYGHEDIHPDVTNN
2	[L] Integrase	vogp0218	DVPEHLMDTVNVFFYARKSKDRVPTEANFSARAHLVTCHKERGGNGWDVVGNFMDVGF SSPVCDPNVDRPDLERILGEFQEGECDIIVVYELSRLTNCTHHALQMVNELZDHGILFISTLE DFFDTSTPMGFSIFTLMAALAQQESNLKAERMMGAKDFNANLGSSNCGSAPCKFHFASDPF EMGPVSNKVNNVVISFMESDDDLPDPVHLVHRVAHWYFHCISDNTIHTTFZKDKTPNPCM SEDGTTEKILANA
2	[L] Replication protein	vogp0221	MSNGCILVHRRIQRRWLZKEKCTFSGYEAWLRLLMEVNHSKDEVPVDSQVVTIKQGQHIT SIVMLSDCWNWSGREVKTFLHLSQADDMVENTIDRVKYTLIKIINYDMCQCVCHGNKHSE DTVHKSNQHQSNIVPAYNQHNTNTKNDNNKADNDNEVVNYQKKTTDZEFNHVISFSNVSP SALDNZNYYLENFKLDFHQIVTASSKISZDSGKITHGYSKPIIQIRSNCSZNSVVTRQADDNH QTEQEKQNYKPF
2	[L] Abc2 protein	vogp0223	MPAPLYGADDPRRCSGNSVSEVLDKFRKNYDLIMSLPQETKEEKEFRHCIWLAEKEERERI YQTSIRPFRKATYTHFPEIDPNLRNYRSRYGAISND
2	[T] NTP-binding motif protein	vogp0225	MKJHKDDKJHNNNSRYLIYGNPGFGKTSTVKYLTGKTLVFYLDKSSKVLSGNQNVNVTNF NPRNKIPVQDMANFZELLNSPANKYDNMVIDNISHLQNSCLVNMGRKAKNHNQQDYRHL DFYFLDFMTILZQLNNNVFFTAWETTHQFTZESGQIYNRYMPNIRTKILNHFLGLTHVVARL VQNTKNGSEEFILQPSNGIYAKNRLDNRKGCKVEELFKTTDMDGKAN
2	[U] Uncharacterized	vogp0232	MYTHIMNGREVLTLPTVIGHKHHDLEKREVLGEVIZYTCRRQDGSLYIISRYNTERDKAAM LNYCLSDWGZN
2	[U] Uncharacterized	vogp0234	THAMEILQHIKALDCYIDSQIQQIENLESQALKVTSMHTDMVQGGKHKGKDDIYVELITTK EELENFTAEAIKQKLEFHHQIANLEDIDSHSLLHMVYIDQLGIWQICDKLGISKATYYLKLR QANKYLDNN
2	[U] Uncharacterized	vogp0235	RTYPYILDPETSHMLFDSFHYSISQNICAINZINEQZKTLLKN
2	[U] Uncharacterized	vogp0237	MSTLYQLQGKFQELLNLAKEDETZLEALEDNSFNDDLENKVEGYVZVIKDLEANIEIIKKEN KRLENZNKSDQNKIDNLKNSLYEAMNMTNQNKVKTTLFTVCFHKNNPALVINDEKLTSED YFSKQKHKPDKKKLSEYLKAGKDVSGAKLMETKSLHISK
2	[U] Uncharacterized	vogp0238	KCPCQPCRNKRRKAYIKDYYRKSPRDKIHDLSHZRGVQHEDYSRTEIMPRWGYTCAYCDS KAYTPLDHVHPLSKGGANZAHNIMLPACASCNLSKGTKTLAEWALTFGPKAED
2	[L] Terminase subunit	vogp0194	MNVSSLKNYSKKDILNKIRIKKAKESFIHFTTRTKPNFINVHRFFTILISQELQKFFEDFIDGQ RPSLMIYAPPRSGKSELFSIHFPAWVLGQNPKLRIIAASYSADLASRMSLGVRDIINNPVYHS MFPNTTVNIKNVSTISFKPLDNAKNMFEIIGGLGAHRTTGFGGGITGMGTDVIIIDDAIEDTE EANSQTIRDRIWDWYTTTLRTRLSPKSSIMLGMTRRHHEDLAGRLIKEAENGGDHWRIIKF AAIT
23	[U] Uncharacterized	vogp0635	MSNSKFKLISAGVGALTKSQEMQDVLTNKATAIKERRHGYGQDFHVGKTRANAMVYPKT FKAKKDNFKNNTLLKAVRN
23	[L] Putative HNH endonuclease	vogp0629	DRTGPHRVPFDKNSNIFLKTHNTCRICGKPIDKRSKYPHPLSPVIDHIIPIDKGGNSAMDNLR SAHWKCNRQKSDNEPKVFGNRNLPQSRDWN
24	[U] Uncharacterized	vogp0639	MEILKGNTTSGFHYEITKERLKNFELVEAISEVDTDPTALPKIVNLLLGDNSKYLKNHVRDA EGIVPVDEIGVEIEEIFASQNELKN
24	[U] Uncharacterized	vogp0636	MIEIIIKKYLDGHLDVPSFFEHEGEAPASFVIFEKTGGTERNHSLSSTFAFQSYAPSMYEAAE LNDKVKQVVERLIELDQISGVHLNSDYNFTDTETKQYRYQAVFDINHY
24	[U] Uncharacterized	vogp0634	MGKIKGITVTLIDKVKTGKDPFGNPIYEDKEILVDNVLVSPASSDDITNQLTLTGKKAVYTL AIPKGDNHDWEDKEVRFFGKKWRTVGVPLEGIEELIPLDWNKKVTVERYE
24	[U] Uncharacterized	vogp0633	MDNFATVDDLTMLWRPLKVDEKKRAEALLEVVSDSLRVEADKVGKDLDDTMADKPSFA TVVKSVTVDIVARTLMTSTHREPMTQDSQSALGYSFSGSYLVPGGGLFIKDSELKRLGLKK QRYGVIDFYGDDKN
24	[U] Uncharacterized	vogp0630	VEDVLPNLLKKVHQDFEKYFGESEIINKALAMLKAKKATYKTANEFAIEVGQILSKALGAS FSSDKLPDGKMYYNIAKRLLNYILGRNYKLISGYSTDVQRNLNQKAQIGLKVQVPQLNQD HINGMVNRFSSEDNFEDVLWLLGEPIVNFSQSIVDDTIKKNADFQAKTGLTPKIFRKLAGNC CEWCRNLVGTYTYPKVPTDFYRRHQRCRCTMDYHPKNIDRQDSWSKNWLKNDKNQETIQ REKKIKESDKVE
24	[U] Uncharacterized	vogp0640	MIQTDEEALICDLAETYGIYDYRQLPPYQVAVFSIGLRYDSRIKMAMSSQKVPFDTFLLAGI ZDRLSTSFWFKTKDGQKGKNKPKLVTEVITNTKEKANNEIFFHSGEDFEKYRQQLLEKGGG ZD
27	[S] Minor capsid protein	vogp0011	TLTHHQLVLQAQRIVDIYPKLEKDLFTRIIQRLKTKSYISSAHNLLVWQVEKLNKMGILNKQ IIKLISKYSGISQQKLFYIVKDKGFQIFKEINNYLNQLEEALAIVPZVCNTHTILDNLTRYYFQ ATYNHYKLINNTMPYCAMRAYQGIIQETTTZVFAGLKTHHQALHHTIIKWVKNGFHANLD KASHHWTPEAYARTVINTTTHQVYNNVQEEPTQELGINSVYISQQATARHACSPFQGQVIS LAKASENN

Nd Int.	Fonction	VOG	Séquence ancestrale
27	[S] Minor capsid protein	vogp0488	MTVKVNVDLGSVKGKVSPEAVAQGQFTLINQVMLDMDQYVPYRCGDRSSGYTITNGNEI TWSTPYARAQFYGINYNKYHSFKFNKYTTPGTSKRWDQRAKSNIMEDWEKAFLKGMGLE N
27	[S] Tail protein	vogp0467	MTRQKNALRGHFVAPYNGGTEPSKVTEDTWLELAKWISDVSDDTDEKTDDQAYYDGDG VEETTVVSVKGAYTFEGTYDPDDKAQALIACMKYKTGDDRKLWHKVVSSDRKKQWVGA ATATEIKAGSGAASDYEAFGCKLSYNSTPKETGIGIPKKNELSVAMTN
27	[S] Minor capsid protein	vogp0029	LIQQIKDLVRSGANYIGITQSLTNITDHPKITINQEEYDRINTNLDYYQSKWHYIHYQNTDGN TKKRQLNTINAKTAAKNIASLVFNEKAKINVKDNAANEFFSDILKNNRFNKNFERYLEYCL ALGGLAMRPYIDGNKNNVAFVQAHLFYPLQSNTQDVSZAAIATNSHKTKNYYTLLEFHQ WQASDYVITNELYQSDYPDKVGTQVPLSKLFKDMEVDAQVTDFTRPIFTYTKSNSTNNKDI NSPLGIN
27	[U] Uncharaterized	vogp0187	MLSLAYPLDNTFKFGGKEYNMDLSFNKVLHVFNLMEDDSLTDSYQAYLAFDILLGQDMN NIEETAYFLIYIITNFIDKEHQDSFRYDIEGNPMPLANNKEEQENFFSLTQDADYIYASFLQD YHINLLEYQGKLPWNKFKALLDALPDNTIIQRIIAIRQCESPCGEGEKERNNLFKLKDHYYL DDQDEEDCWSSQ
28	[S] Minor capsid protein	vogp1149	MRLLTAMDKCLLTDILTIKKVTDKDDZGHLVYCEPFTIKHFRFHPHIVSSGNNNSHTGTSNN LVFIYFYPSYSLVTVNNSWVGSKVVYGGREYTIHKIITNYHPFSNEIFSZEMEVI
28	[S] Minor capsid protein	vogp0509	MNYFENFVSQLIKVSNLPIKPRLDYLTDHDDLAIYPMPGCKVNDEYMDGTQEVSLPFEIAIK TKNQQLANTTMWLVTSALANFNSDNPSSNNSYKFMSLDVNSNSMKDQDNQGYYTYILDI TANIDIZGNNQ
3	[T] Putative DNA binding protein	vogp0362	EIMNRIKQLRKSRKMTRVELAEKIGVTKCTILNWEQCTSYTNPHNSQRKLADFFDVSVPYL LGZDTNZTYSNNEMALKDAIGDSIVTLVVLCLQLGYDVEECLKIAYNNIKDRQGVMN
3	[U] Uncharacterized	vogp0126	MFNIDHSQAKDFGSIKDGTYEVIIDNANQDATKNGAEFIDIHFRIRKDFQQEFQNNNIFHRIW NDKDANKYPMAAFNNIAKAAGFPNGTKFNSLEDZLNHLLNKAFQVTVKNEKSEYKGKTY KNLNVKALAESNIPCNANPVEISEEDLPFF
32	[L] dTMP (thymidylate) synthase	vogp0801	MKQYSNLFRDILDNGYHEEDRTGIGTFSMFGPKLRWDSREGFPFVTTEKMASKSVIGESLW FTSGSTEINHSHEIAHGTHNDFCDZEHDVWESNYKDKEVNNNVGYTNDDLGHMYSKHWY NRNIHPVYKFIIDIDHVNANPTYHNLMVKAWNPYNENEDHVALAPCHFFFQVFVSKQGRFS FEWYKDSEVNFLGLAFDVRSYRLSVHVMSKRSSLEVSNLVFSGSNLDIYNNRVIQVYEPFN HEHRQFAYFANIY
32	[L] Frd dihydrofolate reductase	vogp0802	MMIKSVFASGKSSTFHKNGKLAFGNKNGLPWGHIHEDMLNFKETTKDSFLVMGTKTFKSL PNNLPNRINIVLSTSNHTYRINAKNNZGQRPNIYMHCHFTHSSSKLQNSYNNISVIGGLTML KEALHLADQVFHTIILKATEEDTFDSDIQLSKNFLQNIYFDFYVMSNHYFGNEAISYTSIHKN KKQF
32	[L] NrdB aerobic NDP reductase, small subunit	vogp0805	MSKTIFNKNKVNHLGQPLFLGEDPNISHFZTIQHPIFZQLTKKQZSLFWRQEEVHLTLDSLEF DPMPEPWZHVFTQNLKCQCZSDSLQGRSPDIAFSPIFSDNSLETCITTRDFSETIHSFSYSHIM RNLYYNPZAFLDEIILDKAVLARTEAVTQHZDELMDETQKZNNNNLHEAHILMQESDFQEA LYLCLHAVNALEAIRFZVSFACTFNLAEQGKILEGNPEVMKLISHEEQLHLKGTQALISIWQ NYKDS
32	[U] Uncharacterized	vogp0876	MAIHTSVCSRYTPKQICTLMSSINPPSVILGFASPSGKAYCSNRAIGHCCFAFGMEYNFKNH YYLNFYNILSPYNKYLARKHNGHNIIACGNSEKIFKAMEMVHPIYNNFHSLAFYTQQLHTR NVFQVLGHNLNTPSYFVICYSKQKNGIISGGTNTAWQLALRHSIPCFNNNHQDNIKRLKELL DMKDSSZPHIHLICSYFLEPR
33	[A] rIIA protector	vogp0791	MIHNEDKEIFSSTTNKSTGFNIKASPKAFKILSSNLYKYKERAIIRELSCNTIDTHKETGNHNP FHVHLPTPLDPWFSVRDFGTGNSSKDEVTGLYTTYFASTKDNSNDYIGGLGLGSKSPFSYT NTRNITSYHNGKITIYSTYMENGKPHITHMSNNKNTNEPNGLEVIVPIPDQDISNLKDEANQ VFRSFNDYAPNIIINHIIFNFPDFNNREVFNIHSNHNHCSNIYAIMGSIIYPIPKDSWZDNMIFK DN
33	[L] Nicotinamide phosphoribosyl transferase	vogp0808	MSMZNIPLITTTDSYKITHWYQFPHGTEYFLFYVEPRSGKFDNIIMGGMDYVTPNLQEIVTN EDVTQTQEMFKAHFGQDVFNRKGWDEVVTNGYLPVQIHAVQEGTVVPVKNPILTIQNSNP PFTWVASCLETFILRAFWYTSTVATLSFECKNMIRKHLNETTDLDFNSPQFNFVLTTHLHDF GSRGVSSGESAALGGVAHLNNFMGTDTFEALYTAKHLYSQDMAGSITSISVPAREHSTITS WN
33	[L] NrdG anaerobic NTP reductase, small subunit	vogp0807	MNYMDIHPFDIVNGQGIRVSLFVAGCZHHCEGCFNQSTWKSNTGKEFTZDNLDEILDCLDD DYFQGLSFSGGDPLYHRNLEDVTKIFQKVKAIYPNKSIWLWTGYKLEENMHKPNEIMKYID VFIDGKYDKNLPTIKSWRGSDNQNLWSNVDGVRN
33	[L] NrdD anaerobic NTP reductase, large subunit	vogp0806	IIKKDIMLDIGSSSDINKEKAYKDSNVIPTMRDLMASTLSKHHPLEESFPNHLIKAHMDSNM HIHDMDYSPLIHRSYSMLIHFKGKFQCGFKMGNAQIEQPNYIATASTMVAQIINQVYSPNY GGNTIPTMDQVLAPYVPKSYYKQZEMAHEEQAHSNAGFFSHKRVQKQIYDAFQSLZYQVN TMLTSNCQTPFTNINFPKGTSWEEKLIQKAILQVPIQGLGNNTMTPIFHELVPYLKEGFNZQE GEPCYCIKQ
33	[U] Tk.4 conserved hypothetical protein	vogp0794	MIINYIKGNLIALFNKGNYNPHHQTKDIMGQGCNCFNTMGSGIAGQISKDFPKAZETDKKN SCCDDNKKLGAFYNLDNHFTVHNSCGVNLYTQFPLWNPSHDSHYTAFEZAFKELNHYLED TNQNGDNATVYMPMIGAGMGSWDWEVIKDIINKATPNIDIIIFDYEDDMFINLCSN
33	[A] rIIB protector	vogp0792	MNKSKIIISKYNIKCLAHKEQIEFSKLLGYGKYTTQKELADYHDISTHTIPHILKNSKEAKRD EVMDSEEHTAQDNTHMVKGRZSSKKQVVTNSEIIRNASTKFISITSGSVPYNATPTSHLNFK EILDALVSNNFKKAIQLINNKAIEKYFYGNFSIEGGTLFYQNIELRSSLVNRIMDSMEKGQH SNFKFYFPFLENLRENPSQKAVYRLLDFLVPHDIEITEDGYFYAWKLVHIDYZDFYSHTFDN SPGKV
33	[U] Uncharacterized	vogp0793	ILTMRFLTILILLFTSHFFAGPSAPDFTDTQKTNLSYAYZYGKGYNQRDLGTILAAIAWEESS AGLNTGNKGHHAYGMFQNYZKTVVKNMDQRGIPFPHCYMKKVLENRSGSAZWAMDEL HYWLNVRSNNIRLALASYNAGWNYKAGKAYADRVLSKNQRLKSMEFIYKKCNVN
34	[A] NrdC thioredoxin	vogp0880	MFEVYGLDSSDYKCIYCZKAKRLSEVHKMPFNFIPITKEKDTKDHKDNTEYLTNRVQDTG NVLTTTPQVFSDGYHIGGFDELQAZVNN
34	[U] Uncharacterized	vogp0924	TTVQIKKPTILTDVDGVLLSWASSLPYFSQKYDIPSDRVLKTIMDEKFVAAGELFGCDQHLG VNLMQKYNNSDFIKYLAAYKDALCVVNKLKEDYNFVAITALGDSIQALLNRQFNLNTLFP GAFSEVLTCRHDESKVDLFNNVKEKYNVVRFIDDLAHHFHHSQDVLHVPAYWLTRHDEN KDCYHIEKSLDDFNQKIIKN

Nd Int.	Fonction	VOG	Séquence ancestrale
34	[S] Baseplate wedge subunit	vogp0899	MLFSFFHPINYKGITIANIFKNYSNYFNQVISSFIFHPYYIQGYPRPEQVAHKLYGNPSLYWIL LMLNNIYDPFHDWIKSQEPVYQSAIQQYKNIGNNNQVLYHIDTKGKRYYNLIEYPKGSZN WYNNGHKAHRHFQYHGPLAPVSTLEHEFNHNEENRNTNILSPTHIHTFINTLIRQMEKAH
34	[U] Uncharacterized	vogp0900	MPGLTYDMAPTTGHGTYPPTIVSATQCKVFVEGIPVLVAGDAIVPHTNTIKPYDTHSGYVIP STSKVFVEGKPATHMGDPISYGDTIAQSSSKVFIK
34	[S] Baseplate wedge initiator	vogp0901	MMIKAPSITSLRINKLAANYLHLNWDDVGSNFFYIVEFSIANGYGCEIIHNFAWIHPGVTPD QDWFEDNILDPNTNYEFRISTTFEGFEPSNWVZSDVFQTFNTNAYYISTMAQFIPANVFIKN KLTNNNKNYVNFDKDPNATLMNENFVFYPGIYNITNIANFIAADEDSHLVQEEVQKVCVDI NRTFLAYMDDVLYTFERFQHMAKVSNDGGQNWRYYQAFNGRIGYPVSRTIIYQTNTSSYL LGYDDVFYGR
34	[S] Baseplate wedge subunit	vogp0902	MNKTSIIYRSIVTTKFRTKNLLNFYNSVGDKHDKNTIYATFGRPDPWSYNETDSNFAPPYPN DSIEGIADVWTRTLGTVKIHKSLLRPVIPRKDWGDPRFNNPFTFLIGDIVVINSAPYNRTDFG SGSNIYRCIDVPEVNNCSINSIVDKGECMKMGGKWTPSRQSLEPPSSSDNTGKAIDTGDGYI WEYLYTIPPDVSINHCTNEHIVVPFPQELIAYPSRWGYQHIIMWYPNNYDFIYRIKVHTLRN
34	[S] Baseplate wedge tail fiber connector	vogp0903	MMIQETQKWIDTGNVGNASTGDILYNGGNKLNDNFDSFYNTFGDHRZMDSADGSGPDSII LPTGCYHHKHCQSYSSNPVKVGSLHDINTTTGNLTFTITDGKLGKGIHVINSDGSISITNPKN RASDSIAGFSGHLDITNPYSKVTFWCIASDPSGSDWDYGVKSLFGHTKIALNNTFYITSTVV PDDIPLSGKTQYNTIKLLVTGDYGASANVKTSKIFVIVDAITTKVDTTEYAVLQITDYDEDD DLAYIGSC
34	[S] Baseplate wedge subunit and tail pin	vogp0904	VIINISMIRKTRNDAVNITNDVTHVEYLTKSSVHYIFVYIVIVMGNVNTMDGPTTYSLIVGLD YHDIRFAIDEFQALTSFLIDTFYITDVNALPEYNIRQVKFVIFTGVTTDSQLFRDNEFITHVYTI PIITNISTDTNITCDQVFAKFYDIVYNDILFFFVDNQLCTTTPVFELQFLDSTDHVYIKDSFSD DIAVTREVFVPGVPGFSTRDYLGEEVKTLICSILDNPIILYDFERMDTYN
34	[S] Neck protein	vogp0905	MSGYITNNPRELKDSILRRLGAPIINVEVTEDQIYDCIRRALELYSEYHLDGFNKAYSVFYLS QEEEQAHTGNFDISASTVFAITQIIRTNATMDGNATYPWFTDFLLGLAGGPSGIGNCKYHGP IAVRGNLGYFTQLMSYQNMMKDLLSPLPDYWYNSVNGQLQITGNFREGDLIIMEVYIQSYI EVHKSVGGITGYSAVASCSPTDTTSQDQWNNPYSFSRCNITSSGPNQFTKQGAYNIRWVKD MATALTKE
34	[S] Neck protein	vogp0906	MVTYNKTMFAQLETRKGYNQTYQTNILNPYVNLHKYKNTQTLADMLVAESIQMRGIELY YIPREFVNLDLIFGEDIQSKFTKTCKFATYLESFEGYEGHGNFFSKFGFQVNDEITFSINPKLF KHQVNGKQPKEGDLIYFPLDNSLFEITWVEPYYPFFQDGKLAMRKITAQKFIYSGEEINPEL HHNKGIYVHKFSDLDLAPIKNLDGLTDITLDQYTEDNPFHZEAKDFIEHFDHIISKGTPISHS NSNKPTN
34	[L] Terminase DNA packaging enzyme small subunit	vogp0907	SMEGLDMNQLLDISGLPGISGEELEVYEPSVLDQVKSHPKDRTTDLEDDYDLVRQNMHFQ QQMLMDAAKISLENAKNSDSPRHMEVFATLMGQMTNTNKEILKVHKDMKDITYEAVDTK EAPPPRQPNIQNATVFMGSPTELMDEVGDQYESLDDREKVINGNTNLN
34	[S] Prohead core protein	vogp0910	MSKESVTAKWKASFQGRILETHKNQDFVSETVZDFIPPEADRMQEMZAILDKDDIIIIVKSM SEDNPNLAFAMLSIIENMAIKEAMVKHVNSCGEVTRTKDRKTRKRKASQTTGLSKAKRRQI ARKAAKTKRSNKNIKRKAQRKHKKALKKRKALGLS
34	[S] Head vertex protein	vogp0912	TTNSNSIGRPDLVALTRTTNNLIYTDMVSIQPTNQPFATFYGIKYLNPADECSCZTSPTYPCD PGYLHSEHFTTLTEDSKLTLNEGHLFKFNNYVFEVFEDTHFATDEDSTIEZALHIAIMLGKV RLLSDATYSYKFQDSDTEIPZAHFQIDKWTTSVSSRKLKTSITIELLQDLEADRRFSPDFLED LLATYMANEINKDIIQSLITVSKHYKVSGITHYGFIYZHYPZATDAACSLYRKVCCHTSFHP N
34	[S] Head completion protein	vogp0897	MAYSGKFVPKNKSKYKGDPHKITYRSSWESFFMKWLDHNPQVICWNSEEVVIPYFSNADG NKRRYFMDFYVKLDNGHVFLLEVKPNKETLPPEKPNRNTAKAKKRFIQEIYTWRVNTDK WKAAIELCEQKGWNFKIITEDTLKNLVCZGVKN
34	[S] Baseplate tail tube cap	vogp0922	GMMSNVIMITMNNDTMDSFKASDKTIGCFTSKTTGETCTAQFPTHRASGNDSSRHZNINYL YNNGFLFPAYNYSSITNRSLNYFRKKFNISSSFCNGSFNISGKYMTIGGLYTPFLNEDISHKZS HNSQSVLDDTSHNFTDIAESFISHGGSDLGSALSNTASTAVFGSLESSFRGYLANHCEQIYDT SSSNMKGTDSRTQTFMCHLTPNSLEELKDVSKIFQTFLZLSYASSGNSSSTKZMKGYVDAW YKNTLL
34	[S] Tail tube protein	vogp0895	MAMDFNDITRAFZSGDFARPNLFEVEIPYLGQNFKFQCKATTMPPATVEKVPVSYQNRKIN IAGDRTFDDWTITVYNDEAHNIRQAIVDWQAMAHGLGNDITGDTPAQYKKPAIVRQLDRN GKPINEDTIYGFFPTNVGEVTLDWDSNNEIQTFEVTLALDWWELN
34	[S] Head assembly cochaperone with GroEL	vogp0926	TSEVKDLPIRALGEYVILDSKAAQAGEEVKSZAGLILGQKRTQGEVPLFGMVISVGQDVPK GIKEVGDMVFSPLNNMNNVPHPCIAZGLKQPDEENDHKLVTSHYKDIPTIYNSKPQGN
34	[U] Uncharacterized	vogp0927	AIREILKTMLSMGIPNFVFEKADGTIRTMKGTRDPDLIPAZZHPFKPTLENGKLTEETRKEST DTFPFFDIELGGWRSFCLDKLISVNGMNNN
34	[T] RnlA RNA ligase 1 and tail fiber attachment catalyst	vogp0931	TNIKELFYNLMIMFKSSYKGKFFFTDDISPMGTKFRIFSYYFASYSDWLLPDALECRGIMFE MDDQDKPVRITSRPMEKFFNLNENPNTMNLDLNNIDYLMDKADGSLVSTYLDSDSNFLKS KASIHSNQANTANGLLNSPNHKDLRDRFTDLGQDSNFNLEYIAPSNHIVLAYPEEQLILLNIR HNKTGKYIYFHNLFKDPTLRPYLVDLYQVPPZETNAADWVEEVHTTKDIEGFIAIMEDGQS FKIKTDWYV
34	[T] RnlB RNA ligase 2	vogp0933	MMFKKYSSLENHYNQKFIDKFYTZGFTGGVWVAREKIHGTNFSLIIERNKVTSGKNTAPIL AAZDFYNNN
34	[L] Single-stranded DNA binding protein	vogp0934	MFKRKDPSQLQEQLATLSAKKGFSKADEWKLTTDKSGNGPAVIRFLPSKGEEGLPFVKLV NHGFKENGKWYIENCSSTHGDFDSCPVCQCIZEKNWYNNKKESSLFKRKTSYWANILVI KDPPAPENEGKVFKFRFGKKILDKITAAVNVNTDLGEAPVDVTCLFEGANFSLKAKKVCGF PNYDDSKFSHQSQIPNIEDAYQKZLLEHMHDLTDMIAPSKFKSLEELKTNFKQVMGASKPA GAAASTADAPLNN
34	[L] Loader of DNA helicase	vogp0935	MIKLVLPPRPKIWINGNSVYSLYLTIKDHMTGKYDIFKYKWGMHASDPAFNKHRDKDFFE NLAKKFTLKELASILISNLAANPDAWSGDIANADALAFYREYMGRFEHISYIFKEEVKHIFY FSNKEDINFKDLIYYNNGHPCIFKLVQSSTISFETFIILDSLLNFIDNHDKMSEDDIFWQNYST NIKAYHKLLMINKEQAKDLFIKIINQCKFMS
34	[T] Late promoter transcription accessory protein	vogp0936	KKKCZSZKVMDDLGDKDHPIKGSYNQHICLEIEEMVNQZGLTYLEACSHFLEENYIDITHFP KFIHPTLIDKIKQEAINYNTLRPSFIPSSNFNKLDN

Nd Int.	Fonction	VOG	Séquence ancestrale
34	[T] DsbA dsDNA binding protein, late transcription	vogp0937	NSEVNMAKKETVEFDQAVHGEDLAKLIKEASDHKLKISGYNELIKDIRIRAKEELGVDGKM FNRLLALYHKDDRDQFEAENEEVVELYDTVFTKK
34	[T] RNaseH ribonuclease	vogp0938	KTTASLMYLVNKSEEDYPQGIRLIDFSQITMATIMDTFKPKDKINVDMVRHLILNSMHYNM KNYKEEYPIILAIDYSKGGYWRRNMAYYYKKNRKKDREDSNWDWETIFDSINKITDEFKE NMPSNIMDIDKVEADDIIGVLTKYFSFKGPNVLIISSDGDFTQLHKYKNVKQWSPIQKKWV KPKYGSPZKDLMTKIIKGDSKDGIASIKSRSDYIITRLEGERPPSISTKLLEQMFKAQDHPSVL LTEEEYNR
34	[U] Uncharacterized	vogp0941	MQVIINKALYFGKEITGTFESMGKAWRDTDPAPGEGKVNVMIEGKPRTVWVNKDDIQYV YEDGDNAVAVEKVEESYEEMNKRITKRFKVMDMMTIGIIKGNIRSLIISGAPGIGKTFTLEH ELNKAQDVGYIDFKSVKGKCSGIGLYIQLWENREDDCVLLLDDVDVFSDKDILNLLKAAL DTGEERKVSWATASSYLEEQGIQQELEFKGTIVFITNVDIDRELDRGTKLAPHLQALVSRSI YLDLGVHTNN
34	[L] UvsY recombination, repair and ssDNA binding protein	vogp0960	MNLEDLKEELKADLNIDSTNLQTEATNNPVLYSKWLRYYSYANQEIIRLEAKKKKALKDR LEFYTSRSNNEVCMDIYEKSELKTVMAGDEEVLKVDTTFQYSQVVLDFCCRALEAINHRGF TIKHIMDFRKLESGKV
34	[U] Uncharacterized	vogp0874	EYPDLKKDZMTDKEEEDQEVVEELEEMQAQIEAKAEKKANKLLKKNCREIKRLNKHAEQ ALIDNNKDGYIYAIDKLRTIYKQTLTPDEFLDTLWETSRQQVLDMAKSFSPAKAIQSHVEY HQQTVPFCSSVN
34	[U] Uncharacterized	vogp0917	AKICVVCKTPIEDALAVETHKGPVHPGPCYHYFEELPVSESSEEQLDETQLLLN
34	[L] Topoisomerase II large subunit	vogp0861	MIMSEDFKVSTHKQHFMMRPGMYIGSTTYEAHERFLFGKFKQISYVPGLVKIIDEIIDNSVD EAIRTNFKFANKINVNIKNNQNTVTDNGRGIPQGMVITPKGDQIPRPVAAWTHTKAGSNFD DNNKNMGMNGVGSSLTNFFSNZFIGKTCDGKNTITVPCSNGAENISWATKPGKFKGTYVT FIPDFNHFEGCNMDQHFMDIIZDRLPTLSVIFPQINFNFNGKKVSAKFKDYAKMFGDDSFIK EDNNFSFAIT
34	[L] Dda DNA helicase	vogp0267	SSVSTFYMSTNGQKHAFNIFMEAIZEKKHHITISGPAGTGKTSLTKFIMQDLVSTGKTGIILA TPTHQAKKVLSKAAGQEASTIHSLLKIHPTTYEDNQVFKQKNVPDLDEIHILIFEEASMVDK ELFKILMKTIPPRCLIMAIGDKYQMHPVNPDKNYISPFFTHKHFYKVELNEVKRHDKDIIQV ATDVRNGKWFYNKWDKDHSIQFFHSPTLNEFMDNYFYKVKTPEDLLENRILAYTNNSVDK LNCIIRKH
34	[U] Uncharacterized	vogp0269	MMMZFEHFLAEAPMDATKDIEVFMDKIYSRKTMEGLDELEAYYDKRKKEINLKDTEDISI HDALAGNNKQLEAEDKAEEEDEEVFLNNN
34	[S] Wac fibritin neck whiskers	vogp0316	MMIEQLTMGELPYIDGPPDVGQAHIDWIKNGECLCGTSTETSNDGTLNHPTIKVQKNIETIN NDAETTMDKVDKVVDTINTIKNILGIIGDTDIMDQVNQNAADVEVLKQHVVDTTHHDHIT NNNIAKINITIGQRHPTADPATRTIYNDFSCIKKELGTYDSFNINGCPTIGSPGSGIKYHIIQNT TSLVDHGGRITQLENNWKDSNIGQLTSKVNNMRSEMGHSSLATDKTIYTRLRTVEDYVTG EYNDIAAI
34	[S] Tail fiber protein	vogp0354	MAKFCQPFRATSGLDAASKKVINFAYPDKTICTDGVNVDFFIZENTIQQYYPSRTYHKDFTII YNNRFYIAKVDITTPNAFNSNKWKATRTDPNWDVITSTSNQVNFCZGQYISSHVSITNSVFT LPSNPIDGNTITSKDIGSKHSSNQVFIITSDRTINNINGHATRSFSLASSTTYLVYNCTNWVCQ FWQQYNSNRTHSSSFVPPANYSLPKHNDHLVRHLTYWGRINIILPRYANHGDIINTIDLDGL NPIN
34	[L] Recombination endonuclease subunit	vogp0609	KKFNFNRVNYQNIMSVGNKPINIKLDKFKKTLITGTNGAGKSTLMEAICFALYGKPFRNINK AQLINSNNKKKLLVELWLEYDEKVYHIKRGIKPNIFEITKDGKQINEAASTKDYQAYFEKKI WNNSASFKQIVVLGTAGYTPFMZLPAPKRRKLVEDLLEVSILAHMDKLNKALIKELNQEIQ FLEVKINRIKRQLKTYNDYLKDQQKYTDDNLAQLQAIYHEAVDEAKPFKSKITQLKDQLSN FVKHVDPSK
34	[S] Baseplate wedge subunit and tail pin	vogp0614	MKQSINIGQVVDDGTGDYLRQGGQKINDNFTELYSKLGDGAIPHASGAWKNHTPPCSPTLT PNFGKSWAINTSSGPISVNLPTNKASDYSKAIHLRDVRGTWATHPVTLNPASGDTIKGSPSN RKLYKDYLDLELVYCSPGHWEYVDNKRVNSITTSDFSTVAHKEFLADKDGRQDFSNVFGS TSYNLATIEVYHRGNLLYYGNDLSDNSNYGSITPNTPIDIVPSDGNCIKLRTFPCSPGDTIQIN TYMDGVAC
34	[L] DNA ligase	vogp0621	MFILDILNQMAATDSTNKKQKILEZHKDNKLLKTFFQLAYNKRFNYSIKKZPNPTVSIHSFS MLSLEDALDFMNSNLATRKLTGNAAFNELTNTLTYGNTDDSEVIRRVMIRDLECGTSVSIA NKVWPDLIPKQPRMLASSYDEKSIANIMKPSNFAQLKADGARCFAKVHIDSEDITIFSSRGN EYLGLDKLKEDLKEMGIEGCVIHGELVYINNZFLTPHQDSMKADDSDNLYKAFDLEDKAE DPVQVAKSYN
34	[S] Prohead core scaffold protein	vogp0719	MHMTIKEQLLSEAKNITIAVALDSIFESVELSPEVKAKFSTVFESVVKQQAVKLAESHIAQIA EKAEDLVEEKKEEAYAETEKQNSEQVNKYLDHLVETWMAENKMAVDNGIKVQMFESLL GCMKDVFVEHNVSVPEESVDVVAEMEEELQEAKDESNGSLDEVSELKEYINYIKRDQVIAE ATKDLTESQKEKVTALAEGMDFCDTFADKLTAIVFMVSAFKPEEQAATQIDKNFNYEEEG TNVDPKVDN
34	[S] Tail sheath stabilizer and completion protein	vogp0765	MFGYFYNSSIRRYIILMGNLFSNIQVNRGDRFIKVPITYASKEHFMMNLNKCSINSQEDVAK VETILPRMNLHLVDFTYNPTFKTNITNQNLPHZAAGDIPNVSQYNPFPYKFIFELSIYTRYED DMFQIIEQILPYFQPHFNTKIKELHGNDIPINRRDIQIIFMSAAPDEHIEGDNFSRRRMEWTLT FEFNGWMYPPVNDVKGZIRTIYLDFHANKRELNYPEGNFESVDYEVVPRDVDQEDWDGTF KETFSH
34	[L] dNMP kinase	vogp0804	MLISLRGKKRSGKDTTAHFIHNNHSYNVYQLANPIKDTLALSMDLASDLSITHHLGLTYKD FEGZGYDHQADLTFTIYEVINLVCQCWLYLYNNNFYFADDTYDNITKVFLDHNEPWSIRRL MQTLGTDIMVSFNNHZCWFNLFNKYYKKDFNSDHKYFIITDIRQAHEMELVRDLGTTVFHI HHPSTLVYNHITEKGLPLLEGDPVINNNGSLEELYYQIZNTFKDLKCKN
34	[S] Baseplate wedge subunit	vogp0812	MATKKKTTNIPDIFNGATFDEIKRDLINWLRNQDEFKDYDFKGSRLNVLTDLLAYTTLYIQ QFSNAALFESFIRTALLRSSVVQHAQDMGYLPSSKTASSTTILLKASHPNNPTSIRIPRGTKFI GSVENTRTDSYPFVTSEDVLFIRGHNNPYNPLLNLDQGRIVRTETIFHNASQILIRDPNIDRTQ VRVWVDGSPWTDWTNKSMVNITGTSNIFYMRETIDGYTEIFFGEGETSIZTAGGALEANYI GGLKP
34	[U] Uncharacterized	vogp0898	MVEILPTSSKLHDISEGKTFYVTFSSKAAPADTZFEDIKISNYIPNKGITVEGTKFSGKYQDSF SNGTNSLKYRTNRDRINKSACYFDDLPSNPITTHLYYFKAPTHLQESYTYKVTLTYNQSDSZ SPGCSSKGNSGCSNSNETNTPPVLTEHTMNKIYTQTVLGNWDVWAQQLRNYIHRGGNFP

Nd Int.	Fonction	VOG	Séquence ancestrale
34	[L] Topoisomerase II medium subunit	vogp0860	INFTYYALYAVIHNZAEEYTIYAIENHTFPDVVDSLKPVEYFVSYGTFKKPRSNNENVHNVA NVIDGVAKLGYHHGDATAFEASCQMANASTDNIPFSERQGNFGPSTVQKACSSHYILARV HKNFYKIYNDMRDAPKYNDKEHIPPRFYLPCVPMVLVNSCSGMAAGYATNIFAYAFKSLD NDIYDAFASESVAEPWVEFPKFSGKVIDFNDCNGHZIVEGSYEFYCKAEZNISEIPCEFDSEN DLSEFEDSFE
34	[S] Tail completion and sheath stabilizer protein	vogp0893	MILSNQANITNFRLDIPDTSGTKAFKMNVQGASIPGINIPPTEIPIGPMGLASNNIPCTTIEFDP LIIRFLLDEDLDAYFEVYKWMLSINNYANRNSTTRIDSTVPHAISLHVLDNSKNKIVSTFNFH DAWPSNLGEVEFTYTEESNTSIVFTVNFLYKYFDIEKEGQVIRPFLASYAGSFPKLHPFYSQP PA
34	[U] Uncharacterized	vogp0883	MKTLIKTFFGSHLYGLSTPDSDLDYKGIFIPTAHDILLGGLKDYNSKDTTSGKGKNTVDDVD TEMISLQQFLHLASKGETMSMDMLHTPQZLMTSYHLNIWEYIQQNRSMFYTTNMEAFLGY FRKQATKYGVKGTHLAALEQVMDVIEAFPQDLDDYQEDCSIKEGGSRLSYFEGPLPTDEFC FPAADNYRTGESVSFYKVLGRKYLYGIKFTEFEHHINKMWEEYGDHTNKAKEEVGVDWK ALSHAIHGGZQLL
34	[U] Uncharacterized	vogp0892	FMMAFKEFSQGTYVAVKFSEDTLDALEDLQQTLRIPNPVPRDKLHSTIFYSRVYIPYIPSSNS NFLASSGHLEVWKTQDGTHALVLVLDSHYLRCRHKDASSLGNTHDYPDYTPHITLSYDVG PSSFSGNSNHFQVILSHEYMEPLNLDWAADLK
34	[L] DNA primase subunit	vogp0859	MMYFDNYZAHQTAIHWPRFSKHYHTSSFKFNFCCPICGDSQKNKNKARFWIYKYKDHFRN RCFNCNYHQPLCSYLKEYKQDLYRECLMERRKEKDHHKHPKVKPNNVEKMEKTEMKKV QNLPFCNNLDTLPEDHPIIKYISDRCIPKDKWNLLCFTQEWQKLANHIKPDTYKNEHPNDRL VIPIFNQDSKIQSIQGRALSKDHSNHQYMTIKTHEQASKIYGLERIDSYNPVCFLEGPIDSLFI HNCCAITGGSL
34	[L] Tk thymidine kinase	vogp0887	MAQLYFNYASMNAGKSTSLLQAAYNYKERDMSVLILKPAIDNRDSEGKVFSRIGLYHQAI TFTEDEDLFDLVZRESNPDSVLVDEAQFLTKEQVLRLSTLVDNSNIPVLCYGLRTDFKGNLF QGSPALMAFADKLVEMKGICHCGSNATMVIRVDKEGKVIRDGAQIZVGAENYVSFCRKHF MDGTTNZCNKF
34	[L] DenV endonuclease V, N-glycosylase UV repair enzyme	vogp0865	MTRINLTPVSKLSNQHLMAEYQEIPRLFSTVSKHIRNSNRVHNTFKISPNFILGTGHVTFFYN KLDCLRNRYFELMDELLKRGFNIKDNRTVQIFSHIHHDFLGNYIPNZASISFSHARLDKKIPQ KPTWSKYYGEEVSAIN
34	[R] Sigma factor	vogp0875	PKTNYVNNKDLYQDICKWKQEMNENPDHHIPMPDSIGIAIMKISKGLSRHCNFSGYTQTW KEDMIADAIEASIKGLNNFDKHKYKNPHAYITQACFNAFIQRIKKENKETATKYKYFLHNV YDYHDZDMAQMADEAFIQDIZDKINHYEASINTKNKPKKEZVILDYLYZDEPDSEID
34	[L] Sliding clamp, DNA polymerase accessory protein	vogp0872	MKLSKETIAILKNFTTINPCIVLNPGNFIITKAINNITYAEATINDEIDFELAIYDLNYFLSILDL VGNNEITLHNKSDSIINIPNSHSKVNWPNAKATTIVSPKEPFIPPPYNNFELKAEDLQQILKIS RTLGMDNMAITNKDGHIVINSFTKVEDNNLNPKFSLNLGDYDGTTNFNFIINMDNMKMVT SNYKVYICSIAATRFQGFTSSYIIALEAYSN
34	[L] Clamp loader subunit, DNA polymerase accessory protein	vogp0871	TVITINPKEFMWEQKYRPNSIYECILPEADKKIFKDMVNKGHIPHILLRSPSPGTGKTTVARA LCDDIDAEVLFVNGSNSKNDFVRNDLTPFASSISFYDNGKVIIIDEVDHNMADSQQHLRSLM EAYSNNCSFIMTCNNLEGIIPPLQSRFHQIKFGYPTNDKLRMMKQMIVRCKDICEKEGIPVE DIKVMAALVKQNFPDFRNTINLLDSYANKGKIDEGILSKVTKAPNENEVVEALKSKDFENF RSLAPK
34	[L] Clamp loader subunit, DNA polymerase accessory protein	vogp0870	LGLFLFKDEDQLNEHEIAWHSKDWEAVTELANTFKEKAFDFMDHITZNKHHVNVDNNSD YHQCLINNALSQHTDSLYSAYIMNLMGCSLPDQMHYNYLZHSIPKGKRNGKWAKLTEDL DQNLVMKVVTKAYNKFLKHFKPLATNELLEGVTKNKTEHKQLKKIV
34	[R] RegA translational repressor protein	vogp0869	MVKMIEIKLNNPEDFLKIKETLTRMGIANNKDKILYQSCHILQKQGKYYIVHFKELLRLDGR RVEITREDNQRRNNIAKLLEDWGLCDIVHSHRDDLTGZNNNRVISFKQKDDWTLIHKYKIG N
34	[S] Major head protein	vogp0866	KTKKDLSKKWKPLLEAEGMPEIATTTKQDIMAKILENQDKDINNDPNYKDHKMVQAFDA CLTEADVSGDHGYDPTNIAPGQTSGAITNIGPTVMGMVRRAIPQLIAFDICGVQPMTSPTCQ VFTLRSVYGKDPLAASAEEAFHPTYRPDASFSCQAAAAZATTISLSSTATTCATTDGTLYKAF VSASGSAYSMHFFWASVAVTSAVASEKTDTEYKKWMAEGZLVEIASGMATSLAELQEGF NGSSNN
34	[U] Uncharacterized	vogp0890	TSKLDIHSGTPYPSCALSNLAHHPFVFDGIHCGCMEGFLQSLKFKNLKKQSQISALSGTAAK FRGSKKNWYHDNTLYWQGVPFSRHSDAYQNLLDNAYDEMAIQNEDFRKALSASKNTVLS HSMGRTKPFDTILTVQEFFSRLNRLRYYLLVAVKGCN
35	[L] DenA endonuclease II	vogp0932	KEILEKYCFIKISQLYLEEDDSINPTFTKNHKHFIYAFIVNDKLVYIGQTKNLHKRIDSYCNSK NWKNPTPSHRNSKLLEEALDEGNTVTFZIKQCFKFFITTPSGSTIITTMDLEEPKFIKNFNPPW NTQYKNKQKQQKSFSIN
35	[L] DNA end protector protein	vogp0896	IINKDAHQVHKAAQKRNKKKWIQMGLEYKKAKAKGMTAKDFAEAKGIKYSTFTCAMSR YTSYIKTAHKVKNLKNNKLNKQERRLAMINSFRSSMRTNTRNEGTTJNNKSQKWFTKTIKE AIKGHKVFKPQPGHIYTFTYDAKHKDTLPYWDNFPLIIYLGLGKHNSGNSVMZGLNLHYIP PKARQQFLEELLKQYASTPTVTNKTNLKINWSYQKGFRGSYQMIKAYLPGHIMGSMVEIAP KDWANVILMPAYQN
35	[A] Inh inhibitor of prohead protease	vogp0916	IDKEZFEELKAEANSDYZEATKEAKAKLDKYTKQYGIKVPKTKSFVNIMJYLEEALSDSASE AITETSNGSSDIDLKMAADITDGMDFPNEEVTYEVNLIIDPHEDKPIITEVVANYMKHHTIEV SIEVIQVIKAEVVDILAPIGDVZAHSISHNLDNQVFTFPKDYRPKFKLMGPEGHAYFNFPCWI YHWILEPPNKKHTIHRAYDQDTLYSLIYYIKRNGSVSIRESRNSZFLTFY
35	[U] Uncharacterized	vogp0929	LVQVANICTTFLLLVIVFSGTFWDMKNKVDRLNKEFTZFNKKTKDKAKVFDDLQKQINYIH TKTHKNNELITKLEDENNKLEKDTNQGNVVAQKPKLVEKQINNSFKEFTEDLQEVTKAN
35	[S] hinge connector of long tail fiber, proximal connector	vogp0939	MADQFMAHLGQCYVQTSHLSENNSIKYKLEALGSNANSTINVPFIKLNGVASRLSSYSSGIN SLTISPTSVTSNQATFYFYTPNEHFDTLRSYLISSPSNNIIVLVSYNEMNSTHAFNKFLYQYGS LSWPCIPYLNKVSPNYVAIFNSSLKSICYENFMGNNYTATSDTSAHLEVVFDTIEDIGATGTP QRKVEDIDEYFSRPRSYYTYISYFPTENITTREAIYHFKRDFYNDLVFKVSNGHAPLDLWTT SLN
35	[T] Alt RNA polymerase ADP-ribosylase	vogp0923	MFNMDMKELMIEFFDFDATSPITNLNPKHKPHQLLTIHAGDKDVVFRLCTYTTRGNTLKNF KMGDKTTHILLFKMNNZGNIAQLKNSLGHTPIAIVTSAFNIAMETTHKSQMDAVMFRVPIS KMKGQAQZVQAIVDRLVRTRNAHFKFLQEVDVSNKKFTYVLINRKSMALEDVKGIHNMD NZLFTKVKSNVGEVCVYKDAGEKVTKDEAYAASMVPEEEKDSDPDVLVTAKSSFNDNAP SQGNNTYHZEAHLFQ

Nd Int.	Fonction	VOG	Séquence ancestrale
35	[U] Uncharacterized	vogp0891	MMKAFQILEGSHKGTIIFCLYEDTNDARVIVSKTYKEDSIVEHDIIYCRPARNIEVQPQPTVK IDPVIEGGQHLNVNVLRRETLEDAVKHPEKYPQLTIRVSGYAVRFNSLTPEQQRDVIARTFT ESL
35	[U] Uncharacterized	vogp0928	MMKQFAIILLTWMLVGCSKNPQEVNTDIFHPHNPPPPKPINKFELTWQVNAEDGKPWVTM SFEDSHHFRIWLEDVNRYIQDQKAILCYYRKDLKEEECQPN
35	[U] Uncharacterized	vogp0885	MKFSDFLKSSKDKVDQFIGKCLMSFAYVHSAHFDTNSYAKHKALEDFYZEMPELIDNFTET YMGFTCRQYYPPTLDVPKEITZIAYLQELTQMASEIFNKDLHSSLQSALDDIKALCFZTIYKL TLZA
35	[A] NrdH glutaredoxin	vogp0879	KIEIYGIPKEACHCPGCNSVTKLLDELDIPYTFNPVLTHDGNQLGFNYDRPMIEELANRTSY NSLPFLYPQIFINNKRIGCFQALKDNZDYN
35	[S] Short tail fibers	vogp0766	MMANNTNQHISDKARYVKFNPTNSDFPPSITNVQAALATIYAFAVKGLPDASKATTGIIHM ATKEEVMDGTNNTKAVTPTTLDVNLSYPNASKTVZGFTRYTTKEEALAGANNDTAITPTN LNRALNFTTHIATESTQGTIKISSLAPAKAGSDDTTAMTPLKTQPAIIKLIPSLVPYQESTQGF VQLATIAQVSQGTLREGYAISPYAFTNTPSTQDHAGTIKMATQSQINSSTDDTVVISPTKLAL TSATTSQ
35	[S] Baseplate tail tube initiator	vogp0894	FYCLDFIEHAANZDFQRSDMFSVMFSTTPSTETTYLSDPLGSFSYNNFPYSRKDYFGLTQGM VPSTSTKLLFGGTQSVVRSSGISKHLIGSVSSRAVQALSGQFYVGAYLVDFFNMGYHIPGLT IYSVKMPZNPLIYKMDGDHNPPNVHITGGDGDPLVVSLRTDYZASHYWAMQDWVDSFHD PFTSSRTLPNDVEAYIQVHLHTRNGIPHTIIMFTSCVPVTFSSPQLTYKHNNKIATFDFTFAYR LMQASAIN
36	[S] Baseplate hub subunit/tail length determinator	vogp0610	MNNTLISFHNPQKEADDTTTNTLDALNNIYSKLDNFQAASQLITQTVEEKGNKVIGSIKDNT TSTHNTAEGTQLIAETTQKPNKVLQEINKFSSQISSKLDSLATLLERNFRTSTIMQTTSGTSSA AIKATIPVKVEKPEPSKPLFCSLPTPPKVNKQPNKDNYQASHQEDPNQEPNNKDAINIKMSH TMDKLTKTVHSGFKKTICISNHISCMLFKYTITATIEAAKMAALILGIVIGIDFLMVHFSHWS HLFQ
36	[S] Baseplate hub subunit	vogp0920	MLQRPGYPNFSIKLYQDYEAWLDNRFIELAATFITLTMRDSLYGVNEGLLQFYDTKNMHT KLNGHQIIQISLSTANNKKVYNRIYGIKHFCVSVDSKGDNIITFQLGSIHYIKNLKFSRPFTNN AVDSIQEMMDFIYKDRALLTPPIKAINAYVPNIPWVSTINDYLDFVRELGQAVESEHFLFVW EDIEGINITDYDTMLNKEPIPTIVSEPRLMGQFIHVFDHPVVFDFEWLTKANPYTRNPFKNVT FYALSN
36	[S] Baseplate hub assembly catalyst	vogp0919	TNIIRIKLPEGVQRFKPFTVKDYRDFLLVRNEIKSNPQEQQEILDELLEEIYPEYPPTWHQYIF LKVFTCSMGKTKVPFTFTCPKCNKEHNIPLKLHQEALKAPELEVASIKINFNFPKKFNDZAK TFSETIKEISDGNNQYHWTDLPEEVQDQVIDAISFEELEEVFESMHPIKITQKISCCEHHDLN YTDMLELFKLLLNPDEIFTFYQINHSLVKSNYSLDYIMDMMPIERSIALTLVEKDLKEANN
36	[S] Baseplate hub subunit	vogp0918	YHFKIHLGYKVVNSRTFTFKEYLELVTSKDTSSLEEVVNKIIKDCSSELNKQEAEZVLVKLW AHSLGKVNQENTWNCNSNENPFFMNFSHTQIAZEEAFSYHLGGVKIKFRHPKMFEDKNITQ MVISCMESIHINGQSIPMEDLNDQELDHLFNMLTKDNIIEIMDKLSSNKIYLAIPIKCPEGNSR AHIIRGLKAFLEFLGV
36	[T] RegB site-specific RNA endonuclease	vogp0888	IRRYKLRRLLETEFREINTQIKEACKTYGCCHYFHLKYSHHLMDRSIHRKIDENYVLELFNKJ KDHLPEINEFLSMPHPPNMDZDFIDGVQYRPGRLEITDGNLWMGLTVCKINQDCKYPSNRC RMAIINSHRLPGKASTAVIKVQGL
36	[L] dCTPase	vogp0864	TPQFNECSQLINDMDKAAKASLDGSIPEDKDPLQVMLDMQSCLQVHLAKHKPEYNKDPNK LETCGEILDWLQNQEDYIADEVRELYTSLGGMSNGNEASSIWKPWKAQHGEYRDRRFSEL SPEDQLEVKFELIDIMHFVLNKFIALGMDAKEIFKLYYLKNAENFARQERGY
36	[S] Hoc head outer capsid protein	vogp0795	TAFAVTITPZTPTGVIGGAKNFAAAPAGGKTADGTITYAWTVDNIAQDGAAAAFAYALAA PAGAKNIKVVAANTLSAGDPEITEATTTIAVNNKNNTTAIAVTPASPATVEIGAPIKFTAAFS SQPYGASITYAWZVDGFHVDGQNHSTFZYAPATZGTECIACAVNVTAANYVDZSVKSSPV SLTVNKKAITSTFNITPHSPTTHZGVPITFTANVSGAPZGATTSYHWYMDCSNILDSTSAASK FAPTVVCSN
36	[S] Baseplate hub distal subunit	vogp0921	MKPNLNFIFTIEILDINSKEIKIPKMGFKQHNLLKDVQGPDENLKJLLDSICPSLTPAEADLVIL HLLAFNSKIQADKEVDGFTFNMDEVYICPDSEFHFQGKTFYFKTPTNTYHFITNSDILSKRFY HVNTGKEINFREIPAFILDWANDIFTTMALDTPKGTIYGSTNIMGSLZ
36	[S] Prohead core protein	vogp0618	MEELIEAIKSSDLVAAQKEFAEAMEANKVHIIKQEKIAIACSILIEGEEPKDEDEEEEDEDED DEEDDKEEEEDEEN
36	[A] Holin lysis mediator	vogp0940	TGRHTAAPKVSCCRSDILIGILDRFFKDTASGKILVSRVIFVILLFLMAFIWYSENEFLALYKE THYETYPEILQAERDRNFETAAQEQLQIVHVSSHADFSSVFSFRPKNLNYFVDLMAYEGKS PSTITEKNMGGFPINKTSDEYTVHLSGRHFSTNKDFAYLPTSKKKNDFDYMYSCPIFNLDNI YAGSISMFWKKKSPINKENLDAICNQAARILGRAH
39	[S] Putative virion protein	vogp0271	NSDLYSTHPKWKECPFCPCNPNLWVZECMDIDVWGYSQCDZRQAIGPWAPSFTPAAQKW NPREWZEAYLSSSQPSNQKAN
4	[U] Uncharacterized	vogp0345	TTTESSIQNQIRLALSKAGRTVFRTNVGKVRTPNGRFFNTGSPKGFPDLSGFRRPDGQLFFIE VKNEKGHLRQEQKNFMEFMQFIPN
4	[U] Uncharacterized	vogp0056	KLEYDSKSKEYDASGSAYATKVVLNNRDGSYVPFFLPVEKMDLSNTELLNZALEVIYQENF PQRAZNEKFNELDATIKKDEDLNKAQVTLMNIIDKFZEKKVSTDEDLNNMDZN
4	[U] Uncharacterized	vogp1127	KKLFNWIWSKHNETEENSKWTFENNAWEPSPHRYNQDHGLADNLI
40	[R] Antirepressor	vogp0155	MNMMAVPFHGDSLYVVNHNGEPYVPMKPVVAGMGLAWQSQLAKLRQRFASTITEIVMV AEDGKRRNMVSMPLRKLAGWLQTINPNKVKPEIRDKVIRYQEECDDVLYEYWTKGFVVN PRKMSVMEELNQACADMKRDKNIASVFATGLNEWKQVKAAHVSKIRTLVNEANMLIDFV LADTGKGKITKAD
40	[U] Uncharacterized	vogp0214	MAFKHYDVVRAASPSDLAESLTQKLKEGWQPYGCPLSSAAGYGAALIQAVVAEGDVSTP VVVGGNNAAFSATSDPEYYFVVVLAGQSNGMSYGEGLPLPETYDRPDPRIKQLARRSTVT PGGAACKYNDIIPADHCLHDVQDMSRLNHPKADLSKGQYGTVGQGLHIAKKLLPFIPANA GILLVPCCRGGSAFTTGADGTYSDASGASENSTRWGVDKPLYKDLIGRTKAALKKNPKNV LFAVVWMQGEFDFGGT
40	[U] Uncharacterized	vogp0471	MCCISFRVGTGPCVTSGADRLTKGPLRKSAPVSQPEANPKSRSFSGNFLNHIFRRENPMSEIT SLVTAEAVKEVLRSEEVRSALKQKLRHNLEARLDAEVDAILDELLGVQADPHPEAGDAEA ZSCEFRPHSPVSDATEPN
40	[A] Lysis protein	vogp0479	MYQMEKITTGVSYTTSAVGTGYWFLQLLDRVSPSQWAAIGVLGSLLFGLLTYLTNLYFKIK EDRRKAARGE

Nd Int.	Fonction	VOG	Séquence ancestrale
40	[U] Uncharacterized	vogp1069	MTFLNQLMLYFCTVVCVLYLLSGGYRAMRDFWRRQIDKRAAEKISASQSAGSKPEEPLI
40	[T] Putative single-stranded DNA binding protein	vogp0124	MSNIKKYIIDYDWKASIEIEIDHDVMTEEKLHQINNFWSDSEYRLNKHGSLLNAVLIMLAQ HALLIAISSDLNAYGVVCEFDWNDGNGQEGWPPMDGSEGIRITDIDTSGIFDSDDMTIKAA
41	[U] Uncharacterized	vogp0759	MLTPVFFLDRFTAESGSKTYTNKPDGKSLQAVCCLFPWNNVFADRKVPKITINDNTVTWSN LEQGMGSYIYTFWG
41	[U] Uncharacterized	vogp0466	MRAFYPGNYMSEKIAVVYIGPKPVKKDTITGSRTLFPRLEPVHVDSAMAWQLLGFPDVWV RHEELDDVLKKQQQNEQLRQAQQAQERVLAALAEAENSFVVSVNGQEVDLSKLTSARLA TLCEAEELDIHKDPKETAEAFRIRVREAFRRRVAETEQHGGTE
41	[U] Uncharacterized	vogp0473	MAPVSGERMNDKILWYMQRVVRNSRNPEFMNEVKDACLKKQAFCFEAPDGFLVLRSVLS ADGIPYVLVLLGVCTGSNSVERYLPEVKTLTHLAGGRWAEFHTARRGFIRLGKRLGFERMP DDEDGFMVFRIAV
41	[U] Uncharacterized	vogp0474	MAKTILAPSLSERVYTGTHGNESVAEGVFTVNAAEADSVIHLLSLPVGIRINSLQLVSTGGL GTATVSIKSGEHALIDNSEAVSAKFARYVPVEPYTTQRDGELVTVTIKTAAATGTLNVLLR YTVVGY
41	[U] Uncharacterized	vogp0475	MKKVLIAALISGVSFGAFAQQGGFQGPEAERSTVAQAKELKDDAWVILEGSIVKKVGDER YEFRDNSGTIVTDIDDSIWAGQNVSPKDKVRIEGEIDKDLSSVEVDVKALKLLK
41	[U] Uncharacterized	vogp0480	MATLQELIDLTPEQEKAWNRLVKAVKDFRAAGGKFYSVLDTLSAYNGEHVASIDNDKGY HTASVYMPSIDAPGLTSWADDWHGITLKDGVEVDKD
41	[U] Uncharacterized	vogp0507	MEYGKYETLARAGYSGADRPQGDWQTSAALTRQQYDDWRTRYLPRVARLADLGENNSL MNAQLARVGGLATSSLRTAQMAQDNQMARYGVNRPDNPDSNTLGLRNALAIAGAKNGIR EAEQDRQMNILTGASAPARQKLSVGGQLVAA
41	[U] Uncharacterized	vogp0219	MSGFAQGLLAGFSTVDQAMTRRKELGLREAQLARQQKNNERDFEFAQSQFEHNKNVDQR NFDYRAKVDDRNYALKEREFNANQNYRNASLGMEQQRLQLQKYNQRRLEYNDMIAHSQ PLMEALGKAIEAGDQEAATRLFGQLPKGHPLILMSNEGYAAKAGQAVINLQKIFGDKPDM AIDSLNTPENLDVLSGVFAPELQQRIGMPDSTGDKTIKEARIGSIVPAQQEGYVLIGLDLTYS DGSTAHKPVTEYGSA
41	[U] Uncharacterized	vogp0460	MYGLSIMKPDGSVWISPGFTPQCLINKGTIPATEKSFFKTSIPSGKSCFFFIRTEKKADVMYT HEQIDGYHALRLHVIVRGTNPGVTTVYAFANMVTPPSEYGIAMYNPDGEMIYHGEMMLLD AKLIPVDIKFEKDLGYPCAIMPALVGYYNWKRTPYDRPIYTTSTGATGNKIYSCEHYSGGA TWDIRKPYIDKVLVINTSVYD
41	[U] Uncharacterized	vogp0481	MYRGGLLESEVINMLIRWSEGCRVILVQEFFMPENRRIILDSKESWLIICDSQLGHLMRSMY QGRRFIQLNLEKLKGVHDVALPVKWEFTRRQ
41	[U] Uncharacterized	vogp0138	MVFAKSPARSKTAGKTTCALKESDMAIAASYTMHLYCDCLQCTDGKYKSPDFGEYIGTSW AGCAKEARKDGWRISKDKTRAFAPGHKILRSNKGE
41	[U] Uncharacterized	vogp0340	MPHGFRVFFSICRSGLVTGRGRMSRGWMKWAVIQAEQENDMNILRKLMQSLCGCGKHDD CENGQSLTAQLRLGPADILESDENGIIPEQDRVITQVVILDADKKQIQCVVRPLQILRADGT WENIGGMK
41	[U] Uncharacterized	vogp0255	MDNSWCVVETVTLDAATRTAAGGELMENEGDNIITLVQPKRDEEKLLNITVTGRKNYTQQ SCKHRAIEVHEQDHVILCLQCGCVVDPFQYVLRCANDGEAVVREIRQLHNRHDQLRESVA SLEREEKNTKARLRAARTAILYAENDLKNIEQKVNQ
41	[U] Uncharacterized	vogp0137	MGYGLLDIANQSRREALQGISDADRRREEIEAANKQMAAQQKAQNKQNIGTGIGTGAAIG ASVGGPVGAVAGAVIGGIAGSLF
41	[U] Uncharacterized	vogp0180	MMFTPYRRGTIPAIRIADGTIQAHDDIDEEFFQPVLDGFLISKYTPFDILHALKDGVLQRTG
41	[U] Uncharacterized	vogp0183	MRQKSWIFTKTRKKRLRHSVSGCVRHFAVVLRRLNSMAELSDFLPYVRRHISGPLNIMMT DALSMAAVAFSRQSLVCRREVTVVPVAGKEIVLPYDKDDEECVHIIRISDDNHELFVGRDV DISSGRSLRFACSPGEVSVLYAVAPKAGRSQIPDELLTWPEEVAAGALERLFMQTGVSWSD PLRAQYFSVQFSEGIRRAYRHTLATSPYSSYRNPVRRQRFF
41	[L] Putative small subunit terminase	vogp0185	MAKLDWKKLEQAFRREHAETGITLLDWCRKKKINYNTARTRIKMGKIDHEIDHKTDHEID HDISDEEPCNDAGSGDEKCAKNSEKNCANSAETKRIRGSRLLPPSNAFSQRNTHAVRHRGY AKYLEADNLMDDASDMVLFDELVFTRARALSVTKALKGMFADLEEATDVETRVALYDKI LKAEQALDRNIARIESIERSLLTLDVLAETAPKLRADRERINAARDKLRAETDILTNQRRGV VTPVSDIVSSLHE
41	[A] Host killer protein	vogp0206	MLNTCRLASYVPKGKEKQAMKQQKAMLIALIVICLTVIVTALVTRKDLCEVRIRTGQTEVA VFVDYESEK
41	[U] Uncharacterized	vogp0213	MPGHCAGFFYGVCMAYSEEQRPEAQLGNQNRNSLNIQQPGETDSYEAFFSDPNRWKDNST SFSLGDVLPTMGKGFAQSVRGTGEMARGLGDAMIQSPVKTGARILNEFSRMGLPGVATVQ DIFAGGSRGADEVIDTLPDGKNAVTDTVGKGLKATGKAVSDGAKATDEWLTGKMSPGAV RALNTPMTEGYNDSAVWVAKGVNLIGALVPDMVAGGVARKVGDVTLRKMLTAGLEKK YIAAGMQPERATALAAEAV
41	[U] Uncharacterized	vogp0236	MDFEFTGEETPEQLEKMLEGLGDVDIDSHAQDVVTEDTTEKHADEEAQTQTGDNNVAPTP DASVEQTQDVKEPEAKGVLTRDGKHVIPYEVLEAERSGKQRAEQEAALLRGQIAEEKRRV ELLTSQIHQAGMKPTPLPENEKISDEQIARIREMYPEIGDAVASLIRKNNYLQSRVQQSAQQ AEGNGGEDLSPVLDAMNAVPVLKTWQESDPDRFSVAVSIDGKLQNDPAWKDKTLTERFA EVARRTQVAFGEVS
42	[U] Uncharacterized	vogp1073	MSGATNRHRIPGLTVSAALTGDFSAGVCGDNQKRCTPGFLIFHEMGAISREAACPVRWWK KPDKTTALCKYRSNMVLLCEI
42	[U] Uncharacterized	vogp1053	MLRPTLIANMHRLTKSLWRFTSGFGNKVKPSAPLAFASKEVSNACGMYRLVTVSDSFLGF CANCKEHHPDTKAPPYRYCSGTTSSKEMQESKTMKNRKAKILLVRRNAPGVWQWVRLSN RRMGLMKYYGMMDCGFCKKPSAEQNRWKNHLRTKGE
45	[S] Putative minor head protein	vogp0605	LAQTPZSLQSWKDKVLAPVZTYNHTEARYRHQLEQSVNSMAASYZRSLQHHQGSSLKDN VEANMALYAZALSACSDYSDKFDHTYQSQNYCQGYFYTCAPQVQAGAFQDLZIHNRLKW QCYLRNSSNLPZVVKAZVLDNVHLITSJSQQYQAKVKGEVSRPLVSGSYYKGLQDQLVDIG QVTNNRAAFIARDQCYKATAHFNQARQQELGLHWAIWHPSCAGKEPHPHQVLAGKEZCIF NPQEVGN
45	[U] Uncharacterized	vogp0656	LPTSPIPIDQMIQMPQCVIRKCCAPGPLTSPESTQDSSVQHSQZAYFYQQTDVEHWFGTZSK KAKIAQPYFPRFPNNZHLKFARWLKSDSPTZVYSIPLTGVTLAQZQGYSSSLTITTSSQPICA NIYZAAASSFANSATLIEAASNSQDZLITZYASVYISSVDTNTTCTASAISAVKGNNNZADZS GZZAAEGDSLRASWKSDNTQASPKDGNICLSFNWDTLAKAWTTVIEDNZAWSACNSGQH

Nd Int.	Fonction	VOG	Séquence ancestrale
			YKYSEV
45	[U] Uncharacterized	vogp0722	KSZGHSLLRQYSNYPKSLASLDSFDHWVDQAZFSSYLLDDVWDIYTAQGLRLDLWGNLLG QCRLFQVSQSPNYFGFFCSSRAPNKPWGRAHWYSSQAPWAVSFPLQDDYYRZLLZVKASN NFHTCNCPSINDSRRSMFGDCCKCFVSYDMTLPMYISFPYQFFPTSKKKAIIKSGLLPHQAG TNYIYKTLTY
48	[U] Uncharacterized	vogp0616	MELMYQCLRCGCIFHKRREVVQHLLAGHKHKHLTLDYYYJYFRVRCQ
48	[U] Uncharacterized	vogp0619	NSLTKYTSEKMSHLLPVIYNZZIISHNNTQNIWVTQLSHCSRRSCLMCKKGETELASKZAVK MHVCSCLHVDSQYILHEQGLZTKVLVQPKTSLGFQILGRIDIZYKEENTIYKLKYTHNDKLD CFHGINHLRQLNYYMZMADSLTGRKVIIHPKSCVEEIKHNWAETNLEKRAKAFSIYFZENIL PPNKVQQDZQCPZCPFCHYCWLCNARSLSQY
49	[U] Uncharacterized	vogp0724	MMSALGDVIYVLGILFPALGLVSRNYLVNLMGFVMGAIGFLVFVQGYTDIAFSASTFYLAF LPLSLGLVNLGFFFNWLKEERI
49	[U] Uncharacterized	vogp0725	MRWMSNGZVSPKNIRRNSPRFRFNYSHYCSTIQKVKAQERETNIKPQIHRVFSLLCFNISLT RRALCLFHHLRNLLYGFIH
49	[U] Uncharacterized	vogp0727	MSDGKLVSAWEEELRKAQSLEELKQKYEEVQKQIADCKTLKRLYKVYEKRQFELKZQQF RQLKAELSKRKKVIKKEKVDVRVRVVKKWINSRLFTAEHYVAMLQQSKDGLQLLFLRKA KLVENQGYLMLENRKMRKSWVLNGEPLLLEKSKFPFGKKFVAVHFVLPDYPYTLNLTVD EKJRQLTLKTLNAPQIIHSIVQTKFFEALARVGSGPDYTMLIIGAIMGIGIGVAIGFGIANANL THLLSQHVTNSTVT
49	[U] Uncharacterized	vogp0728	MAKKNGVTELEQLNKENEELKKKLEALANNNNDEDDEEEELQEIENPYTVTNRAITELVZP KDTMFYLSGNQISLILTAFEFSRLPSYFGEEPVVELAQFADKLKHYLVSKGGRGRRDILRVL RVSSGQVRENVNKSLFKQLLQGGDDSDMKEDDN
49	[U] Uncharacterized	vogp0729	MCIPKKTKKHNKTLRLGQKIIKAYKNKGFQSTIIFGKQGTGKTTYALKVSRDVFWYLNNLY TKDKAWQZAZNSLFFZLHEALYKMKDIIHNDNRIPFLIFDDACIWLSKYFWYKDZMMKFY KIYNLIRTRVSAFIFTTPSPKDLAFYLREKGWNLIQVTRNSHETDNTPQALSZMDFSRINFIK GKITAKFKZKALDFFKVQIPNNFYKEYMKRRKDNEKKLSEELQEILYTFNVNDPPTZHN
49	[L] Putative integrase	vogp0730	KGKGNNFYICCNFYIQEIKGKYYVYSMEKDKDGNRHHYIGSLDQIIKYYFYMKVCGTCFNP SQAFPPGTMVGTPGHTPSPDNNVRFKNTDVTTSDKYVTCDNIKAFFEWSMNQHNISKETIK DYIYCIKQPCNNTNNSIKAYRLFYQFLANRNIITPZEVKIKKTNPDFHIPTLEEIKKTLHQVKE YPHNLNLYRLSLESGIRLSEALKVLNDYNNDIWEEGFCIYLLNWTRGQKKSFYIFHITHLKQ IKN
49	[U] Uncharacterized	vogp0732	MFRCPICGFKTIRLFALKQHTRRNHVLTKCPVCNNSYIRLNQHFYSKYDIDHLIYCYLFSTY KLPKNVRLAIKRKLEVEN
49	[U] Uncharacterized	vogp0733	YRRESSSKMLSVLSARFITSVVTGIGVDSGSKKYLSNHKGJIIHVTLEELKRYHSLTPEQKRLI RAIVKTLIHNPQLLDESSYLYKLLASKAVSPYVCPLCLMPFSSSVSLKQHIRYTEHTKVCPV CEKEFRNTDSALDHVCKKHNICVS
49	[U] Uncharacterized	vogp0734	MILIDILLFYGFQFNDYWTTVLGLRVGAQEANPIASLFIKTPZRLAFYKFGLATIALFIVFIFLS FQNTKIFLSITDVVECLVTLNNTLTIRRHKGRSNN
49	[U] Uncharacterized	vogp0735	MKWGSLFSIFFISLFSLNSSASLIGGGGPNNSGAGVYTQTITVDGGTVSTKVNGSTLSTAPW LNPTYVSIYNTYYLQVLPNQEYITNNVSLSLSSSQIALNVTWLLASSSNTGSYGSIAIGYGVN FPSGFTGSYAPASPYASDGIVVYMEKGSFPTYRLFVYFDGVKQLNVSVGSISVGQQIGLGF WYLPASNQLYVYYYNGTLKTFSITPGQILNTINSNYVIDAQNVGPGYGYGQWVIVN
49	[U] Uncharacterized	vogp0736	MTDAISLALQTGLGPVVAVVIILAMMGLTYKMAGKIPAILVGIASTFALMFMDFLPLFWGI AVIFGLIAGLVLGGRDGD
49	[U] Uncharacterized	vogp0737	MGTKLIVYVLLFDVFLSLMVGAYCGITPPSIPPVPSYSFDQALASSIVWTVGWPPITLWGPV TLIPPFSILGANFPGLTLPGFTIPGVTLFSISFSWLAPIFYIAGWIIWIFQTVASVLGYLLSIFTSS VALLSSVPVVGPFLTAFVLIVNFILIWELVKLIRGYGP
49	[U] Uncharacterized	vogp0739	MRAVAVLSAPKKLRRAEATNIGVLLGLFIFILIGIVLLPVIVSQVNNLTSGTAPPVTGTNATL LNLVPLFYILVLIIVPAVVAYKIYKD
49	[U] Uncharacterized	vogp0617	MKARVEYIKLPKSSYPKTYRKIEVTKNDGTIELTLEQTMEVISFKLPPELNAKLERVAFKRK KSKSEIIRVALAKYLENVN
5	[L] Putative DNA packaging protein	vogp0050	SVSSEDFLDHLNLDDKDHKPLMPTYFKAAKAYVQNSICSDTDDTHFYNLEEVHPLFNMAF MAZATYYFNNPVSFTHSTTFPFNTTINHIISQLRCLYSNZZEAQDGQNN
51	[L] DNA stabilization protein	vogp0295	MPIQQLPLTKGLGKDFKNADYIDYLPVNMLATPKEVLNSSGYLRSFPGIDKNQDVKGVSRG VQYNTDQNAFYRVCGGNLYKGEEEVGDVAGSCRVSMAHCSPSQAVCFKGQLVZYRYDG TVKTFSNWPEDZGNQYELGZVIDITRZRGRYAWSQEGTDSWFITDLEDESHPDRYPPQYRA ESQPDGIICIGTWRDFIVCFGSSSIEYFSLTGSTTACPPLYVPQPSLMVQKGIAGTYCKTPFPD YYAFFSHN
51	[L] DNA transfer protein	vogp0298	AKAWKDVIASPQYQALTKEQKAQAQDQYFDEVVAPQAGDNWDQANDPFYAAYPPHLHQ QDNPSLTQQASQRSSCGPYTGQTAKQAGRGLVNIPFDVLQGGASLINAISQALGGPNILEDI YRPVYRPTHPYAQAGZTIGDYLLPIGNATGTTAAGNLAEAFGYSGNMIADSSANPANHLA DFAQQAATNRGVNNGVRALLADIGHDIAQSMSTALGIGVLTSATDISTTPNSGTGCQIIGSQ SAPVANETTQATQ
51	[L] DNA stabilization protein	vogp0296	MADSNLNEPVHQATRLDTSILPRNIFSQSYLLYVIAQGTDVGNVANKANEAGQGAYDAQV HNDEQDFVLADHDERIAATKEAINILEVRLTEAEGNIDVLRNNVHGVESDVSDIKSDYVSK KNTDSQSLSSPLAVYTSYSVDGIQVVGARQTGWTAATGTPFZGSFNANRSFTVGTTYTQSE VPALATGLVAARQRILALEDALRSHGLIDIDN
51	[L] DNA stabilization protein	vogp0294	TKILTKGEMVLFALRKLGIASKATLTDVEPQSMQEGVNDLEDMMAEWQIDPRDIGYLFSTE EEHPTPDDDSGLPCKYKQAVGYQLALRMSSDYSLEPTPQIMATAQYSZESLLTHTSVFPSM RPPYNMPIGQGNNYAILNSGRYYPGDRPSVYSTTPNSGN
51	[A] Scaffolding protein	vogp0293	MENKLIMDSQVITLSEDQDTPPDDTIHTDSQDDNQAQEDDFEDVSNZPENTAEQHQDYTSQ IGHQQIQLNPQEEYHMQGQPAPRGVKDLCKGFKDAQKENPDLNPQLEEAGAEQSDHHQPH HPDAIPPEPTLGSWNYQEEAFEQASAHWHENKGLIQQQQQHKQAQRQEYQEPFKQHIQAH QQQAAELPLKDHPDMEAVILSZFPAMQQEIMIHSADEGAKSLAZSSGEGQPSCQHLTPZSDP ITSTFLLGQISK

Nd Int.	Fonction	VOG	Séquence ancestrale
51	[A] Portal protein	vogp0292	AETLEKRQEPIMSRFDPACSPQEEVRQKCMEATFFSRIPQWEAWLAQHFNKHPKFQINKVS TELNHMVSEYRHNPITVZFRPRDNASQDSANKLKGLFRTDZQQTNGRKACNNAFHERTNC GFGCFRLTTNLVNNDPTYNHQVICLEPIHYPSSPVWLDPNSKKZDKSDSZRAFCMHSLSPN GYEDKYDZDPPTFPSFHNPNSWDYHWLNQDIIHIAKYYEVIKEZVNVFSYQDPITCETVTYY YRDN
51	[S] Tail spike protein	vogp0291	MTNJITNVVISMPSQLFTMARSFKAVANGKIYIGKINTDPINPNNQIQVYIENEDGSHVPISQP IIINAAGYPVYNGQMAKFVTVQGHSMAIYDAYGSQQFYFPNVLKYDPNQFCAKADQQLA YTHKLYDYPTQNTTASATVFILQNDRNYILDYQSDKDNZDVQLLTLECSDKYISDNYQGIP NSGSGTFTFPTSSKCTMTSYTITLSTIHCPRLKVPSAVNVNIKGNTHQCPHINFFNCPIFLWFN NWEPAPN
51	[L] DNA transfer protein	vogp0297	GEGGSSYNGAKEAARAPQYAADLQNQQFNRVMENLAPYAAVGZPALGQLQNLSTLEGQG QALNQYYNSQQYKDLADQARYQSLNAAEATGGLGSTATSNQLATIAPTLGQNWLSGQMQ NYGNLVNVGLGAAPGQASAGQNYANNVGHLSQQTAAIGSPGSDQPSTLRSAISGGTSGAL AGAGLAAMLGSSSTWGAGIGAGMGLLGSLF
51	[L] DNA transfer protein	vogp0031	TGTRRGSNGGLLAGVSFQNSNSPSVQDINHTLRLIRQNNDFDRSGANNVGLTASQGLAGIA HVFHQEKQQQPQKDFQQAYAKAZATGDPYALRQLATENPDQIEPVHQGMGFIDEDQRNA MGNLATGANLASSQGPEAFPSWLPNNNDNLDHVGVNPNQVIQTYQQNPZGFAQLIDPMG MGALGPKKYFNVQDKMAGREIDRGRLAETIRSNKAGEGLQARGQNITMRGQDMSAPTAP RGQDLATQHSKARAIYR
52	[U] Uncharacterized	vogp0780	MAEIIPLTEEQQFQLDIYKLVLNHNAAAEEAFHFIGTYKLKLELFKIHLLSGRAKCNISARTM KAIRKSKEALDLFTTGA
55	[L] DNA polymerase	vogp0141	KMSRTSFIKDDNIKVFTFDHKSNDLYGIFZTFSLFPFLZKYKDHRYYYRDKTFFYRKNYDY NILVDYLRFNYNYLZEYKTNNFSNYLFIINNFDSFFILNFFLHHKFHFELYILEHFYYNFKTFY IKKTIREIRMRKDNYNLYYLKNYPLVPGMTMDKFVTMFNKAQAIFNNYVVYZYZGLSNNS LVFKDFNDFEIHYFITNRRPNSIFHFVNKDDYKYYNNFTFSPIPRNNCRYKYDZIYEDFNTFD FNHZYDG
56	[A] Encapsidation protein	vogp0149	EATEKCYNPQKMLSYDQRRNFIISGRGIRETFALKRDSFKRFIKKGEQFIYVRCDKSELDHIN RYFNNMDKZFTAQLFELVCRNFEVIHKDACZILPRIIGHADKFEDDENIFVVSSTKFVZNSKI FGYLNNKFSAWVNLMDTLYNKRMSICYDZFFIYIAAIDAYFDNFDVDPDFNVIFTIFRHLIR WITYVLRDTSNFNNPYFGFFKFDDDNSKRFFRLQLNATFNDFPPDSAFQTKLVKZRGYYKF VRNSCV
65	[S] Minor structural protein	vogp0322	MTEHFITLSTTEPNNNIGIVKLRHADVNSQAIVAQIVENGQPKNFEGLQPFFCLMAQEVTGQ GVSEESVVSFDAKNGTLKYVASDNALQMVGRNEAYFSFRKQEGGRWIEQFSTRTFHYIVE KSIYSQPFKDSNYWWTFKELYRIFNKYIEDGKNSWEQFVEANREILESIDPGGDLLAEVLDI HKIIHDKVPSGFNLVIEHDSQYQPEVKVTSYKNSMGTZANGFDTGPVLGWDRIYNVPVSLS YIRQKIHFQL
68	[U] Uncharacterized	vogp0741	EKQETQQEVYILNQLGVKDISEQNDNKFYKFAVYGKFGSGKTTFSTQDNDALVLDINEDGT TVTEEGSVVQIENYQHFSSVVNMLPQILQQLRENGQQIDVVVIETIQKLRDITLDDVMDGKS KKPTFNDWGEGADRIVSMYRYICKLQEHYQFHLAISGHEGINKDKDDEGSTINPTITIEAQD QIKKAVISQSDVLARATIEELEQDGEKNYRYVLNAEPSNSFETKIRHSSYINTNNKRFVNPSI NDVVQAN
68	[U] Uncharacterized	vogp0758	MQQQAYINATIDIRJPTEVEYQHFDDVDKEKEALADYLYDNPDELLKYDNLKIRDVDVEVE
68	[U] Uncharacterized	vogp0311	MWITMTIVLAILLLVCISINSHHARDIQALRYMNDYLLEQVVKTKGYKGLEEYRIELKRINK DIKK
68	[U] Uncharacterized	vogp1139	MTNTLTIDQLQELLQIQKEFDDRIPTLNLGDSKISZIVEFFEWFNTLETFKNWKKQPGKPLD VQLDELADZLAFCLPLTYQVGVYZEZVEEAMESSFZNYKLLNLFNLQYKZFAQDALVSTH QIILEEFYPDLLALIMPFYFAYNZYTIDQLIDAYKKKMKRNHERQDGTADAGKGYV
68	[U] Uncharacterized	vogp0757	MPKEKYYLYREDGTEDIKVIKYKDNVNEVYSLTGAHFSDEKKIMTDSDLKRFKGAHGLLY EQELGLQATIFDI
68	[T] Single strand DNA binding protein	vogp0756	MKITGRAQFTQETNQENFYKGCACLRAGEFTVKVDNVEFNDRENRYFTIVFENDEGKQYK HNQFVPPYQQDFQEKQYIELVSRLGIKLNLPSLDFDTDHLIGKFCPLVLKSKFNEEQGKYFA HLSFVKPCNKGDEVVNKPVPKTDEQKAKEHQGARQQTSMSQQSNPFDSSNNFGYDDHDL PF
68	[U] Uncharacterized	vogp0754	MPKIIVPPTPENTYRGEEKFVKKLYATPTQIHQLFGVCRSTVYNWLKYYREDNLGVENLYI DYSATGTLINISKLEEYLIRKHKKWY
68	[U] Uncharacterized	vogp0747	INSNKLNDRIHFCHSENNDPNPMKQENKMLYSCFACIQDSZVSDTQTNMTTGSKFIMTIIIR DTQGDYLPTNKHYVZIEGRZYNKRFNIKZVNPDYQDKAYITIYGEDVI
68	[U] Uncharacterized	vogp0704	MTFTLSDEQYKNLCTKSNKLLDKLHKALKEREEYKKQRDELIVDIAKLRECNKELEKKAS NWDRYCKSVQKDLINELGNDDERFKFGMESNNKIFMEDDADENN
68	[U] Uncharacterized	vogp0444	LGWSKNHEQEWRLTRLEENDKTMLYNLNHIKZCQKTQEQVYIKLDNTLEDLQKDKEKDE KNKKENDKNIRDMKMWVLGLVGTIFCSLIIALLRTLLGI
68	[U] Uncharacterized	vogp0443	MTQYLVTTFKDSTGRKHTHITKAKSNQRFTVVEAESKEEAKEKYEKQVKRDAVIKVGQLF ENIRECGKN
68	[U] Uncharacterized	vogp0434	MYYKEDILNMIDNHKRKSNILDSKVPDYESTSIAQYGFQSTLPKAQGNTSYKVZVKVINZN KAZNKYDHLIKKIDFINQYQQYLANKDYRFLQMLKQNESLNNNMSISDLSRNNFZSRLKDL VNNLYZLQQN
68	[R] Similar to bacteriophage phi-11 int gene activator	vogp0310	MGCLVVVKEILRLLFLLAMYELGKYVTEQVYIMMTANDDVEAPSDFAKLSDQSYLN
68	[U] Uncharacterized	vogp0309	MSKTYKSYLVAVLCFTVLAIVLMPFLYFTTAWSIAGFASIATFIFYKEYFYEE
68	[U] Uncharacterized	vogp0286	MNNREQIEQSVISASAYNGNDTEGLLKEIEDVYKKAQAFDEILEGMTNAIQHSVKEGIELDE AVGIMAGQVVYKYEEEQENDN
68	[U] Uncharacterized	vogp0283	MYYEIGEISRKIIHVNGFDFKLSVMKGHKGISIQVKDMNDVPIKRFYVVDENDLYTASDFLN QAIYEWIEENTDEQDRLINLVMKW
68	[U] Uncharacterized	vogp0279	MNILYNVQGVLSQDPKLKNHNSNIFYYKESHNAKDSKPPFIITPVFDZPSPSTYNKCZZWD YSFLIQIESSNNHKNKYITNIIRYZLYQHNQKQSCSQLETCYEYNSNPEZSQDYQGFPYYIFY KGLFFKEN
68	[U] Uncharacterized	vogp0167	MTDSARKEYLNQFFGSKRYLYQDNERVAHIHVVNGTYYFHGHIVPGWQGVKKTFDTAEE LEIYIEQRGLEYEEQKQLTLFNN

Nd Int.	Fonction	VOG	Séquence ancestrale
68	[U] Uncharacterized	vogp0748	MSVKVKGNQDLENKLEKLFGIQDMVEVQDKALIEGSKVFIKDFKKQLKPSKDTGASIKEIT YSKPNGIYRERTISIHWVGPMDRYNIIHLIEHGHIQKNKLFNPKALGGINKTLAQGQNKYFE TLKKELNKL
69	[U] Uncharacterized	vogp0750	MLDTIQIDNKTIPWLVVQRGFEIPSFNFVSEKENVEGRSGSIFKNRRLKYFYFDLPLIVRNDZ LSPGGEKTHDDVLEDLVKFFNFEDQTNLQFKSQNWYWFAYFEGPLKLPKNPPGSVKFTIKV VLTDPYKYSVTGNKNTAISDQVSVVNSGTADTPLIVEARAIKPSSYFMITKNDEDYFMVGD DEVTKEVKDYMPPVYHSEFRDFKGWTKMITEDIPSNDLGGKVGGDFVISNLGEGYKATNF PDAKGWVGAG
69	[U] Uncharacterized	vogp0749	LDNLQGYKLRLEZDHESKGTNLQDYSNHQFNYLZITZQDEYZNKQEEYENYTAKHMEVSE ELLDLVVKAYDNQFTSKDLQEGLPNPQGGIDZIGQFICITQGRQTZDTNKFIQNNQK
69	[U] Uncharacterized	vogp0231	MAKEIINNTERFILVQIDKEGTERVVYQDFTGSFTTSEMVNHAQDFKSEENAKKIAETLNLL YQLTNKKQRVKVVKEVVDRTDLSSDETVDSETM
7	[S] Putative tail component protein	vogp0259	MACFEEALQGILNQADSMCTNLTVEDKAKITKAGAKVFNKDLAKVTHDKHYSNRKACKD RHLADCFFVQNTNIDGFKDGISTVGWENTNZQGTRNTHIINKGTRFPNYPNQTGSNZNKAG PETYHTDHFITKAQNNNQKAVLZAEAEAYQKIINKKGDENN
7	[S] Putative head-tail joining protein	vogp0247	AVGRYKPSDFSYKAEFGTYQTTPNKFTGKVSNPNFVTQFTFRFTPHTRTLTQEYQAMGTGL DDTRVIVIRHNSZVLEGQVVTFNGTQYDIVHISPDENFSFIHYDFLTLKKRKKVGN
7	[S] Major tail protein	vogp0087	MATKGLKMVNLALVDPETGDIIKGHTGLSTNGVFPINSKMLGTNTANITNLASTPTKIYGN NSLVDANIAKGTPSVAFAFNNLPFDIKNKLLGRVNDNKGGYTQGHDMPQVAVLIQTTTMD PTNPLYFAFANGNMNETTMNNNTNTNTAQTRLDDTLTYNAFSVSPWGNPAVKFFYTGDS NFNKAIMLKEVFGGYTRASISKVST
73	[U] Uncharacterized	vogp1144	MQISVNKNNEVICYAITGGLQECLDIENLPNNFYQEFHPKDFKYSNGEIKFNZDYZKEEDNQ TYSQQQNSVSSDEELRSMVASMQKQVAQSTKLSMQVDQQNALMAQQLANLNNKZEZAK GETKN
78	[U] Uncharacterized	vogp0665	MPSLCHTZPSLADISTRTTHNGINPHIVEQLNKTNDILDDTTFIQCNGCSKHKTTIRSGLPKPT WRKYNQGVQPNKSHTVQVKDSTGMLETZAZVDKALADFYSNSAAWRLSEDMAZLQGLN HNLATSSFYGNSSIEPENFMGLTPRFNSLSTYKGHSNFYAGGSGSNNTSIWLTFWGHNTSHT IYPKGSLAGLQPQDLGEDSLFDATGGHYQGYRTHYKWDIGLTLRDWRYIFRICNIDVSNLT KNASTGADL
78	[U] Uncharacterized	vogp0669	MHSTNMKTTAGSLLCWECNSDNNPQYPWGLCINLNTEALQRLGLNTLPAVGSFVZITAMA IVRSVSSRNNGQPNNYVDLQITDMDLHPNDESZESQDSYCPSCKSA
78	[U] Uncharacterized	vogp0664	MMLDQQSQVSDGQAITASAASTNVIDSGQSKPAGKDGHSHLVIQVDESSNTSSSSTVTFQT QDYSYYSSSTDVTSTGTPWESYSAACKQPVIPLPTKLDRYCRLYYTVTTSPPTTGKFSARVV TGIQHNVTN
78	[U] Uncharacterized	vogp0668	MSNYDHEDLRSQEDLNNHZSEHTEEADFKWLMSSKQGRRIIWSLLEKARVYYCSCFYTNPL ATAFTEGQGNZGLVLLDQIQAQCPEQYSNMVKEHYQQDN
78	[S] Tail protein	vogp0660	QCYPAITZAGTSNTZSGTSDSIRYHQAKSSSATGDLNAYVAKSGAQYSLEAGQHQAADLTF KASQSKGTQAATLAANGTDLGKGCASEVLVYSDLFRHLDTYTSLTNAHRKAWGYQAQGT ESQSQACSQAZNNQAGALGTZLTTPSRSZGAYHLGRCSWSPCTHPKGAPFSSSLCSPSGLTD
78	[S] Tail-head connector	vogp0670	AEQTEKELLLYHWGQLNTERNSWKSHWKDLSDYLZPRACRFLIQDLNRGDKRHNNILDPT GSMAQRILASGMMAGMTSPARPWFNLTTSIPDLEESAPVKAWLDDVHRFMZKIFTKSNSY QSLPSMYEYLGTFGTASSVVLQDFQAVIYPQSLPTGYYSLATYPQGSVNTWFRQFHITVPQ MVQEFGZDKCSTSVQGLWDNGTZEHWVKVIHCINPNANRDPCKMDYNNKACKSVYFEPG ADZTNSLRESGYHYF
78	[A] Protease	vogp0666	TTTGTSPTESDPTTYGRPTSHPAAEQPNPGAGKPRQRQQQPNQQQGNQGQRPKPGENKTE GDKEHKHQGAPEQYEFQAPEGLDLDNGVLGHFZHVAKELDLPQDQAQZVLDZKWPNMQ AMLQQQLGESWQNISNHWVASSKADKEFGGDKLQDNLATAQKALDHFGTPELNDZLNYT GLGNHPDLVKVFFKAGZAMSEDKLVTGNNQPRRCDSISSYSPYNMNN
78	[S] Tail fiber protein	vogp0655	STLSTEVVPSPYPCNGLTTHFPFNFKVFNKCELVALLSDSNGIKSNLTLGTDYTVTWNGNYP CGTLTSSSPZATGYQISLSSEVPNTQQKTLTNQGSFYPKVLEHALDNLAMQIQQLAYWLSL ALQDPSSGGNNSCNFLDQLAEHSPPLYAASSHZANIZSEATNZSDINSFSPSLGAPDKITSZPC CSWAASNSPDFGSIAIPPISNPHPPCGYIVLIDZSKPYVYTLAGYSRYIYSIPTQLCYKLQIAN Y
78	[U] Uncharacterized	vogp0661	SDFSCLQPSFGGGEICPYLFGRIDKSKYQSGLAKCHNFIVKHQGPAENRPGSPFVCQAKDPA NKGRLIPFHYSTTQTZALELGPSYLRFLNDGGSLLDSNVPYEMAKPYAEADLFNIHYTQSA DVLTLVHPNYPPKELRRZGPTNWQLATITSNYPPSNPTYVTDTSKDZGSDYTCRNDSKGYS DSSPSZQGTCSYYLZHPGGAYTPAWZTSZSTSIYNVYKEQCSZYSSISQTKSTSLVPYHTDPK SWDGN
78	[U] Uncharacterized	vogp0662	MASEVDICNLALAYLGNZASITYPPEGSKQAEHCSLLYPSCRDSSLEZLTWGFATKCAHLA ATGIPPPEWHFAYHQPSDCINITEILPPGTPNQAGTHNTQPFSCYIEDSSANFIYTNQAKAWS NYMSLVKDTTKFSPLFMQALAWHLASTLARPLSRCDVWATKSNQAYZSLALVSGSHHHK NYQDPQPEGAGFTDGNLPSL
8	[S] Putative head-tail joining protein	vogp0095	IFKNFFKAIWDIFDVLMFILAAITLNITTFLLNFVSCGITLTVTFILAGLVSEFVAKKGN
8	[S] Putative tail component protein	vogp0175	MWPTLEVTQIISGKAFCPSYEVFSNFIPKEQVDNSDKTQVLLTESNPYITNYGNSTFISMIZG VNIQIFYSNDFDINMVSSEIELMKFLKDNDWSIIHSKPHYMDPYTNQITKNFTVQZIMIINN
8	[S] Putative tail component protein	vogp0228	MNLYMPEISKKAFEVFTSNNNIZNMHDYQLALSKIQDNLAYSQAKDQAQGSLSILKETZNF ITDILNLDKKDYDKLLDLDNEHSQEMPYKLVCYLQGZTDEQLDNSSSEEDPZEEN
8	[U] Uncharacterized	vogp0303	MKNKFYANVELGGEITQVSFQAASPSDVIEQIWRTYGISTPIIEIWAELSDEDNNTK
80	[U] Similar to HNH nucleases	vogp0372	MSWAASDRRRRLPEDWALIYRWPVLSAANWFRKINGPGCVSAATEVDHIRRGDDHSRLQ AACHLCHGKKSAAEGVARRRELRARRKRPPQRHPGRRN
80	[L] RPprimase/helicase	vogp0066	MYRGFLAIPYLRWSPERRWSVVSIRYRRLDDGERKYMTMPGDNPRLYNTLALSRHSHDM AITEGEIDAITAQVCGIPTVGVPGAQAWKPHFREPFLGYREVFILADGDEPGMQFAKTVAK TLPNAKIIPMPRNEDVNSLVIEQGKHALLERVSN
80	[S] Major tail subunit	vogp0041	MALNDDAVLTAAVGYVYTAPVGTAAPTPAQLKTIDLTDPDAWTRTGWDSVGHTSRGDLP EFGFDGGDSEVRGSWQKKKLREVTTEDPVDYFTVFLHQFDEQALELYYGPNASATPGVFG VNATGDTNEKAFLVVIVDGDVRLGFHAHKASVRRDDAIQLPTDDFAALPVRATFLDHKDE LSFSWINEDLFNVAEDPEVYLN

Nd Int.	Fonction	VOG	Séquence ancestrale
80	[L] Putative DNA primase	vogp0067	ETTQTLIAKVIQRYHPDWDPPHDSRYDWIKCLCPFHGDETPSASVSYKLKAFNCLACSVRG NAISIIRHQEEVSYPEAIRIAQELSPCGNIPIPRQPSREPCRRVFGDSRSCRSRHRCCGYTVRPC IRGRSTPWSRN
80	[U] Uncharacterized	vogp0061	IGWETNLKAEDMVQELWVWYLESPYJQNTLEDLRHGEALNYLREQVHNILSGSSKARDLF QEGCHYSSDDVKDALRGNSTNRYLVEVLPLAMKDLGSHNEAYAEALRIKYTDGVFPENES AEEALLNGTHNSLNEHINIITITAGIQRDDNGKVIFKDRPCSQHAIFPHIRKVQANWPSDPPD NIAEKLVKHPEN
80	[U] Uncharacterized	vogp0069	PTEPTPKANPLHQQVLGALIDTRPTVWTHKSIDPESPDPKKPLVIETKVHGPEFTALARNVS EHNVDRTAKRWIK
80	[U] Uncharacterized	vogp0076	ATETTNKRKPRSVSQLNQYDRCPYSYKLARIDKVWQRPAAWLPQGTAFHTVSEEYKLCES DGYPMSLEEAQEMFKEEYAKDVSQFTEETPNFDWWFRSGPYNGQCDMERRWHIGLEQVE KFFAWTEAHPQEVIWTPPDGINGIELPFDIELDGILVRGYIDTVLRDDGEVRVRDYKTGNTP GDDFQLGVYSLALAETYGVEAPKTGDYFMAGKKGIKGKPTYPYDLTEWTRERVAEKFRQ LEENJAAZRFDPYPE
80	[L] Proteins containing DnaB-like helicase domain	vogp0351	LQSLYIKGSAGDPLPTVWDALDEKGTHFLRGQLALVCAGPGTGKSAFVLAYALKARVPTL YFSADSDAFTQSSRSLSILSGWSLEKSARAVREGNIGEAAADDFHDIPIRFNYNASPSLDEIE TSLAAYEEVYGDFPALIVIDNITNVRTGSSDEDDPPNGLESLMDYLHEMARZTCSCVVGLH HVTGRYNDGDKAIPLSGIKGQICRVPEMISTLHRVSDGFGSDSLRVSTVKNRGGRSDPSGRD YAYLN
80	[U] Uncharacterized	vogp0530	TLTTPQQLPPLSLEVIEALKATGZTQADIARMYGVTPQAVSWHKHTYGGHLTTRQVVRQE YPFEVPKPLGQSTPFKRLRDHGEYMATGCNGMSEDKLKRLRSFYRMLRDNDLVLEFDPNIP PIPGVSNRGGNZVPPNEZDEDLLIRVNEYTNLTQKGHHIIWRFPSIEPLL
80	[U] Uncharacterized	vogp1122	MKKKPNLDDPEVRSWLZRTEEAPHESAAVLRMHRAGYPGPLIMKTLKLRGTQLMQALNK ALTEEQDAASRGRAIHDALIARGTKN
80	[U] Uncharacterized	vogp0090	TTPAAPERASDNKPPTQPSATATPTATPVKDSKVVPPSKGKVFTTFKGGSHFYAPWIVIQAA SLEEAZAQLDTNFKDLMDKVQKIARPFASSAASQAPNRGSGGSNQSHAPQAAQEAPNGEK RNPHGEMVYKSGVSKKTGKPYHLFSCPAPDRNQQCNAQCPNKK
80	[U] Uncharacterized	vogp0035	MTSEEYAAQQAVISSASASYVLRLATLFANPALSVAEWLRLLELLFPEVQRRYAEAAALAR NFYDSQRALHHPELPPNERLLSELHFEWFVQNMEPARVELSQEDAAQSAVAQLALRAVRE VEMAGRRQIIGAVKDDPAPPNIRGWARVATGRETCAWCLMLISRSNSSAQAPVSCVDTDT GLDLEDDSGDDLETSCEEICEYMEQWHAGCDCKVVPVFKVEDWPGZEAQKRALHLWIDA GKEASRLMDSGKART
80	[U] Uncharacterized	vogp0043	NNELPNGCRFYAEKRGGQQFRGWDEYRYALAAIVNAVRALKYTYLAANSYPEKPKPKAP PEFPTPHRAASKTHKAGSFAPMAVKPIAAARKRKAQTEAN
80	[U] Uncharacterized	vogp0060	MQNILDPKFNGMPGSEMYRAQVFPELFPHQKPMLLDNWSQEDWEMYFGGEFTHGYNRNE TEN
80	[U] Uncharacterized	vogp0030	MGTRGPIRKRSDERVRRNKDEYPTETVPVIGTVNIPQLGLGDPHPMVRDLYNSLTQSAQVN FYQPSDWQYARFALHFLDRLLKSPKPNGQNLTAVNQMLSSLLVSEGDRRRVRLEIERTPSD DAEGMVVDVAQIFKHRLAKASCCC
80	[A] Putative scaffold protein	vogp0036	TSDTPTATTTPTAAHKPQEPPAETFSRDYVQELRQEAAAARVAKKEAVEAAEAPNNDEYE SKLAERDTHYTELETQLGPACLELAKLHTSLDAKVPSDKVLAFVTILQGKDEDSITESAKA AZELVGGFNTKQPPFDFTQGQGFNLPLNGDPILDALKGTLGIKK
80	[U] Uncharacterized	vogp0037	TAZATPZDVTTFWAREPTAEEMALINRRLAQVERMIKRRIPDLAIKASSSPVFRADLIDIEAE AVLRLVRNPEGYLSETDGNYTYMLQAHLSHGKLEILPEEWEILGINRRRSNAVMVPNLVMP T
80	[U] Uncharacterized	vogp0038	MSLLDSGAGYQPFIVYPEEMVIDYDGNTRTRPSKTGIPAMARFQVQGQSGTSARRAEQDNE GFETEKVYRMRFPRSFTTEHGVLNQIEWRGQRWALFGDANFYDSSPRMARVDYTVKRY
80	[U] Uncharacterized	vogp0039	LSKGHTDTNKVISHLVGVNDAVHAEAKEVTRRAKANLAHAHTSTHWDKICYHSTTITDAD GYVDSYINLEAPNPLAMEFGHHPSGFFGPNDTKAPQGLYILTCAASFGSLSTN
80	[U] Uncharacterized	vogp0040	MAPMPRVQAVVSPILRTDPPLAGVNVGTWVQDIDFRTFPMINVRRVGGTRHPNRPNLFALP VVEMTAYSPDGLIETEELYEDALEVLYDAVKHRTQTPAGYLHSIMETMGATQFSSSFQDS WRVQGLIRLGVRPPRSTN
80	[U] Uncharacterized	vogp0042	MSNVFTLDSFREEANHKYDPVKVEMSKDITVFLKNFLHLRKNARKDVFQLLEEVDAIPKD DEGKEEAEDDVDESYMEVNVDMVFRMMESVANKZTSASDLVEDFRNDLALTSKVLNAW MEETQLGEASPSPA
80	[U] Uncharacterized	vogp0051	TYEAEDHEFFDILYQQWSQTTGAKDSYWVVEEDEDEHLQWQVLAVDQAANGSKLWLGS FHCEADADFVAGLHGALPDLIRRLHEAIDEAARKDZAHDISQGHLAEAZLENQGLKAQIZE LEGQLSNQTSLN
83	[U] Uncharacterized	vogp1101	SFPPEVKELIWERAHGYCEFCGZYSSTTHHHRPKTSEGCHWEPTNCSTVCCRGIRGCHGWV KANPNSASEEGWQVHPWQDPIEIPFLYRGNWVLLGQEGSINHIPN
83	[U] Uncharacterized	vogp0811	MKRYPSNPITPHGTYFFLQDKKPNMTYRSHDDTMVFHLMGGMALPDKVNSPKSVQLNSN GLQGLIPNMDQKGATQDGVTFIDTLYDPTELDMTVEVCGQTPQHTRQLVRYWVASWDPK KPGKLSFFNPDSGSNWAPVRWPQNPPDKLLRNNFSSQTFTWPFNTHDAFWRSYDYVYQFR FTYEDVTDDFNYLDNGSRGHNWILDYSGDGDGYICATHSPVNCWDHRTGWGTNNIREVV CAPYWDFRTTSDGQLTN
83	[U] Uncharacterized	vogp1087	WEPVPERZATLDDSHHELSYGDPTTAADAAKGSAKEESIISHDWVFTIRYKLWQHTGELGS DFIZVTGTYPRNDVETSTFILKGTCPLVPLFMQCKKTLVRFTVETGGMRWPCYVNTFNYDF KDGSNTGTSQLVGIYDILNYIFVWPTWWLPIQAQPFSHAIFIGPSITCMKTMIAEHSLRLQSG MWELVNNLFSLNPDMQAWFGTILQSNPSYNGSLGDLLDTLKTPIYVVPTNPLLDTSPLVSIT VRMETCGT
83	[U] Uncharacterized	vogp1091	MTYTPSTPVSGRQAQEQATKYFCFSAYAIMDMNGCSAMEILPNPNPGLLDDDPLERZEELQ LHMEKCDREEDLVIPQLTLEDCKNVLPTPTNKGDLKTPYHKDGELVYPPHNVHLAQAILGE ENYERFKAGQNRSNHIRLAWACLDPKFQERQENDPKSGGCCSELEGISEGDSK
83	[U] Uncharacterized	vogp1093	RPLIPDGTHZIFEGYIILIHFDPSASAFIFLVRPQGGIGGGFPAMAKGEPGIPPNFDTIVNFTLLD HDDPTPAZASFTEITPHCTCTPCGYHLNFSLHSGPKGQDGNNVWDPTDFAGTPVAGQVPVV NSTADGLVLAAQRVGDMYIPASJSNTPSCGNTTYTLAHVSIPAQPFHTHWRPRVWGYTLVT GKGDIRVDLVARLNGPTCGNIVGRCQGIASSKPPILVSSSASNPATSSSNDSYSVTASTPATI YFRC
83	[U] Uncharacterized	vogp1094	EKHFLWGSLGTGLLGGLALTGLLSWTFATGPPALDFFLKHDTSFY

Nd Int.	Fonction	VOG	Séquence ancestrale
83	[U] Uncharacterized	vogp0638	MTQPFTGTTRAAWGYTNINNHFNGHTGLIAISIRDYWGTSTNISPQDANSTGTVNWSPLAQ DGQLCKDLFAHKLDSGNWVKNTNPKEGWYLTCTFSEGNSPISHPYITTHNQMIQQSNWPF DSDITKKPZPFTFQALQTLYPSIQLLAANLPLSNANGNPLVELPGMANCFFCQPVDAEYIGH QFSLLCIRKKGECYLYKVEDYTLANLNNKGYPQLGKPGPSTPELTFKPZPYGYFMATVPSN SNPIIKDTFNV
89	[L] Endonuclease	vogp0159	LNSNSSTPKGPLKAGYRSGLEEQISKHLEKQEVLFTYEESKINZLVSKVRTYTPDFFLPNGVI IETKGFLEAADRHKHLLIQKQFPNZDIRLVFSNSKAKLYKGSHSSYADWCNKHGFLYADKT IPEDWLKEPN
90	[U] Uncharacterized	vogp0250	KYHNTRSKCPINSAVRYYGIGNKLFKGLNGHPKGFSHACFVKFNFSSHLSEVHHNHZIPSPK VRAMDPYQNEEPNTEVNYEVTKPNHYMLFDDIEAIKVIARSMTVEQFRGYCLGNILKYRL RAGKKSNTEKDLKKADFYKELFQKHRGZCYDASL
90	[S] Tail tubular protein B	vogp0427	MAZVQRSFNNLMGGISQQPHTLRLPGQGSQQINMRSDPTQGSQKRPGTTFITQLLDNGYPG TAPCLHLINCDGQEQYFFIFQKGGVNVFDZQGSNCMVQSNAZNYYSKDANPCEDLHFITIA DYTFMSNRKMILKTRSNSTHFPSLHLNSDALVYIGGGQYGTPYSITINGKDTATYHTRSSTE AHSSPKKNNTNTKWIADNLDQSLHSTVSNSDNNQNQVGTGSIHITAHNYSNVYHIATKDSY TGQFIVPITH
90	[S] Tail tubular protein A	vogp0428	KAPIETTSELDAFNVMLTNIGQPPIFTMDDTNPQVAMAQPVLNQVNSQVLTQGWIFNIEZG YTLTPHSFSNIHYSSDYLSFTATSRQTQZVNWGNYDRSTKTDRFTNGIKVNLICFRDFDQMP ECFHYYIVTKASCHFNTHFFGPHEIEGVLQEEEQEARHQCSEYQSDYGSFNTLDSDSFTCCF SSCRT
90	[S] Head-to-tail joining protein	vogp0436	MAAIKHTGLMKTTAHALWEKLRNNRGPYZDRAQQFAQSTLPYLLTNHTHGSCSNFQTPW QSVGAQGVNNLASKLALSLFPTHTSFFRLKLTDAEAKELDSHHQDITKVDTALSRVERIAT QHLDQNSYCATLIQVMKQLIVTGNALLYQDYPEGTNNSYPLHSYVVHRDANGNFMEIVSK QQIDFSALPEEFQKAFKGQGAGNNSAETDDVHLYTHIQHDSGNKGDYFQFKQDKDKVLLG ZEGSCPTDSCPYIPL
90	[L] DNA ligase	vogp0454	TYKVEVIIFKTDPFHAIYYNEKAIKAVLEKSSYLMVDLKKDGFRSNICVHSWLSRFGKRFPA LDRLKDCNKRWLQLSNNDEGFVLDCELILEGIDFNKGTGZMASKSPZTKDKEFRLHFVLFN STHKGLZVMYGKFHLPFHTNRILVHFFTVVPLHIITSGMDZDVTQSFRQDQVEYQCSRLQE YFPZMDWMVTETYZVYYMDTMNHLYKNKPEEGQEGLIFKDPMGIYSHSKDSGWWKMKP ZEEAYGIIQATSW
90	[L] DNA maturation protein	vogp0172	MTPTKPQDRFQVLHQVQDTFPFFZFFLGNVLNLAFHTNCLQDDVKDFLQGYHKNFMVQA QRGKAKSTITCIYVVWCITHNPHTNIMMVSGSKEQAEENSILINKIIMHLDLLDYLRPKARN HRTSTISFDVSRALHGNSPSINCIGITAHLQGSRADIMIPDDIETPKNSSTPTDRAKLRHLAQE FNSICNPLNILYLGTPQSNKSIYNDLPEAGYPILIWPGRYPKTNEQACYGHCLAPSLLAHFKD WEEKGHT
91	[L] Exonuclease	vogp0439	MYKLNLVMDTDYLIFQAMSTSEEEVDWGEEIWTLNCDHNKAHHILYHSIKTFKTNINAQL DSMIISANNWRKEVVEPTYKANRNKRKPFGYPHFWQGVMEHWDPNKEPSLEGDDVMGIL ATNPCNDKVILISRDKDFDTIPYCNFFWWTKSNLITQNEDTADZLHLFQTITGDITYGYSGIP GRGKETTKGFFKDPYFFHPVITVLKSGNHKGQEVTYWIYCAPNSKEIHWFQKAN
91	[S] Capsid protein	vogp0201	LKVLWGEVLTAFACASITAHRHTVCSISNGKSAPFPIMGHTKASYLAPCENLDDKPKDIKH TEKVIHIDNVLIYNIQDTMNRYYVPAEYSAQLGESLAMAADGSFSAQMACLCNSPATSNZN IEGLGKPAVLNIGKGSSVNSTNPVAZGKAIVSCLTTAWASLTKNYIPSGDHNFYCTPDNYSP ILSALMPEAANYPALMYPQTGTICDITGFZVIQVPHLTASG
92	[A] Capsid assembly protein	vogp0435	NADIYASFGVNSAVMSCSTITEHEQNMLALDISARDGDDAIESAYHETZAKRDSYDDRDTL FQEDDEDDGHVQIGTSZEGSNAEFTPLGDTPEELVQASQQLGQHEEGFQEMIKQAIKRGLS AETINHIQEEYKEDNELSEESYAELEAVGYSRAFIHSYIRGQEALVNKYVNQVLAYAGGQE RFQAIYSHLEATNPEAAQSLEKAMTNQDLATIKAIINLARESHTKKFGKTPPRSVTNRAN
92	[A] N-acetylmuramoyl-L-alanine amidase	vogp0447	MAKVQFKPRPTTEAIFVHCSATKPSQNIGVREIRQWHKEQGWLDVGYHFIIKRDGTVEAGR HQLAVGSHVKGYNHNSLGVCLVGGIDDKGKFHANFTPAQMQSLRSLLVTLLAKYQGSLL RAHHDLAPSNCPSFDLKRWWEKNELVN
92	[A] Endopeptidase	vogp0276	MLKFLRIALPWVSAGMSFTGGRHLGSNPMNTKRKEEVQNEYIQZVEATDPAQHTVAPISN EYQEDLASFEGSTDRIITDLRSNNKGLSVHIETTTSNPKGNGCSZPNGRAELHYYYAKRIMG IAQEADAQVQALQNTIRELQSKQHN
92	[U] Uncharacterized	vogp0441	MCFKPKVKTPKTDTNQIRAPEPAPLTQPKSVDLGGSZDENTTEATTDSSNVDSNVKSYKRV NSANATAKDTAKAKTSSNAFSFGSKKN
92	[U] Uncharacterized	vogp0438	MLKPINHFLKNPDDIPYIPPAAAEYLQVRFNHAYLMASGHIRSLRADGYSEAYISGFMQGS HSASNIMDEIEVRKEQLREN
92	[S] Host specificity protein B	vogp0437	MGKKVKKAVKKVTKSVKKVVKEVTSPFKGVSGGRAEEVIQQAAPVMVPAPLATAQIVDV PQKDZAYTEDEAQTESARKKARAGGKKALSVARSSGGGINI
92	[R] Single-stranded DNA-binding protein	vogp0458	MAGFTKKIFTSPLGTAEPYSYIPKPDYGNQERGFGNPRGVYKVYLTIPNKDCQPMVDEIVK THEEAZAAAFEEWEANNPQVARGKKPLKPYQGDMPFFDNGDGTTTFKFKCYASFQDKKT KETKHINLVVVDSKGKKIQEVPIIGGGSKLKVKYSLVPYKWNTAVGASVKLQLESVMLVE LATFGGGGEDWAGQEEEDCYATSCYAEGSRDQQEEQEEEHDENPYEEDDF
92	[S] Internal virion protein C	vogp0431	MASKLNSVLGNTATPGTEHLRGVSGTDYQASTIQAQQPCTSLSDSMGHFAKAGSDTYIAK DQQDQKSADERSNEIIRKLTPEQRRQAINNGTLLYQDDPYAMEALRLKTGRNAAYLVEDE VQQKVKEGHFHSNKEMEQYRHSRLQECSKSYAEQFGIDQKDPEYQAGFNSDITQRNISLYG SHDNFLSHRAQKGPVMNSRVELNRFLQDPNMLRCPYSGEFFQNYINNCLVTGSIPSHAQAI HSFSLAFYDASSN
92	[S] Tail fiber protein	vogp0430	MATTIKTVMTYPLDGSNTDFNIPFEYLARKFVLVTLIGVDRKELILNQDYRFATNTTISTTR AWGPADGYTLIEIRRFTSATDRLVDFTDGSILRAYDLNISQVQTLHVAEEARDLTADTIGVN NDGNLDARGRRIVNLADAQDVGDAINLGQIQRWNDSALNSANRAKQZADRATARANDA KDSANDSASSASSSAGSAEZAKRWATYDTVFESDSESSRTYALHSMLYPHETKDYADRSA VSETNSKASEGRP
92	[A] Lysis protein	vogp0426	RLYLDFNNELVKAAPIVGTGVAEVSSQLFCLSLNEWFYVATISYTIVQISAZVFNKMMDWK RDNKEN
92	[L] DNA maturation protein	vogp0425	MSDKTLIKLLETLDTETAQRMLADLRDEERRTPQLYNAISKLLDRHKFQISKLKPDEHILGG LAAALEEYNEMVGDNGLTDDDIHN
92	[S] Internal virion protein D	vogp0368	KDKYDKNVPSDYDGSFQKAADTNGVSYDLLHKVAFTESSFNPKAKSPTGPLGLMQFTKAT AKALGLRVTENRLNPKLAINASAQHSSDLIGKYKGDKLKAALAYNQGKGRLGTPQLEAYY

Nd Int.	Fonction	VOG	Séquence ancestrale
			KGDFASISQEGRNYMRNLLDVANSPPNGQLEAFSRIAPKAKGNSEDSLAGIGHKQKVTPEL PESTGFNVEGGEQQAPHTPFPKDFWEKNGPTLDECDSRSAFFSFNNATSAQSHNSLTGMAF HPGPLDKGF
92	[S] Internal virion protein B	vogp0101	
92	[S] Internal virion protein A	vogp0432	TMTINHNLQTHSDDFTPSHQDVLEAQASGIEPSFTPAYETITVFLDSSLLAVGGNCGNRFWF FTSNQVRSLTRKEHLEFRKLIMEYCNEMLDQYHPFWNNVWVGNRSHIRFLKTMGAVFHK EYTSDGQFRLFTITRCSN
92	[U] Uncharacterized	vogp0422	MSDYLKVLRAIKGCPKTFQSNYVRNNASLVAEAASRGHISCLTTSGRNGGSWEITASGTRF LKRMGGCF
93	[U] Uncharacterized	vogp0407	MSTLRKLNLPQPLKQQSMSCPLSIKKTLPPWKGALIGWFLICVATISGCPSDSNLPEPPKVSV DSSLMVEPNZTSQMLNVLSQ
93	[R] Bacterial RNA polymerase inhibitor	vogp0457	ESHSSTSSLAHHNKKYWSTMEGYKQSFEVPVFAHSLEEATLQEWQYEHAGFVVTRIRPWV KPN
93	[U] Uncharacterized	vogp0456	EDFTSCSEWCPNMWEQTFEDAYLQLYELWKSRCZN
93	[U] Uncharacterized	vogp0455	LRLHYNNSIDNFSVRREDRSIVCASERHAKLPLIGYAVPLAPSVHLFITRGDFEKAMNKKRP LLZAAVTRWPFVRLLLKRIKEVL
93	[U] Uncharacterized	vogp0448	MFKFINNLGQLVVKLYFIEAKKLDKKAKEESHQSIDLAKQSNEKSDSAFSRVPKSAPIATQA QHLSKFFE
93	[U] Uncharacterized	vogp0446	SINHANTIRLPDTADQCTRRVHINVRGEKVTMVYRWKDHKSPKAHTQRMTLDDKQVCRL MGALTKAADNVVRDDRQLLVDLGACVQEIIN
93	[U] Uncharacterized	vogp0440	MAMTKKFKVSFDVTSKMDSETQEILDEKMLDLAKQAGSGEKITPMEQELLVQALTHGPEG SAAFSVRQGFRKAIKDMHZESSYNDSLKLSPATVREVFN
93	[T] RNA polymerase	vogp0406	MLRLLIALLRHRVTWRFLLVLTAALGYACLTDHLGHLEVAFCSILSCCD
93	[U] Uncharacterized	vogp0405	MGTPQSSGLPCIRVPVHNTCTTHSVWPLCSAWSSTMGSRRSZGYQLTPCASTWKTYVSVS WNTVRLASLYSSCLTRTLVMPRSTPNN
93	[U] Uncharacterized	vogp0404	MVTPIRPLSRSLRKRSNGQSRRWSTRVTSVMVCSVRYSVLSFLNTTTPRWKRFVLVV
93	[R] Host recBCD nuclease inhibitor	vogp0442	TYSDSVTIPRDVWNDIQGYIDYLEKDKDSLKNRLKEGDEYFAELEEKZNGAS
96	[S] Putative minor tail protein	vogp0108	TTMDTFCWCTKVQGRGTLDTTFSVRKVQFGNGYTQLASSGLNTNIRTYTFPFSGNPDEVTA IKDFLDRHSGVKSFSWTPPLGDIGLWVFKTNSZGDTLMNSKVIDITATFKQAFTPN
96	[S] Putative minor tail protein	vogp0109	KQYINMSFNADFQKLEPGNLVRLFEVDCTDFGTADFFRFHTHNLPAKSICWQGEEYKAWP FQVEGIEATTYGTSPQPKLTVANLDSSISALCLAZDDLLRAKVTIHDTLAKYLDARNFPHGN PTADPTQEKFKLFYIDDQNTETDSZVVEFILSSPMDLQGRMIPTRQMHSNFSWRYRNKYRT GDSTRYFDKDNNPVSDPSLDQCSGTLTACKLRFGDNNELPFGGFPGTSLIRS
96	[S] Putative tail protein	vogp0110	FMHQKIKKAIMAHASAEYPRECCGLVTQKACVQNYFPCRNLAADPTDHFPLSPEDYAPAE DRGKIITLVHSHPNASTQPSEADNPMCDHTELPWRVVSCPEGHMRNNQPCCESPLMGRPFV LGLFDCYGLVMACYQQARSIZLPDFHLEYHWWKNNSHNLYQDHZQEAGFFEVPCPTNPQ VGDGDLMLIRFKAPIRNHAGIYLGDNQLLHHMHGHLSHRDLYGGZWQDRTITILRHKDFSS NEACRKYNN
96	[S] Putative tail component	vogp0111	TTEDMTVIQLYGILGSRFGRFHHLAVYSTPEAIRALSTQIPGFQEYLTSSRDRGSTFAIFIGSL NLGHHEZENSIGSKEIRIVPIITGSKTGGLFQIILGAAFVAVAFFTTCASLAPWGTTSCSGNSL FTLGASMVLGGIIQMLSPQPGSNSSEQSSNNKPSYSFSGPVNTAAQGYPLPFFYGZMVVGCT VISSGIYAEDQGNG

Tableau E.3 : Nœuds internes ACP associés aux séquences de protéines ancestrales

Annexe F – Publications

F.1 Article 1 : HGT-Simulator : logiciel pour simuler des transferts horizontaux de gènes

Dung Nguyen, Alix Boc et Vladimir Makarenkov (2005).
XII^{es} Rencontre de la Société Française de Classification 2005, Montreal, p. 215-219.

F.2 Article 2 : Étude de la classification des bactériophages

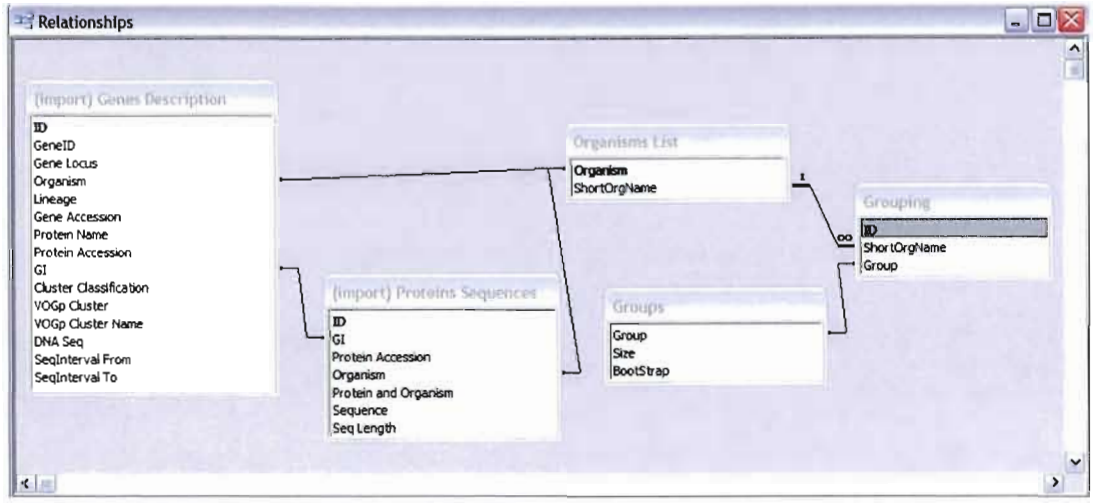
Dung Nguyen, Alix Boc, Abdoulaye Baniré Diallo et Vladimir Makarenkov (2007a).
XIV^{es} Rencontre de la Société Française de Classification 2007, Paris, p. 161-164.

F.3 Article 3 : Étude de la classification des bactériophages

Dung Nguyen, Abdoulaye Baniré Diallo, Alix Boc, Dunarel Badescu, Mathieu, Blanchette et Vladimir Makarenkov (2007b).
Soumis à Mathématiques et Sciences Humaines – 2007.

Annexe G – Modèle relationnel en MS Access 2003

Dans cette annexe, on présente le modèle relationnel utilisé pour concevoir les tables de données dans Microsoft Office Access 2003.



Les tables « *(import) Genes Description* » et « *(import) Proteins Sequences* » comme leurs noms indiquent sont des tables de données relatives aux informations de séquences ainsi que les séquences de protéines elles-mêmes importées de Genbank/NCBI.

Les tables « *Organisms List* », « *Groups* » et « *Grouping* » sont des tables de travail permettant, notamment, de simplifier les références faites sur les noms des phages (qui peuvent être très longs) et le regroupement en 22 ensembles d'espèces après des traitements d'analyse phylogénétique.

Annexe H – Programmes en MS Visual Basic v6.3

Dans cette annexe, on présente le code source des programmes en Visual Basic version 6.3. Les différents programmes (ou modules selon la terminologie de MS-VB) peuvent être regroupés en plusieurs types de traitement tel que détaille le tableau suivant :

Section	Type	Nom du module	Description
Annexe H.1	Importation directe de données de séquences brutes dans MS ACCESS	• Extraction de données de NCBI en XML • Importation de XML dans BD Access	Les utilitaires standard inclus dans MS ACCESS sont utilisés pour importer les données brutes dans les tables de données (voir les spécifications relationnelles en Annexe G) à partir des fichiers XML qui sont des extraits de données de séquences NCBI.
Annexe H.2	Définition et déclaration de variables et de constantes	*Globales*	Module MS VB permettant de définir et de déclarer les variables et les constantes utilisées par les autres modules.
Annexe H.3	Calculs de distances	*Distances des Espèces*	Module MS VB permettant de faire des calculs de dissimilarités inter-génomiques entre les espèces étudiées en fonction des différents coefficients de corrélation utilisés (e.g. Pearson réel et binaire, Tanimoto binaire, etc.).
Annexe H.4	Conversion de séquences d'Acide Aminé en séquences d'ADN	*Conversion AA2DNA*	Module MS VB permettant de faire des conversions de séquences d'Acides Aminés (AA) en séquences d'ADN. Ces conversions en séquences d'ADN s'effectuent sur des séquences d'AA qui se trouvent dans les fichiers FASTA générés à partir des données des tables MS ACCESS. Le programme Ancestor [Diallo et al. 2006] utilise ces séquences pour reconstruire les séquences ancestrales.
Annexe H.5	Conversion de séquences d'ADN en séquences d'Acides Aminés	*Conversion DNA2AA*	Module MS VB permettant de faire des conversions de séquences d'ADN en séquences d'Acides Aminés (AA). Ces conversions en séquences d'AA s'effectuent sur des séquences d'ADN qui se trouvent dans les fichiers FASTA générés à partir du programme Ancestor [Diallo et al. 2006]. Les séquences d'AA ainsi obtenues représentent les séquences ancestrales des phages étudiés.
Annexe H.6	Traitements sur Clusters VOG	*Traitements Clusters VOG*	Module permettant de lire les tables puis d'écrire dans les fichiers sur disque les séquences de protéines (AA) en format FASTA pour chacun des 602 clusters VOG étudiés.
Annexe H.7	Calcul des statistiques de détection des THG	*Statistiques Detection THG*	Module MS VB permettant de calculer les statistiques de détection de THG.
Annexe H.8	Reconstruction de l'arbre de séquences et de fonctions ancestrales	*Nœuds Ancestraux Internes*	Module MS VB permettant de reconstruire les nœuds internes (i.e. iNodes) de l'arbre d'espèces qui représentent les ancêtres des phages étudiés.

H.1 Importation de données de séquences

Les utilitaires standard inclus dans MS ACCESS sont utilisés pour importer les données brutes dans les tables de données (voir les spécifications relationnelles en Annexe G) à partir des fichiers XML qui sont des extraits de données de séquences NCBI.

H.2 Module Globales

Module MS VB permettant de définir et de déclarer les variables et les constantes utilisées par les autres modules.

Globales

```
'*************************************************************************************************
' DESCRIPTION :
'   Les définitons et autres variables globales sont déclarées dans ce module, lesquelles sont utilisées par les autres modules.
'
' TYPES DE VARIABLES GLOBALES ET DE CONSTANCES :
'   Variables Globales, Utilitaires et Debug :
'      g_MAGIC_NUMBER : Numero Magique utilise pour le debuggage à cause d'un nombre important d'exemples à tester.
'                       '>0' => mode Debug; '=0' => mode Normal.
'      g_lenDataSet2BootStrap : Nombre d'échantillons de donnees utilise pour le BootStrap.
'
'      g_dbs : Variable d'ouverture de la base de donnees à utiliser par le module appelant.
'      g_SQL As String : Variable contenant la requete SQL utilisee localement par le module appelant.
'
'      g_BinaryMatrix() As Boolean : Tableau de taille indefini de donnees binaires.
'      g_DistanceMatrix() As Double : Tableau de taille indefini de distances.
'      g_asX, g_asY As Integer : Indices de dimensions X et Y utilisés dans les matrices.
'
'     g_ReplicaBinaryMatrix() As Boolean : Tableau Replica (ou echantillon en termes de BootStrap) de taille indefini de
'                                          donnees binaires.
'      g_ReplicaDistanceMatrix() As Double : Tableau Replica (ou echantillon en termes de BootStrap) de taille indefini de
'                                            distances.
'
'      g_lenOrgName : En mode Normal, ce nombre est lu directement dans le ficheir Input de donnees; en mode Debug, il est
'                     force a un chiffre donne.
'       g_lenClusterNum : En mode Normal, ce nombre est lu directement dans le ficheir Input de donnees; en mode Debug,
'                         il est forceé à un chiffre donné.
'   Requetes SQL :
'      g_OrgNameList : Selection de la liste des noms d'organismes.
'      g_VOGpNumList : Selection de la liste des numeros d'identification des Clusters.
'      g_SpeciesInVOGpListSeqProtein : Selection de la liste des Especes dans Un VOGp.
'      g_SpeciesMoreThanOneTime : Verification si une Espece apparait plus d'une fois.
'      g_SpeciesJustOneTime : Selection de la liste des Especes qui apparaissent seulement une fois.
'
'*************************************************************************************************
```

```
Option Compare Database

Public Const g_MAGIC_NUMBER = 0
Public Const g_lenDataSet2BootStrap = 100

Public g_dbs As Database
Public g_SQL As String

Public g_BinaryMatrix() As Boolean
Public g_DistanceMatrix() As Double
Public g_asX, g_asY As Integer

Public g_ReplicaBinaryMatrix() As Boolean
Public g_ReplicaDistanceMatrix() As Double
```

Globales

```
'Public Const g_lenOrgName = 163          'Utulise en mode Debug
'Public Const g_lenClusterNum = 621       'Utulise en mode Debug
Public g_lenOrgName As Integer            'Utulise en mode Debug
Public g_lenClusterNum As Integer         'Utulise en mode Debug

'Selection liste OrganismName
Public Const g_OrgNameList = _
   "SELECT [Organisms List].ShortOrgName, [Organisms List].Organism " _
   & "FROM [Organisms List] " _
   & "ORDER BY [Organisms List].ShortOrgName;"

'Selection liste ClusterNum
Public Const g_VOGpNumList = _
   "SELECT [(stats) Number of Clusters].[VOGp Cluster] " _
   & "FROM [(stats) Number of Clusters];"

'Selection liste des Especes dans Un VOGp
Public Const g_SpeciesInVOGpListSeqProtein = _
   "SELECT [List of Species in One VOGp].ShortOrgName, [List of Species in One VOGp].Sequence, [List of Species in
One VOGp].[Protein Accession] " _
   & "FROM [List of Species in One VOGp];"

Public Const g_SpeciesInVOGpListSeqNucleotide = _
   "SELECT [List of Species in One VOGp].ShortOrgName, [List of Species in One VOGp].Sequence, [List of Species in
One VOGp].[Gene Accession] " _
   & "FROM [List of Species in One VOGp];"

'Verification si une espece apparait plus d'une fois.
Public Const g_SpeciesMoreThanOneTime = _
   "SELECT [List of Species in One VOGp].ShortOrgName, Count([List of Species in One VOGp].ShortOrgName) AS
CountOfShortOrgName " _
   & "FROM [List of Species in One VOGp] " _
   & "GROUP BY [List of Species in One VOGp].ShortOrgName " _
   & "HAVING (((Count([List of Species in One VOGp].ShortOrgName))>1)) " _
   & "ORDER BY Count([List of Species in One VOGp].ShortOrgName) DESC;"

'Selection liste des Especes qui apparaissent seulement une fois.
Public Const g_SpeciesJustOneTime = _
   "SELECT [List of Species in One VOGp].ShortOrgName, Count([List of Species in One VOGp].ShortOrgName) AS
CountOfShortOrgName " _
   & "FROM [List of Species in One VOGp] " _
   & "GROUP BY [List of Species in One VOGp].ShortOrgName " _
   & "HAVING (((Count([List of Species in One VOGp].ShortOrgName)) = 1));"
```

H.3 Module Distances des Espèces

Module MS VB permettant de faire des calculs de dissimilarités inter-génomiques entre les espèces étudiées en fonction des différents coefficients de corrélation utilisés (e.g. Pearson réel et binaire, Tanimoto binaire, etc.).

Distances des Espèces

```
'*****************************************************************************************************
' DESCRIPTION :
'    La matrice de dissimilarités inter-génomiques est composée de plusieurs matrices intermédiaires . la Matrice binaire
'    (de présence et d'absence de gènes dans cluster VOG), la Matrice de distances originale (avant BootStrap) utilisée comme
'     référence, les N (e.g. 100) réplicas de matrices bianires, et les N (e.g. 100) matrice de distances à partir des N matrices
'     binaires qui seront utilisés comme intrant à l'outil externe appelé Neighbor-Joining (NJ) pour créer une matrice de
'     distances unique de consensus. Les fonctions utlisées (voir ci-après) sont de 3 types : Principale,  Secondaire et Utilitaire.
'
' LISTE DE FONCTIONS :
'    [Principale] SpeciesDistances()   Fonction principale de création de matrices de distances. Cette création s'effectue en
'     plusieurs étapes .  (1): CreateBinaryMatrix2File(); (2): CreateDistanceMatrix2File(); (3): ReturnValue = Shell("path");
'    (4) CreateReplicaDistanceMatrix2File().
'
'    [Secondaire] CreateBinaryMatrix2File() : Création de la matrice binaire (de présence et d'absence de gènes dans cluster
'    VOG).
'    [Secondaire] CreateDistanceMatrix2File() : Création de la matrice de distances originale (avant BootStrap), et écriture dans
'    fichier.
'    [Secondaire] ReturnValue = Shell("path") . Création de N (e.g. 100) réplicas de matrices binaires.
'    [Secondaire] CreateReplicaDistanceMatrix2File() · Création de N (e.g. 100) matrices de distances dans un seul fichier de
'    sortie.
'    [Utilitaire] PearsonCoefficient(ArrayType As Boolean) As Double   Calcul de la matrice de distances avec le coefficiet de
'    Pearson. Cette fonction retourne une matrice de valeurs réelles.
'    [Utilitaire] BinaryPearsonCoefficient(ArrayType As Boolean) As Double : Calcul de la matrice de distances avec le
'    coefficiet de Pearson BINAIRE. Cette fonction retourne une matrice de valeurs réelles.
'
'    [Utilitaire] IsOrgInCluster(orgName As String, ClusterNum As String) As Boolean : Vérification de l'appartenance de
'    l'organisme (i.e. Espèces) dans le Cluster VOG considéré. Cette fonction retourne une valeur booléenne.
'    [Utilitaire] ReadFile2ReplicaBinaryMatrix() : Lecture des N (e.g. 100) réplicas binaires à partir des fichiers.
'    [Utilitaire] PrintBinaryMatrix2File() : Ecriture de la matrice binaire dans un fichier.
'    [Utilitaire] PrintDistanceMatrix2File() : Ecriture de la matrice de distances dans un fichier.
'    [Utilitaire] PrintReadingReplicaBinaryMatrixFOR_DEBUG() · Vérification en mode DEBUG de la lecture des N réplicas.
'    [Utilitaire] PrintReplicaDistanceMatrix2File() · Ecriture des matrices de distances des N réplicas dans les fichiers.
'
'*****************************************************************************************************

Option Compare Database
   Dim rstOrgName As Recordset
   Dim rstClusterNum As Recordset

   Dim matrixHeader As String
```

```
Sub SpeciesDistances()
   Dim RetValue
   Dim wait As Boolean

   Set g_dbs = CurrentDb

   'Sélection de la liste des OrgName & ClusterNum.
   Set qdfOrgName = g_dbs.CreateQueryDef("", g_OrgNameList)
   Set qdfClusterNum = g_dbs.CreateQueryDef("", g_VOGpNumList)
```

Distances des Espèces

```
    'Ouverture des enregistrments à partir de qdfOrgName et de qdfClusterNum.
    Set rstOrgName = qdfOrgName.OpenRecordset(dbOpenSnapshot)
    Set rstClusterNum = qdfClusterNum.OpenRecordset(dbOpenSnapshot)

    'Création de matrices de Distances et Binaires des VOGp### Organisms/Clusters dans les fichiers.
    '.....
    CreateBinaryMatrix2File
    CreateDistanceMatrix2File

    'Duplication de 100 réplicas de matrice binaires avec l'utilitaire SeqBoot
    '.....
    ReturnValue = Shell("C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Replica\My-SeqBoot", 1)

    'Création d'un fichier unique de 100 matrices distances à partir des 100 réplicas de matrices bianires
    '.....
    CreateReplicaDistanceMatrix2File

    'Fermeture des variables de BD
    qdfOrgName.Close
    qdfClusterNum.Close
    g_dbs.Close

    Debug.Print "SpeciesDistances() ---- >>>> Terminé !!!!!"

End Sub

'#########################################################################################################
Private Function CreateBinaryMatrix2File()
    Dim ClusterNum As String
    Dim orgName As String

    Dim indI, indJ As Integer
    Dim i As Byte

    'Mémorisation des enregistrements et comptage du nombre d'enregistrements.
    rstOrgName.MoveLast
    g_lenOrgName = rstOrgName.recordCount

    'Mémorisation des enregistrements et comptage du nombre d'enregistrements.
    rstClusterNum.MoveLast
    g_lenClusterNum = rstClusterNum.recordCount

    ReDim g_BinaryMatrix(1 To g_lenOrgName, 1 To g_lenClusterNum)

    rstOrgName.MoveFirst
    i = 1

    For indI = 1 To UBound(g_BinaryMatrix, 1)          'Pour chaque Organism, faire...
        rstClusterNum.MoveFirst
        orgName = rstOrgName!Organism

        'Affichage de l'enregistrement Cluster Num.
        ClusterNum = rstClusterNum![VOGp Cluster]

        For indJ = 1 To UBound(g_BinaryMatrix, 2)      '... pour chaque groupe de cluster, assigner la chaîne binaire
            If IsOrgInCluster(orgName, ClusterNum) Then 'si l'orgName, i.e. l'espèce, est présent dans cluster
                g_BinaryMatrix(indI, indJ) = 1          'mettre la position à 1.
            Else
                g_BinaryMatrix(indI, indJ) = 0          'sinon à 0.
```

Distances des Espèces

```
        End If

        If indJ < g_lenClusterNum Then
          rstClusterNum.MoveNext
          ClusterNum = rstClusterNum![VOGp Cluster]
        End If
      Next indJ


      rstOrgName.MoveNext

      If Not rstOrgName.EOF Then
        orgName = rstOrgName!Organism
      End If

      If g_MAGIC_NUMBER > 0 And indI = g_MAGIC_NUMBER Then   'Est utilisé en mode DEBUG.
        indI = UBound(g_BinaryMatrix, 1)
      End If

  Next indI

  PrintBinaryMatrix2File      'Sauvegarde du fichier sur le disque local.

End Function

'#############################################################################################
Private Function IsOrgInCluster(orgName As String, ClusterNum As String) As Boolean
  Dim qdfTmp As QueryDef
  Dim strTmp As String
  Dim rstTemp As Recordset
  Dim recordCount As Integer

  'Cération de la requête booléenne.
  strTmp = _
  "SELECT [(import) Genes Description].[VOGp Cluster], [(import) Genes Description].Organism " _
  & "FROM [(import) Genes Description] " _
  & "WHERE [(import) Genes Description].[VOGp Cluster]='" + ClusterNum + "' " _
  & "AND [(import) Genes Description].Organism='" + orgName + "';"

  Set qdfTmp = g_dbs.CreateQueryDef("", strTmp)
  Set rstTemp = qdfTmp.OpenRecordset(dbOpenSnapshot)
  recordCount = rstTemp.recordCount
  qdfTmp.Close

  If recordCount <> 0 Then
    IsOrgInCluster = True
  Else
    IsOrgInCluster = False
  End If

End Function

'#############################################################################################
Private Function CreateDistanceMatrix2File()
  Dim indI, indJ As Integer
  Dim Distance As Double

    If g_MAGIC_NUMBER > 0 Then
      ReDim g_DistanceMatrix(1 To g_MAGIC_NUMBER, 1 To g_MAGIC_NUMBER)
    Else
      ReDim g_DistanceMatrix(1 To g_lenOrgName, 1 To g_lenOrgName)
```

Distances des Espèces

```
    End If

    For indI = 1 To UBound(g_DistanceMatrix, 1)

      For indJ = indI + 1 To UBound(g_DistanceMatrix, 2) 'indJ = indI+1 pour éviter de calculer la comparaison Identitaire.
        g_asX = indI: g_asY = indJ
        Distance = 1 - BinaryPearsonCoefficient(True)
        g_DistanceMatrix(indI, indJ) = Distance
        g_DistanceMatrix(indJ, indI) = Distance     'Copie Mirroir dans la matrice carrée.
      Next indJ

    Next indI

    PrintDistanceMatrix2File

End Function

'#########################################################################################################
Private Function CreateReplicaDistanceMatrix2File()
  Dim indI, indJ As Integer
  Dim dataSetStep As Integer
  Dim Distance As Double
  Dim ind_OrgName As Integer

  If MAGIC_NUMBER > 0 Then
    ReDim g_ReplicaDistanceMatrix(1 To (MAGIC_NUMBER * g_lenDataSet2BootStrap), 1 To MAGIC_NUMBER)
    ReDim g_ReplicaBinaryMatrix(1 To (MAGIC_NUMBER * g_lenDataSet2BootStrap), 1 To (g_lenClusterNum))
  Else
    ReDim g_ReplicaDistanceMatrix(1 To (g_lenOrgName * g_lenDataSet2BootStrap), 1 To g_lenOrgName)
    ReDim g_ReplicaBinaryMatrix(1 To (g_lenOrgName * g_lenDataSet2BootStrap), 1 To (g_lenClusterNum))
  End If

  ReadFile2ReplicaBinaryMatrix
  'Point de Debug
  PrintReadingReplicaBinaryMatrixFOR_DEBUG

  dataSetStep = 0
  ind_OrgName = 1
  For indI = 1 To UBound(g_ReplicaDistanceMatrix, 1)

    If (indI Mod g_lenOrgName) <> 0 Then ' N'est pas nécessaire grâce à la copie mirroir, voir plus bas.

      'indJ = ((indI - dataSetStep) + 1) pour éviter de calculer la comparaison Identitaire.
      For indJ = ((indI - dataSetStep) + 1) To UBound(g_ReplicaDistanceMatrix, 2)
        g_asX = indI: g_asY = indJ
        Distance = 1 - BinaryPearsonCoefficient(False)
        g_ReplicaDistanceMatrix(indI, indJ) = Distance
        g_ReplicaDistanceMatrix(indJ + dataSetStep, indI - dataSetStep) = Distance  'Copie Mirroir dans la matrice carrée.
      Next indJ
    End If

    ind_OrgName = ind_OrgName + 1
    If ind_OrgName = g_lenOrgName Then
      dataSetStep = dataSetStep + g_lenOrgName
      ind_OrgName = 0
    End If

  Next indI
```

Distances des Espèces

```
'#############################################################################################
PrintReplicaDistanceMatrix2File

End Function

Private Function PearsonCoefficient(ArrayType As Boolean) As Double
  'Ecrit par P. Wester et adapté par D. NGUYEN
  'wester@kpd.nl
  'Rev. 1.00 29 july 2002, Rev. 2.00 19 august 2002 changed weightfactors in absolute weightfactors.

  Dim i As Integer
  Dim SumX, SumX2, SumY, SumY2, SumXY, SumW As Double
  Dim Weight, Numerator, Denominator As Double

  'Initialisation
  SumX = 0: SumX2 = 0: SumY = 0: SumY2 = 0: SumXY = 0: SumW = 0
  Numerator = 0: Denominator = 0

  'Calcul de SumX, SumX2, SumY, SumY2 and SumXY
  If ArrayType Then
    For i = 1 To UBound(g_BinaryMatrix, 2)
      Weight = 1  'Reminescence du code source de P. Wester utilisée pour la pondération, par défaut est égale à 1.
      SumX = SumX + g_BinaryMatrix(g_asX, i) * Weight
      SumX2 = SumX2 + g_BinaryMatrix(g_asX, i) * g_BinaryMatrix(g_asX, i) * Weight
      SumY = SumY + g_BinaryMatrix(g_asY, i) * Weight
      SumY2 = SumY2 + g_BinaryMatrix(g_asY, i) * g_BinaryMatrix(g_asY, i) * Weight
      SumXY = SumXY + g_BinaryMatrix(g_asX, i) * g_BinaryMatrix(g_asY, i) * Weight
      SumW = SumW + Weight
    Next i
  Else
    For i = 1 To UBound(g_ReplicaBinaryMatrix, 2)
      Weight = 1  'Reminescence du code source de P. Wester utilisée pour la pondération, par défaut est égale à 1.
      SumX = SumX + g_ReplicaBinaryMatrix(g_asX, i) * Weight
      SumX2 = SumX2 + g_ReplicaBinaryMatrix(g_asX, i) * g_ReplicaBinaryMatrix(g_asX, i) * Weight
      SumY = SumY + g_ReplicaBinaryMatrix(g_asY, i) * Weight
      SumY2 = SumY2 + g_ReplicaBinaryMatrix(g_asY, i) * g_ReplicaBinaryMatrix(g_asY, i) * Weight
      SumXY = SumXY + g_ReplicaBinaryMatrix(g_asX, i) * g_ReplicaBinaryMatrix(g_asY, i) * Weight
      SumW = SumW + Weight
    Next i
  End If

  'Calsul du Numérateur et du Dénominateur à partir du coefficient de corrélation.
  Numerator = SumXY - (SumX * SumY / SumW)
  Denominator = Sqr((SumX2 - SumX * SumX / SumW) * (SumY2 - SumY * SumY / SumW))

  If Denominator <> 0 Then
    PearsonCoefficient = Numerator / Denominator
  Else
    '
  End If

End Function
```

Distances des Espèces

```
'#########################################################################################################
'Private Function TanimotoCoefficient(ArrayType As Boolean) As Double
Private Function BinaryPearsonCoefficient(ArrayType As Boolean) As Double ' Autre coefficient alternatif à Pearson.

  Dim i As Integer
  Dim a, b, c, d As Integer
  Dim Numerator, Denominator As Double

  'Initialisation
  a = 0  'Présent dans les deux échantillons.
  b = 0  'Présent dans l'échantillon i, mais absent dans l'échantillon j.
  c = 0  'Présent dans l'échantillon j, mais absent dans l'échantillon i.
  d = 0  'Absent dans les deux échantillons.

  'Calcul
  If ArrayType Then
    For i = 1 To UBound(g_BinaryMatrix, 2)
      If g_BinaryMatrix(g_asX, i) = True And g_BinaryMatrix(g_asY, i) = True Then
        a = a + 1
      End If
      If g_BinaryMatrix(g_asX, i) = True And g_BinaryMatrix(g_asY, i) = False Then
        b = b + 1
      End If
      If g_BinaryMatrix(g_asX, i) = False And g_BinaryMatrix(g_asY, i) = True Then
        c = c + 1
      End If
      If g_BinaryMatrix(g_asX, i) = False And g_BinaryMatrix(g_asY, i) = False Then
        d = d + 1
      End If
    Next i
  Else
    For i = 1 To UBound(g_ReplicaBinaryMatrix, 2)
      If g_ReplicaBinaryMatrix(g_asX, i) = True And g_ReplicaBinaryMatrix(g_asY, i) = True Then
        a = a + 1
      End If
      If g_ReplicaBinaryMatrix(g_asX, i) = True And g_ReplicaBinaryMatrix(g_asY, i) = False Then
        b = b + 1
      End If
      If g_ReplicaBinaryMatrix(g_asX, i) = False And g_ReplicaBinaryMatrix(g_asY, i) = True Then
        c = c + 1
      End If
      If g_ReplicaBinaryMatrix(g_asX, i) = False And g_ReplicaBinaryMatrix(g_asY, i) = False Then
        d = d + 1
      End If
    Next i
  End If

  'TanimotoCoefficient = (a + d) / ((a + d) + (2 * (b + c)))          'Cas où le coefficient de Tanimoto est utilisé.

  "Calsul du Numérateur et du Dénominateur à partir du coefficient de Pearson binaire.  'Cas où le coefficient de Pearson
bianire est utilisé.
  Numerator = (a * d) - (b * c)                                      'Cas où le coefficient de Pearson bianire est utilisé.
  Denominator = Sqr((a + b) * (a + c) * (b + d) * (c + d))           'Cas où le coefficient de Pearson bianire est utilisé.

  If Denominator <> 0 Then                                           'Cas où le coefficient de Pearson bianire est utilisé.
    BinaryPearsonCoefficient = Numerator / Denominator               'Cas où le coefficient de Pearson bianire est utilisé.
  Else                                                               'Cas où le coefficient de Pearson bianire est utilisé.
    '                                                                'Cas où le coefficient de Pearson bianire est utilisé.
  End If                                                             'Cas où le coefficient de Pearson bianire est utilisé.

End Function
```

Distances des Espèces

```
'#############################################################################################
Private Function PrintBinaryMatrix2File()
   Dim str2File As String
   Dim indI, indJ As Integer

   'Ouverture de fichier pour écriture.
   Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Replica\BinaryMatrix" For Output As #1

   'Total du nombre d'organismes + taille du vecteur binaire.
   str2File = "   " & CStr(g_lenOrgName) + "          " & CStr(g_lenClusterNum)
   Print #1, str2File

   rstOrgName.MoveFirst
   For indI = 1 To UBound(g_BinaryMatrix, 1) 'Pour chaque organisme, affichage du vecteur binaire.
      'Nom de l'Organism dans la première colonne.
      str2File = rstOrgName!ShortOrgName + "              "
      'Les valeurs de la matrice de distances dans le reste du tableau.
      For indJ = 1 To UBound(g_BinaryMatrix, 2)
         If g_BinaryMatrix(indI, indJ) = True Then
            str2File = str2File + "1"
         Else
            str2File = str2File + "0"
         End If
      Next indJ

      Print #1, str2File
      rstOrgName.MoveNext

   Next indI
   Close #1

End Function

'#############################################################################################
Private Function PrintDistanceMatrix2File()
   Dim str2File As String
   Dim indI, indJ As Integer

   'Ouverture de fichier pour écriture.
   Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Replica\DistanceMatrix" For Output As #1

   Print #1, UBound(g_DistanceMatrix, 1)
   rstOrgName.MoveFirst
   For indI = 1 To UBound(g_DistanceMatrix, 1) 'Pour chaque organisme, affichage de la valeur de la distance.
      'Nom de l'Organism dans la première colonne.
      str2File = rstOrgName!ShortOrgName + "              "
      'Les valeurs de la matrice de distances dans le reste du tableau.
      For indJ = 1 To UBound(g_DistanceMatrix, 2)
         str2File = str2File + " " + CStr(g_DistanceMatrix(indI, indJ))
      Next indJ

      Print #1, str2File
      rstOrgName.MoveNext

   Next indI
   Close #1

End Function
```

Distances des Espèces

```
'#############################################################################################
Private Function ReadFile2ReplicaBinaryMatrix()
  Dim inCar As String
  Dim orgNumber As Integer
  Dim isHeader As Boolean
  Dim isOrgName As Boolean
  Dim ind_LenBinVector As Integer
  Dim lenBinVector As Integer
  Dim ind_OrgName As Integer

  'Ouverture de fichier pour écriture.
  Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Replica\ReplicaBinaryMatrix" For Input As #1    ' Open file for input.

  isHeader = True
  isOrgName = False
  ind_LenBinVector = 1
  orgNumber = 1
  ind_OrgName = 1
  orgName = ""
  matrixHeader = ""
  lenBinVector = g_lenClusterNum

  Do While Not EOF(1)              ' Bouclage jusqu'à la fin du fichier
    inCar = Input(1, #1)          'Récupération d'un caractère à la fois.

    Select Case inCar             ' Évaluation du caractère.
    Case Chr(32)                  ' C'est un caractère blanc et non une entête.
      'Debug.Print "Blank"
      If isHeader And ind_OrgName = 1 Then
        matrixHeader = matrixHeader + CStr(inCar)
      End If
      If isOrgName Then
        orgName = ""
        isOrgName = False
      End If

    Case Chr(10)          ' C'est un saut de ligne (linefeed - LF).
    Case Chr(13)          ' C'est un cariage return (CR).
      'Debug.Print "CR"
      If isHeader Then
        isHeader = False
        isOrgName = True
      End If


      If ind_LenBinVector > lenBinVector Then
        If orgNumber < g_lenOrgName Then
          isOrgName = True
          orgNumber = orgNumber + 1
        Else
          orgNumber = 1
          isHeader = True
          isOrgName = False
        End If
        ind_LenBinVector = 1
        ind_OrgName = ind_OrgName + 1
      End If

    Case Else             'C'est une valeur binaire (caractère 0/1).
```

Distances des Espèces

```
        If isHeader And ind_OrgName = 1 Then
          matrixHeader = matrixHeader + CStr(inCar)
        End If
        If Not isHeader And Not isOrgName Then
          g_ReplicaBinaryMatrix(ind_OrgName, ind_LenBinVector) = inCar
          ind_LenBinVector = ind_LenBinVector + 1
        End If

      End Select

  'Debug.Print inCar

  Loop
  Close #1   ' Fermeture de fichie.

End Function

'#########################################################################################################
Private Function PrintReplicaDistanceMatrix2File()
  Dim str2File As String
  Dim indMax, indI, indJ, indK As Integer

  'Ouverture de fichier pour écriture.
  indK = 1
  str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Replica\ReplicaDistanceMatrix" + CStr(indK)
  Open str2File For Output As #1

  Print #1, CStr(g_lenOrgName)
  rstOrgName.MoveFirst
  indMax = UBound(g_ReplicaDistanceMatrix, 1)
  For indI = 1 To indMax 'Pour chaque ensemble de (replica) d'organismes, affichage de la valeur de la distance.
    'Nom de l'Organism dans la première colonne.
    str2File = rstOrgName!ShortOrgName + "          "
    'Les valeurs de la matrice de distances dans le reste du tableau.
    For indJ = 1 To UBound(g_ReplicaDistanceMatrix, 2)
      str2File = str2File + " " + CStr(g_ReplicaDistanceMatrix(indI, indJ))
    Next indJ

    Print #1, str2File
    If (indI Mod g_lenOrgName) <> 0 Then
      rstOrgName.MoveNext
    Else                        'Next set of (replica) organisms
      If (indI + 1) < indMax Then
        'Ouverture de fichier pour écriture.
        Close #1
        indK = indK + 1
        str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Replica\ReplicaDistanceMatrix" + CStr(indK)
        Open str2File For Output As #1
        Print #1, CStr(g_lenOrgName)
        rstOrgName.MoveFirst
      End If
    End If

  Next indI
  Close #1

End Function
```

Distances des Espèces

```
'#########################################################################################################
Public Function PrintReadingReplicaBinaryMatrixFOR_DEBUG()
   Dim str2File As String
   Dim indI, indJ As Integer

   Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Replica\ReadingReplicaBinaryMatrixFOR_DEBUG" For Output As #1

   For indI = 1 To UBound(g_ReplicaBinaryMatrix, 1)
      str2File = ""
      For indJ = 1 To UBound(g_ReplicaBinaryMatrix, 2) 'Pour chaque organisme, affichage de son nom.
            If g_ReplicaBinaryMatrix(indI, indJ) = True Then
              str2File = str2File + "1"
            Else
              str2File = str2File + "0"
            End If
      Next indJ
      Print #1, str2File
   Next indI

   Close #1    ' Fermeture de fichier.

End Function
```

H.4 Module Conversion AA2DNA

Module MS VB permettant de faire des conversions de séquences d'Acides Aminés (AA) en séquences d'ADN. Ces conversions en séquences d'ADN s'effectuent sur des séquences d'AA qui se trouvent dans les fichiers FASTA générés à partir des données des tables MS ACCESS. Le programme Ancestor [Diallo et al. 2006] utilise ces séquences pour reconstruire les séquences ancestrales.

Conversion AA2DNA

```
'*************************************************************************************************
' DESCRIPTION :
'   Lecture du fichier intrant avec des codes des Acides Aminés, puis conversion en codes d'ADN et écriture dans le fichier
'   extrant. Le fichier extrant avec des codes DNA sera utilisé par le programme Ancestor pour reconstuire les séquences
'   ancestrales.
'
' LISTE DE FONCTIONS :
'   ConvertAA2DNA() : Fonction permettant de convertir les codes AA en code DNA.
'
'*************************************************************************************************

Option Compare Database
```

```
Sub ConvertAA2DNA()
    Dim qdfTmp As QueryDef
    Dim rstTmp As Recordset
    Dim indI, lenRecord As Integer
    Dim str2File As String
    Dim isTitle As Boolean
    Dim bigString As String

    Set g_dbs = CurrentDb
    g_SQL = _
    "SELECT [(import) Analysed VOG].File " _
    & "FROM [(import) Analysed VOG];"

    Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

    rstTmp.MoveLast
    lenRecord = rstTmp.recordCount
    rstTmp.MoveFirst
    For indI = 1 To lenRecord
        'Affectation du chemin du répertoire TEST.
        str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\SetOfCluster\TEST\"
        str2File = str2File + rstTmp!File
        Open str2File For Input As #1     'Ouverture de fichier pour lecture.
        str2File = str2File + ".out"
        Open str2File For Output As #2    'Ouverture de fichier pour écriture.

        '(1)Lecture du fichier intrant, et (2) Remplissage dans le fichier extrant.
        '-------------------------------------------------------------------------------
        isTitle = False
        bigString = ""
        Do While Not EOF(1)               'Bouclage jusqu'à la fin du fichier.
```

Conversion AA2DNA

```
inCar = Input(1, #1)          'Récupération d'un caractère à la fois.

Select Case inCar             'Évaluation du caractère.
Case Chr(62)                  'C'est un caractère ">".
   isTitle = True
   bigString = bigString + inCar

Case Chr(13)                  'C'est un cariage return (CR) ou
   bigString = bigString + inCar
Case Chr(10)                  'C'est un saut de ligne (linefeed - LF).
   If isTitle Then
      isTitle = False
   End If
   Print #2, bigString        'Affichage dans le fichier à écrire.
   bigString = ""

Case Else                     'Remplacement du code de l'Acide Aminé par le code codon, and "U" par "T".
   If isTitle Then
      bigString = bigString + inCar
   Else
      Select Case inCar       'Évaluation du caractère.
         Case Chr(45)         'C'est le caractère "-".
            bigString = bigString + "---"

         Case Chr(70)         'C'est le caractère "F".
            '"UUC"
            bigString = bigString + "TTC"

         Case Chr(76)         'C'est le caractère "L".
            '"UUG"
            bigString = bigString + "TTG"

         Case Chr(73)         'C'est le caractère "I"
            '"AUC"
            bigString = bigString + "ATC"

         Case Chr(77)         'C'est le caractère "M".
            '"AUG"
            bigString = bigString + "ATG"

         Case Chr(86)         'C'est le caractère "V".
            '"GUC"
            bigString = bigString + "GTC"

         Case Chr(83)         'C'est le caractère "S".
            '"UCC"
            bigString = bigString + "TCC"

         Case Chr(80)         'C'est le caractère "P".
            bigString = bigString + "CCC"

         Case Chr(84)         'C'est le caractère "T".
            bigString = bigString + "ACC"

         Case Chr(65)         'C'est le caractère "A".
            bigString = bigString + "GCC"

         Case Chr(89)         'C'est le caractère "Y".
            '"UAC"
            bigString = bigString + "TAC"
```

Conversion AA2DNA

```
            Case Chr(72)                'C'est le caractère "H".
              bigString = bigString + "CAC"

            Case Chr(81)                'C'est le caractère "Q".
              bigString = bigString + "CAG"

            Case Chr(78)                'C'est le caractère "N".
              bigString = bigString + "AAC"

            Case Chr(75)                'C'est le caractère "K".
              bigString = bigString + "AAG"

            Case Chr(68)                'C'est le caractère "D".
              bigString = bigString + "GAC"

            Case Chr(69)                'C'est le caractère "E".
              bigString = bigString + "GAG"

            Case Chr(67)                'C'est le caractère "C".
              '"UGC"
              bigString = bigString + "TGC"

            Case Chr(82)                'C'est le caractère "R".
              bigString = bigString + "CGC"

            Case Chr(71)                'C'est le caractère "G".
              bigString = bigString + "GGC"

            Case Chr(87)                'C'est le caractère "W".
              '"UGG"
              bigString = bigString + "TGG"

            Case Chr(88)                'C'est le caractère "X".
              '"N'importe quel Nucleotide"
              bigString = bigString + "NAT"

            Case Else                   'C'est un cas IMPOSSIBLE.
              Debug.Print "ERRRRRRRRRRRREEEEEEEEEUUUUUUUUUUURRRRRRRRRRR!!!!!!"
              bigString = bigString + "ERROR!!!"
              Exit Do
          End Select
        End If
      End Select
    Loop
    Close #2    'Fermeture de fichier.
    Close #1    'Fermeture de fichier.

    rstTmp.MoveNext
  Next indI
  rstTmp.Close
  qdfTmp.Close

  g_dbs.Close
  Set g_dbs = Nothing

  Debug.Print "ReformatOutpuFile() ---- >>>>TERMINÉ !!!!!"

End Sub
```

H.5 Module Conversion DNA2AA

Module MS VB permettant de faire des conversions de séquences d'ADN en séquences d'Acides Aminés (AA). Ces conversions en séquences d'AA s'effectuent sur des séquences d'ADN qui se trouvent dans les fichiers FASTA générés à partir du programme Ancestor [Diallo et al. 2006]. Les séquences d'AA ainsi obtenues représentent les séquences ancestrales des phages étudiés.

Conversion DNA2AA

```
'*************************************************************************************************
' DESCRIPTION :
'   Lecture du fichier intrant avec des codes d'ADN, puis conversion en codes d'Acides Aminés et écriture dans dans la table
'   (import) Analysed VOG"  Les séquences de cette table représentent les séquences protéiques ancestrales des Phages
'   étudiés.
'
' LISTE DE FONCTIONS :
'   [Principale] ConvertDNA2AA() : Fonction principale permettant de convertir les codes ADN en code AA. Cette fonction
'   fait appel à la fonction UdapeAncestralSeq() pour mettre à jour la table iNodes.
'
'   [Secondaire] UdapeAncestralSeq(numVOG As String) : Fonction permettant de mettre à jour la table iNodes des séquences
'   ancestrales.
'
'*************************************************************************************************

Option Compare Database
Dim bigString, numVOG As String
```

```
Sub ConvertDNA2AA()
    Dim qdfTmp As QueryDef
    Dim rstTmp As Recordset
    Dim indI, lenRecord As Integer
    Dim str2File, inCar As String
    Dim in1C, in2C, in3C, in3Car As String
    Dim numVOG As String
    Dim Group() As String
    Dim exec As Boolean

    Set g_dbs = CurrentDb
    'Création temporaire de regroupement d'espèces.
    g_SQL = _
    "SELECT [(import) Analysed VOG].File " _
    & "FROM [(import) Analysed VOG];"


    Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

    rstTmp.MoveLast
    lenRecord = rstTmp.recordCount
    rstTmp.MoveFirst

    str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\OutFiles\Errors"
    Open str2File For Output As #2     'Ouverture de fichier pour écrire d'eéventuelles erreurs.
```

Conversion DNA2AA

```
For indI = 1 To lenRecord
    'Affectation du chemin du répertoire OutFiles.
    str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\OutFiles\"
    numVOG = rstTmp!File
    str2File = str2File + numVOG + ".fasta.out.output"
    Open str2File For Input As #1    'Ouverture de fichier pour lecture.

    '(1)Lecture du fichier intrant, et (2) Remplissage dans la table iNodes.
    '-------------------------------------------------------------------------------
    bigString = ""
    in1C = ""
    in2C = ""
    in3C = ""
    in3Car = ""
    Do While Not EOF(1)              'Bouclage jusqu'à la fin du fichier.

      inCar = Input(1, #1)           'Récupération de 3 caractères à la fois.

      Select Case inCar              'Évaluation du caractère.
        Case Chr(63)                 'C'est le caractère "?".
          bigString = bigString + inCar

        Case Chr(13)                 'C'est un cariage return (CR) ou
        Case Chr(10)                 'C'est un saut de ligne (linefeed - LF).
          UdapeAncestralSeq numVOG   'Mise à jour dans la table iNodes.

        Case Else                    'Remplacement de 3-ADN par l'Amino Acide.
          If Len(in3Car) < 3 Then
            exec = True
            If exec And Len(in3Car) = 0 Then
              in1C = inCar
              in3Car = in3Car + inCar
              exec = False
            End If
            If exec And Len(in3Car) = 1 Then
              in2C = inCar
              in3Car = in3Car + inCar
              exec = False
            End If
            If exec And Len(in3Car) = 2 Then
              in3C = inCar
              in3Car = in3Car + inCar
            End If
          End If

          If exec Then               ' Conversion selon la table de CODON
                                     ' e.g. http://www.kazusa.or.jp/java/codon_table_java/
            If exec And (in3Car = "---" Or in1C = "-") Then
              'Si c'est un GAP (i.e. "-" or "---") ne rien ajouter
              'bigString = bigString + "-"
              exec = False
            End If

            If exec And (in2C = "-" Or in3C = "-" Or (in2C = "-" And in3C = "-")) Then
              bigString = bigString + "N"
              exec = False
            End If

            If exec And (in3Car = "NAT") Then
              '"N'importe quel codon"
```

Conversion DNA2AA

```
    bigString = bigString + "X"
    exec = False
End If

If exec And (in3Car = "GCT" Or in3Car = "GCC" Or in3Car = "GCA" Or in3Car = "GCG") Then
    bigString = bigString + "A"
    exec = False
End If

If exec And (in3Car = "CGT" Or in3Car = "CGC" Or in3Car = "CGA" _
Or in3Car = "CGG" Or in3Car = "AGA" Or in3Car = "AGG") Then
    bigString = bigString + "R"
    exec = False
End If

If exec And (in3Car = "AAT" Or in3Car = "AAC") Then
    bigString = bigString + "N"
    exec = False
End If

If exec And (in3Car = "GAT" Or in3Car = "GAC") Then
    bigString = bigString + "D"
    exec = False
End If

If exec And (in3Car = "TGT" Or in3Car = "TGC") Then
    bigString = bigString + "C"
    exec = False
End If

If exec And (in3Car = "CAA" Or in3Car = "CAG") Then
    bigString = bigString + "Q"
    exec = False
End If

If exec And (in3Car = "GAA" Or in3Car = "GAG") Then
    bigString = bigString + "E"
    exec = False
End If

If exec And (in3Car = "GGT" Or in3Car = "GGC" Or in3Car = "GGA" Or in3Car = "GGG") Then
    bigString = bigString + "G"
    exec = False
End If

If exec And (in3Car = "CAT" Or in3Car = "CAC") Then
    bigString = bigString + "H"
    exec = False
End If

If exec And (in3Car = "ATT" Or in3Car = "ATC" Or in3Car = "ATA") Then
    bigString = bigString + "I"
    exec = False
End If

If exec And (in3Car = "TTA" Or in3Car = "TTG" Or in3Car = "CTT" _
Or in3Car = "CTC" Or in3Car = "CTA" Or in3Car = "CTG") Then
    bigString = bigString + "L"
    exec = False
End If
```

Conversion DNA2AA

```
If exec And (in3Car = "AAA" Or in3Car = "AAG") Then
   bigString = bigString + "K"
   exec = False
End If

If exec And (in3Car = "ATG") Then
   bigString = bigString + "M"
   exec = False
End If

If exec And (in3Car = "TTT" Or in3Car = "TTC") Then
   bigString = bigString + "F"
   exec = False
End If

If exec And (in3Car = "CCT" Or in3Car = "CCC" Or in3Car = "CCA" Or in3Car = "CCG") Then
   bigString = bigString + "P"
   exec = False
End If

If exec And (in3Car = "TCT" Or in3Car = "TCC" Or in3Car = "TCA" _
Or in3Car = "TCG" Or in3Car = "AGT" Or in3Car = "AGC") Then
   bigString = bigString + "S"
   exec = False
End If

If exec And (in3Car = "ACT" Or in3Car = "ACC" Or in3Car = "ACA" Or in3Car = "ACG") Then
   bigString = bigString + "T"
   exec = False
End If

If exec And (in3Car = "TGG") Then
   bigString = bigString + "W"
   exec = False
End If

If exec And (in3Car = "TAT" Or in3Car = "TAC") Then
   bigString = bigString + "Y"
   exec = False
End If

If exec And (in3Car = "GTT" Or in3Car = "GTC" Or in3Car = "GTA" Or in3Car = "GTG") Then
   bigString = bigString + "V"
   exec = False
End If

'"Z" represente le 'codon Stop'
If exec And (in3Car = "TAG" Or in3Car = "TAA" Or in3Car = "TGA") Then
   bigString = bigString + "Z"
   exec = False
End If

If exec And (in3Car <> "") Then
   Debug.Print "Codon = " + in3Car
   Print #2, in3Car       'Print to OUT file
   Debug.Print "ERRRRRRRRRRRREEEEEEEEEEEEUUUUUUUUUURRRRRRRRRR!!!!!!"
   exec = False
End If
in1C = ""
in2C = ""
in3C = ""
```

Conversion DNA2AA

```
                in3Car = ""
              End If
          End Select
        Loop
        Close #1    'Fermeture de fichier.

        rstTmp.MoveNext
    Next indI

    Close #2    'Fermeture de fichier.
    rstTmp.Close
    qdfTmp.Close

    g_dbs.Close
    Set g_dbs = Nothing

    Debug.Print "ConvertDNA2AA() ---- >>>>TERMINÉ !!!!!"
End Sub

'#############################################################################################################
Private Function UdapeAncestralSeq(numVOG As String)
    Dim dbs As Database
    Dim qdfTmp As QueryDef
    Dim rstTmp As Recordset
    Dim lenRecord As Integer
    Dim indI As Integer

    Set dbs = CurrentDb
    'Création de la liste temporaire de HGT Detection.
    g_SQL = _
    "SELECT [(import) iNodes].VOG, [(import) iNodes].AncSeq " _
    & "FROM [(import) iNodes] " _
    & "WHERE ((([(import) iNodes].VOG)='" + numVOG + "'));"

    Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenDynaset)

    rstTmp.Edit
    rstTmp!AncSeq = bigString
    rstTmp.Update

    rstTmp.Close
    qdfTmp.Close
    dbs.Close

End Function
```

H.6 Module Traitements Clusters VOG

Module permettant de lire les tables puis d'écrire dans les fichiers sur disque les séquences de protéines (AA) ou de nucléotides en format FASTA pour chacun des 602 clusters VOG étudiés.

Traitements Clusters VOG

```
'*************************************************************************************************
' DESCRIPTION :
'   Pour chacun des 602 clusters VOG, un fichier est créé sur disque dans lequel on retouve des séquences de AA ou de
'  nucléotides en format FASTA associées à ce cluster.
'
'  NOTE  dans le cas des séquences AA, on peut trouver dans un même cluster VOG, plusieurs protéines d'un même phage.
'  Par conséquent, il faut générer en autant de combinaisons qu'il y a de protéines de ce phage afin de déterminer la
'  combinaison qui a le score d'alignement de séquences (via l'outil ClustalW [Thompson et al. 1994]) le plus élévé, et de
'  retenir celle-ci pour représenter le cluster VOG.
'
' LISTE DE FONCTIONS :
'  [Principale] CreateSetOfClusters()  Fonction principale permettant de créer un fichier par cluster VOG avec des séquences
'  de protéines (via PrintClusterSeqProteine2File()) ou des séquences d'AA (via PrintClusterSeqNucleotide2File()).
'
'   [Secondaire] PrintClusterSeqProteine2File(ClusterNum As String) : Fonction permettant de créer un fichier par cluster
'  VOG avec des séquences de protéines. Elle fait appelle à la fonction RecombiningSpecies().
'  [Secondaire] PrintClusterSeqNucleotide2File(ClusterNum As String) : Fonction permettant de créer un fichier par cluster
'  VOG avec des séquences d'AA.
'
'  [Utilitaire] RecombiningSpecies() . Fonction permettant de créer des combinaisons de séquences AA.
'
'*************************************************************************************************

Option Compare Database
   Dim SpeciesInVOGp() As String
   Dim RecombMatrix() As Integer
   Dim qdfTmp As QueryDef
   Dim rstTmp As Recordset
```

```
Sub CreateSetOfClusters()
   Dim rstClusterNum As Recordset
   Dim ClusterNum As String
   Dim indI, maxCluster As Integer

   Set g_dbs = CurrentDb

   ' Sélection de la liste ClusterNum.
   Set qdfClusterNum = g_dbs.CreateQueryDef("", g_VOGpNumList)
   Set rstClusterNum = qdfClusterNum.OpenRecordset(dbOpenSnapshot)
   rstClusterNum.MoveLast
   g_lenClusterNum = rstClusterNum.recordCount

   rstClusterNum.MoveFirst
   For indI = 1 To g_lenClusterNum

      ClusterNum = rstClusterNum![VOGp Cluster]

      'PrintClusterSeqProteine2File (ClusterNum)
```

Traitements Clusters VOG

```
      PrintClusterSeqNucleotide2File (ClusterNum)

      rstClusterNum.MoveNext
      If Not rstClusterNum.EOF Then
        ClusterNum = rstClusterNum![VOGp Cluster]
      End If

      If MAGIC_NUMBER > 0 And indI = MAGIC_NUMBER Then
        indI = g_lenClusterNum
      End If

    Next indI

    qdfClusterNum.Close

    Debug.Print "CreateSetOfClusters() ---- >>>> FIN NORMALE !!!!!"

End Sub

'#########################################################################################################
Public Function PrintClusterSeqProteine2File(ClusterNum As String)
    Dim lenOrg As Integer
    Dim str2File As String
    Dim OrgNum, indI, indJ As Integer
    Dim recombining As Boolean

    Set g_dbs = CurrentDb
    'Création temporaire de la liste d'Espèces dans UN cluster VOGp.
    g_SQL = _
    "SELECT [Organisms List].ShortOrgName, [(import) Proteins Sequences].Sequence, [(import) Genes
Description].[Protein Accession] INTO [List of Species in One VOGp] " _
    & "FROM [Organisms List] INNER JOIN ([(import) Genes Description] INNER JOIN [(import) Proteins Sequences] ON
[(import) Genes Description].GI = [(import) Proteins Sequences].GI) ON [Organisms List].Organism = [(import) Genes
Description].Organism " _
    & "GROUP BY [Organisms List].ShortOrgName, [(import) Proteins Sequences].Sequence, [(import) Genes
Description].[Protein Accession], [(import) Genes Description].[VOGp Cluster], [(import) Genes Description].[Cluster
Classification] " _
    & "HAVING ((([(import) Genes Description].[VOGp Cluster])='" + ClusterNum + "') AND (([(import) Genes
Description].[Cluster Classification])='VOG')) " _
    & "ORDER BY [Organisms List].ShortOrgName, [(import) Genes Description].[VOGp Cluster];"

    DoCmd.RunSQL g_SQL

    Set qdfTmp = g_dbs.CreateQueryDef("", g_SpeciesInVOGpListSeqProtein)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

    'Comptage du nombre d'enregistrements, et remplissage d'enregistrements dans la matrice "SpeciesInVOGp".
    rstTmp.MoveLast
    lenOrg = rstTmp.recordCount
    ReDim SpeciesInVOGp(1 To lenOrg, 1 To 3)
    rstTmp.MoveFirst
    For indI = 1 To UBound(SpeciesInVOGp, 1)
      SpeciesInVOGp(indI, 1) = rstTmp!ShortOrgName
      SpeciesInVOGp(indI, 2) = rstTmp![Gene Accession]
      SpeciesInVOGp(indI, 3) = rstTmp!Sequence
      rstTmp.MoveNext
    Next indI
    'Fermeture de la table temporaire.
    qdfTmp.Close

    'Vérification si une espèce apparaît plus d'une fois.
```

Traitements Clusters VOG

```
    Set qdfTmp = g_dbs.CreateQueryDef("", g_SpeciesMoreThanOneTime)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

    'Comptage du nombre d'enregistrements.
    lenOrg = rstTmp.recordCount
    If lenOrg = 0 Then      'Pas de recombinaison nécessaire.
       recombining = False
       'Construction de la matrice d'espèces RecombMatrix.
       ReDim RecombMatrix(1 To 1, 1 To UBound(SpeciesInVOGp, 1))
       For indJ = 1 To UBound(RecombMatrix, 2)
            RecombMatrix(1, indJ) = indJ
       Next indJ
    Else                'Recombinaison requise.
       recombining = True
       RecombiningSpecies
    End If

    For indI = 1 To UBound(RecombMatrix, 1)
       'Affichage dans un ou plusieurs fichier(s).
       If recombining = False Then
          str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\SetOfCluster\" + ClusterNum
       Else
          str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\SetOfCluster\RecombiningClusters\" + ClusterNum + ".1\" + ClusterNum + "."
+ CStr(indI)
          'str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\SetOfCluster\RecombiningClusters\" + ClusterNum + "." + CStr(indI)
       End If
       ' Ouerture de fichier pour écriture.
       Open str2File For Output As #1

       For indJ = 1 To UBound(RecombMatrix, 2)
          OrgNum = RecombMatrix(indI, indJ)
          str2File = ">" + SpeciesInVOGp(OrgNum, 1) + " | "        'ShortOrgName.
          str2File = str2File + SpeciesInVOGp(OrgNum, 2)           'Protein Accession.
          Print #1, str2File
          Print #1, SpeciesInVOGp(OrgNum, 3)                       'Sequence.
       Next indJ
       Close #1    ' Fermeture de fichier.
    Next indI

    'Fermeture de la table temporaire.
    qdfTmp.Close
    g_dbs.Close
    DoCmd.DeleteObject acTable, "List of Species in One VOGp"

End Function

'#########################################################################################################
Public Function PrintClusterSeqNucleotide2File(ClusterNum As String)
    Dim RetValue
    Dim lenOrg As Integer
    Dim str2File As String
    Dim OrgNum, indI, indJ As Integer
    Dim recombining As Boolean

    Set g_dbs = CurrentDb
    'Création temporaire de la liste d'Espèces dans UN cluster VOGp.
    g_SQL = _
    "SELECT [Organisms List].ShortOrgName, [(import) Genes Description].[Gene Accession], [(import) DNA
```

Traitements Clusters VOG

```
Sequences].Sequence INTO [List of Species in One VOGp] " _
   & "FROM [(import) DNA Sequences] INNER JOIN ([Organisms List] INNER JOIN [(import) Genes Description] ON
[Organisms List].Organism = [(import) Genes Description].Organism) ON [(import) DNA Sequences].[DNA Seq] =
[(import) Genes Description].[DNA Seq] " _
   & "WHERE ((([(import) Genes Description].[VOGp Cluster])='" + ClusterNum + "')) " _
   & "ORDER BY [Organisms List].ShortOrgName;"

   DoCmd.RunSQL g_SQL

   Set qdfTmp = g_dbs.CreateQueryDef("", g_SpeciesInVOGpListSeqNucleotide)
   Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

   'Comptage du nombre d'enregistrements, et remplissage d'enregistrements dans la matrice "SpeciesInVOGp".
   rstTmp.MoveLast
   lenOrg = rstTmp.recordCount
   ReDim SpeciesInVOGp(1 To lenOrg, 1 To 3)
   rstTmp.MoveFirst
   For indI = 1 To UBound(SpeciesInVOGp, 1)
      SpeciesInVOGp(indI, 1) = rstTmp!ShortOrgName
      SpeciesInVOGp(indI, 2) = rstTmp![Gene Accession]
      SpeciesInVOGp(indI, 3) = rstTmp!Sequence
      rstTmp.MoveNext
   Next indI
   'Fermeture de la table temporaire.
   qdfTmp.Close

   'Vérification si une espèce apparaît plus d'une fois.
   Set qdfTmp = g_dbs.CreateQueryDef("", g_SpeciesMoreThanOneTime)
   Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

   'Comptage du nombre d'enregistrements.
   lenOrg = rstTmp.recordCount
   If lenOrg = 0 Then      'Pas de recombinaison nécessaire.
      recombining = False
      'Construction de la matrice d'espèces RecombMatrix.
      ReDim RecombMatrix(1 To 1, 1 To UBound(SpeciesInVOGp, 1))
      For indJ = 1 To UBound(RecombMatrix, 2)
            RecombMatrix(1, indJ) = indJ
      Next indJ
   Else                'Recombinaison requise.
      recombining = True
      RecombiningSpecies
   End If

   For indI = 1 To UBound(RecombMatrix, 1)
      'Affichage dans un ou plusieurs fichier(s).
      If recombining = False Then
         str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\SetOfCluster\" + ClusterNum
      Else
         str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\SetOfCluster\RecombiningClusters\" + ClusterNum + ".1\" + ClusterNum + "."
+ CStr(indI)
         'str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\SetOfCluster\RecombiningClusters\" + ClusterNum + "." + CStr(indI)
      End If
      ' Ouerture de fichier pour écriture.
      Open str2File For Output As #1

      For indJ = 1 To UBound(RecombMatrix, 2)
         OrgNum = RecombMatrix(indI, indJ)
```

Traitements Clusters VOG

```
        str2File = ">" + SpeciesInVOGp(OrgNum, 1) + "|"        'ShortOrgName.
        str2File = str2File + SpeciesInVOGp(OrgNum, 2)         'Nucleotide Accession.
        Print #1, str2File
        Print #1, SpeciesInVOGp(OrgNum, 3)                     'Sequence.
      Next indJ
      Close #1    ' Fermeture de fichier.
    Next indI

    'Fermeture de la table temporaire.
    qdfTmp.Close
    g_dbs.Close
    DoCmd.DeleteObject acTable, "List of Species in One VOGp"

End Function

'#########################################################################################################
Public Function RecombiningSpecies()
    Dim indI, indJ, lgth As Integer
    Dim lenOrg As Integer
    Dim K, L, Step, Iter As Integer
    Dim indRecomb, indPreviousRecomb As Integer
    Dim UniqueList() As Integer
    Dim DuplicatList() As Integer
    Dim RecombVector() As String
    Dim qdf As QueryDef
    Dim rst As Recordset
    Dim aRecombStr() As String
    Dim isExistUniqueList As Boolean

    'Liste d'espèces qui apparaissent juste une seule fois.
    Set qdf = g_dbs.CreateQueryDef("", g_SpeciesJustOneTime)
    Set rst = qdf.OpenRecordset(dbOpenSnapshot)

    'Construction de la liste Unique d'espèces qui apparaî une seule fois.
    isExistUniqueList = False
    If Not rst.recordCount = 0 Then
      isExistUniqueList = True
      rst.MoveLast
      ReDim UniqueList(1 To rst.recordCount)
      L = 1
      rst.MoveFirst
      Do While Not rst.EOF
        K = 1
        Do While True
          If StrComp(rst!ShortOrgName, SpeciesInVOGp(K, 1), 1) = 0 Then
            UniqueList(L) = K
            Exit Do
          End If
          K = K + 1
        Loop
        rst.MoveNext
        L = L + 1
      Loop
    End If

    'Construction de la liste DuplicatList d'espèces qui apparaissent plus 'dune fois.
    rstTmp.MoveLast
    lenOrg = rstTmp.recordCount
    rstTmp.MoveFirst

    'Mise dans la colonne 0 pour sauvegarder le nombre d'occurences.
```

Traitements Clusters VOG

```
'Le plus grand nombre de rstTmp!CountOfShortOrgName dans la requête est au dessus de la pile.
ReDim DuplicatList(1 To lenOrg, 0 To rstTmp!CountOfShortOrgName)
For indI = 1 To UBound(DuplicatList, 1)
   DuplicatList(indI, 0) = rstTmp!CountOfShortOrgName
   For indJ = 1 To UBound(DuplicatList, 2)
      DuplicatList(indI, indJ) = 0
   Next indJ
   rstTmp.MoveNext
Next indI

rstTmp.MoveFirst
For indI = 1 To UBound(DuplicatList, 1)
   K = 1
   For indJ = 1 To rstTmp!CountOfShortOrgName
      For L = K To UBound(SpeciesInVOGp, 1)
         If StrComp(rstTmp!ShortOrgName, SpeciesInVOGp(L, 1), 1) = 0 Then
            DuplicatList(indI, indJ) = L
            K = L + 1
            Exit For
         End If
      Next L
   Next indJ
   rstTmp.MoveNext
Next indI

'Construction le vecteur RecombVector de toutes les espèces recombinées.
lgth = 1
rstTmp.MoveFirst
Do While Not rstTmp.EOF
   lgth = lgth * rstTmp!CountOfShortOrgName
   rstTmp.MoveNext
Loop
ReDim RecombVector(1 To lgth)
For indI = 1 To UBound(RecombVector, 1)
   RecombVector(indI) = "0"
Next indI

indRecomb = 1
For indI = 1 To UBound(DuplicatList, 1)

   Step = 1
   indPreviousRecomb = indRecomb

   indRecomb = indRecomb * DuplicatList(indI, 0)

   Iter = indPreviousRecomb

   For indJ = 1 To DuplicatList(indI, 0)

      K = Step

      Do While K <= UBound(RecombVector, 1)
         For L = K To Iter

            If indI = 1 Then    'Remplissage de la première itération.
               RecombVector(L) = CStr(DuplicatList(indI, indJ))

            Else            'Remplissage des itérations subséquentes en additionnant le caractère d'espèce (" ").
               RecombVector(L) = RecombVector(L) + " " + CStr(DuplicatList(indI, indJ))

            End If
```

Traitements Clusters VOG

```
            Next L
            K = K + indRecomb

            Iter = Iter + indRecomb

          Loop

          Step = Step + indPreviousRecomb

          If indI = 1 Then
            Iter = Step
          Else
            Iter = Step + indPreviousRecomb - 1
          End If
        Next indJ

    Next indI

    'Construction de la matrice d'espèces RecombMatrix.
    If isExistUniqueList Then
      ReDim RecombMatrix(1 To lgth, 1 To (UBound(UniqueList, 1) + lenOrg))
      For indI = 1 To UBound(RecombMatrix, 1)
        For indJ = 1 To UBound(UniqueList, 1)
          RecombMatrix(indI, indJ) = UniqueList(indJ)
        Next indJ
      Next indI
    Else
      ReDim RecombMatrix(1 To lgth, 1 To lenOrg)
    End If


    For indI = 1 To UBound(RecombMatrix, 1)
      'Séparation de la chaîne passed-in afin de la convertir en un tableau de chaînes.
      aRecombStr = Split(RecombVector(indI))

      K = 0
      If isExistUniqueList Then
        L = UBound(UniqueList, 1) + 1
      Else
        L = 1
      End If

      For indJ = L To (L + lenOrg - 1)
        RecombMatrix(indI, indJ) = CInt(aRecombStr(K))
        K = K + 1
      Next indJ

    Next indI

End Function
```

H.7 Module Statistiques Detection THG

Module MS VB permettant de calculer les statistiques de détection de THG.

Statistiques Detection THG

```
'*****************************************************************************************************
' DESCRIPTION .
'   Les statistiques de HGT Detection sont calculées suivant les critères de mouvements de transferts (1) Inter/Intra groupes
' d'espèces,
'   (2) Intrant/Extrant relatif à chacun des groupes d'espèces, et (3) Interactions entre chacun des groupes d'espèces entre eux.
'
' LISTE DE FONCTIONS :
'   [Principale] HGTDetectionStats() : Fonction principale permettant d'afficher les statistiques de HGT Detection dans 3
'    fichiers distincts grâce aux 3 fonctions : Print2FileHGTInterActionGroup(), Print2FileHGTIntrantExtrant() et
'    Print2FileHGTInterIntraMovementGroup(). Elle fait également appel à la fonction IdentifyHGTtype().
'
'   [Secondaire] IdentifyHGTtype(tmpString As String) As String : Fonction permettant d'identifier le type de HGT, s'il est de
'   type 'X' cela signifie qu'il s'agit d'un  groupe non numéroté, et donc considéré comme un transfert de/ou à partir d'un groupe
'   non numéroté.
'   [Secondaire] Print2FileHGTInterActionGroup(numSrcGroup As String) : Fonction permettant de calculer les statistiques
'   relatives aux transferts Interactions de groupes.
'   [Secondaire] Print2FileHGTIntrantExtrant(numGroup As String) : Fonction permettant de calculer les statistiques relatives
'   aux transferts Intrant/Extrant de groupes.
'    [Secondaire] Print2FileHGTInterIntraMovementGroup(numGroup As String, ind As Integer) : Fonction permettant de
'   calculer les statistiques relatives aux transferts  Inter/Intra de groupes.
'
'*****************************************************************************************************
Option Compare Database
   Dim Group() As String
   Dim str2File As String
```

```
Sub HGTDetectionStats()
   Dim qdfTmp As QueryDef
   Dim rstTmp As Recordset
   Dim srcString(), dstString() As String
   Dim lenRecord As Integer
   Dim indI As Integer

   Set g_dbs = CurrentDb
   'Création temporaire de regroupement d'espèces.
   g_SQL = _
   "SELECT Grouping.ShortOrgName, Grouping.Group " _
   & "FROM Grouping;"

   Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
   Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

   'Comptage du nombre d'enregistrements, et remplissage d'enregistrements dans la matrice "Group".
   rstTmp.MoveLast
   lenRecord = rstTmp.recordCount
   ReDim Group(1 To lenRecord, 1 To 2)
   rstTmp.MoveFirst
   For indI = 1 To UBound(Group, 1)
      Group(indI, 1) = rstTmp!ShortOrgName
      Group(indI, 2) = CStr(rstTmp!Group)
      rstTmp.MoveNext
```

Statistiques Detection THG

```
    Next indI
    rstTmp.Close
    qdfTmp.Close

    'Création temporaire de la liste de HGT Detection.
    g_SQL = _
    "SELECT [(import) HGT Results].[Organism Source], [(import) HGT Results].[Organism Target], " _
    & "[(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
    & "FROM [(import) HGT Results];"

    Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenDynaset)

    ' Modification de données dans l'enregistrement local.
    Do While Not rstTmp.EOF
       rstTmp.Edit
       rstTmp![Group Source] = IdentifyHGTtype(rstTmp![Organism Source])
       rstTmp![Group Target] = IdentifyHGTtype(rstTmp![Organism Target])
       rstTmp.Update
       rstTmp.MoveNext
    Loop
    rstTmp.Close
    qdfTmp.Close

    'Création temporaire du nombre de groupe.
    g_SQL = _
    "SELECT Grouping.Group, Groups.Size, Groups.BootStrap " _
    & "FROM Groups INNER JOIN Grouping ON Groups.Group = Grouping.Group " _
    & "GROUP BY Grouping.Group, Groups.Size, Groups.BootStrap;"

    Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

    'Affichage dans le fichier HGT_IntrantExtrant_Stats.txt.
    str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de Recherche\TRAVAIL\Version1\WorkSpace\HGT
Detection\HGT_IntrantExtrant_Stats.txt"
    Open str2File For Output As #1
    str2File = "Group      " + "Intrant[X]      " + "Extrant[X]     " + "Intrant[!X]      " + "Extrant[!X]     "
    Print #1, str2File

    'Count the number of records, and populate Recordset in the matrix "Group".
    rstTmp.MoveLast
    lenRecord = rstTmp.recordCount
    ReDim Group(1 To lenRecord, 1 To 3)
    rstTmp.MoveFirst
    For indI = 1 To UBound(Group, 1)
       Group(indI, 1) = rstTmp!Group
       Group(indI, 2) = rstTmp!Size
       Group(indI, 3) = rstTmp!BootStrap
       Print2FileHGTIntrantExtrant Group(indI, 1)
       rstTmp.MoveNext
    Next indI
    Close #1

    'Affichage dans le fichier HGT_InterActionGroupe_Stats.txt.
    str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de Recherche\TRAVAIL\Version1\WorkSpace\HGT
Detection\HGT_InterActionGroupe_Stats.txt"
    Open str2File For Output As #1
    str2File = "Group      "
    rstTmp.MoveFirst
    For indI = 1 To UBound(Group, 1)
```

Statistiques Detection THG

```
    str2File = str2File + Group(indI, 1) + "          "
  Next indI
  Print #1, str2File

  For indI = 1 To UBound(Group, 1)
    str2File = Group(indI, 1) + "       "
    Print2FileHGTInterActionGroup Group(indI, 1)
  Next indI
  Close #1

  'Affichage dans le fichier HGT_InterIntraMovement_Stats.txt.
  str2File = "C:\Documents and Settings\Owner\Desktop\Sujet de Recherche\TRAVAIL\Version1\WorkSpace\HGT
Detection\HGT_InterIntraMovement_Stats.txt"
  Open str2File For Output As #1
  str2File = "Group      " + "Intra-HGT      " + "Inter-HGT      " + "Size      " + "Bootstrap      "
  Print #1, str2File
  For indI = 1 To UBound(Group, 1)
    str2File = Group(indI, 1) + "       "
    Print2FileHGTInterIntraMovementGroup Group(indI, 1), indI
  Next indI
  Close #1

  rstTmp.Close
  qdfTmp.Close
  g_dbs.Close
  Debug.Print "HGTDetectionStats() ---- >>>> FIN NORMALE !!!!!"

End Sub

'#########################################################################################################
Function IdentifyHGTtype(tmpString As String) As String
  Dim Value, Val As String
  Dim indI, indJ As Integer
  Dim aStr() As String

  Val = ""
  Value = ""

  'Séparation de la chaîne passed-in afin de la convertir en un tableau de chaînes.
  aStr = Split(tmpString, ";")
  For indI = 0 To UBound(aStr)
    For indJ = 1 To UBound(Group, 1)
      If aStr(indI) = Group(indJ, 1) Then
        Val = Group(indJ, 2)
        Exit For
      End If
    Next indJ

    If Value = "" Then              'Première assignation.
      Value = Val
      If Value = "X" Then           'Le premier element de la chaîne "X".
        Exit For
      End If
    Else
      If Val = "X" _
       Or (Not Val = Value) Then    'Un des éléments est "X" ou différent du reste.
        Value = "X"
        Exit For
      End If
    End If
  Next indI
```

Statistiques Detection THG

```
    IdentifyHGTtype = Value

End Function

'#########################################################################################################
Function Print2FileHGTIntrantExtrant(numGroup As String)
    Dim dbs As Database
    Dim int_qdfTmp, ext_qdfTmp As QueryDef
    Dim int_rstTmp, ext_rstTmp As Recordset
    Dim int_SQL, ext_SQL As String
    Dim int_lenRecord, ext_lenRecord As Integer
    Dim int_qdfTmp2, ext_qdfTmp2 As QueryDef
    Dim int_rstTmp2, ext_rstTmp2 As Recordset
    Dim int_SQL2, ext_SQL2 As String
    Dim int_lenRecord2, ext_lenRecord2 As Integer
    Dim indl As Integer

    Set dbs = CurrentDb
    int_SQL = _
    "SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
    & "FROM [(import) HGT Results] " _
    & "WHERE ((([(import) HGT Results].[Group Target])='" + numGroup + "'));"  'Nombre cible.
    ext_SQL = _
    "SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
    & "FROM [(import) HGT Results] " _
    & "WHERE ((([(import) HGT Results].[Group Source])='" + numGroup + "'));"  'Nombre Source.
    Set int_qdfTmp = dbs.CreateQueryDef("", int_SQL)
    Set int_rstTmp = int_qdfTmp.OpenRecordset(dbOpenSnapshot)
    Set ext_qdfTmp = dbs.CreateQueryDef("", ext_SQL)
    Set ext_rstTmp = ext_qdfTmp.OpenRecordset(dbOpenSnapshot)

    int_SQL2 = _
    "SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
    & "FROM [(import) HGT Results] " _
    & "WHERE ((([(import) HGT Results].[Group Target])='" + numGroup + "') AND (Not ([(import) HGT Results].[Group
Target])='X'));"
    ext_SQL2 = _
    "SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
    & "FROM [(import) HGT Results] " _
    & "WHERE ((([(import) HGT Results].[Group Source])='" + numGroup + "') AND (Not ([(import) HGT Results].[Group
Source])='X'));"
    Set int_qdfTmp2 = dbs.CreateQueryDef("", int_SQL2)
    Set int_rstTmp2 = int_qdfTmp2.OpenRecordset(dbOpenSnapshot)
    Set ext_qdfTmp2 = dbs.CreateQueryDef("", ext_SQL2)
    Set ext_rstTmp2 = ext_qdfTmp2.OpenRecordset(dbOpenSnapshot)

    'Comptage du nombre d'enregistrements à partir des requêtes Intrant/Extrant.
    int_lenRecord = 0
    ext_lenRecord = 0
    int_lenRecord2 = 0
    ext_lenRecord2 = 0
    If Not int_rstTmp.recordCount = 0 Then
        int_rstTmp.MoveLast
        int_lenRecord = int_rstTmp.recordCount
    End If
    If Not ext_rstTmp.recordCount = 0 Then
        ext_rstTmp.MoveLast
        ext_lenRecord = ext_rstTmp.recordCount
    End If
    If Not int_rstTmp2.recordCount = 0 Then
```

Statistiques Detection THG

```
    int_rstTmp2.MoveLast
    int_lenRecord2 = int_rstTmp2.recordCount
  End If
  If Not ext_rstTmp2.recordCount = 0 Then
    ext_rstTmp2.MoveLast
    ext_lenRecord2 = ext_rstTmp2.recordCount
  End If
  str2File = numGroup + "          " + CStr(int_lenRecord) + "          " + CStr(ext_lenRecord) + "          " _
                      + CStr(int_lenRecord2) + "          " + CStr(ext_lenRecord2)
  Print #1, str2File

  int_qdfTmp.Close
  int_rstTmp.Close
  ext_qdfTmp.Close
  ext_rstTmp.Close

End Function

'#########################################################################################################
Function Print2FileHGTInterActionGroup(numSrcGroup As String)
  Dim dbs As Database
  Dim qdfTmp As QueryDef
  Dim rstTmp As Recordset
  Dim lenRecord As Integer
  Dim indJ As Integer

  Set dbs = CurrentDb
  For indJ = 1 To UBound(Group, 1)
    g_SQL = _
    "SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
    & "FROM [(import) HGT Results] " _
    & "WHERE ((([(import) HGT Results].[Group Source])='" + numSrcGroup + "') " _
    & "AND (([(import) HGT Results].[Group Target])='" + Group(indJ, 1) + "'));"
    Set qdfTmp = dbs.CreateQueryDef("", g_SQL)
    Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

    'Comptage du nombre d'enregistrements.
    lenRecord = 0
    If Not rstTmp.recordCount = 0 Then
      rstTmp.MoveLast
      lenRecord = rstTmp.recordCount
    End If
    str2File = str2File + "       " + CStr(lenRecord)
  Next indJ
  Print #1, str2File

  qdfTmp.Close:  rstTmp.Close
End Function

'#########################################################################################################
Function Print2FileHGTInterIntraMovementGroup(numGroup As String, ind As Integer)
  Dim dbs As Database
  Dim qdfTmp As QueryDef
  Dim rstTmp As Recordset
  Dim lenRecordIntraGrp As Integer
  Dim lenRecordInterGrp, lenRecordInterGrp2 As Integer
  Dim indI As Integer

  Set dbs = CurrentDb

  'Stats Intra : Source <=> Target
```

Statistiques Detection THG

```
g_SQL = _
"SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
& "FROM [(import) HGT Results] " _
& "WHERE ((([(import) HGT Results].[Group Source])='" + numGroup + "') " _
& "AND (([(import) HGT Results].[Group Target])='" + numGroup + "'));"
Set qdfTmp = dbs.CreateQueryDef("", g_SQL)
Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

'Comptage du nombre d'enregistrments.
lenRecordIntraGrp = 0
If Not rstTmp.recordCount = 0 Then
   rstTmp.MoveLast
   lenRecordIntraGrp = rstTmp.recordCount
End If
qdfTmp.Close:  rstTmp.Close

'Stats Inter : Source = Groupe A ;
'Cible = N'importe quel Groupe MAIS Non Groupe "X", et par opposition (i.e. Non Groupe "X" MAIS n'importe autres).
g_SQL = _
"SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
& "FROM [(import) HGT Results] " _
& "WHERE ((([(import) HGT Results].[Group Source])='" + numGroup + "') " _
& "AND (Not ([(import) HGT Results].[Group Target])='" + numGroup + "' " _
& "And Not ([(import) HGT Results].[Group Target])='X'));"
Set qdfTmp = dbs.CreateQueryDef("", g_SQL)
Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

'Comptage du nombre d'enregistrements.
lenRecordInterGrp = 0
If Not rstTmp.recordCount = 0 Then
   rstTmp.MoveLast
   lenRecordInterGrp = rstTmp.recordCount
End If
qdfTmp.Close
rstTmp.Close

g_SQL = _
"SELECT [(import) HGT Results].[Group Source], [(import) HGT Results].[Group Target] " _
& "FROM [(import) HGT Results] " _
& "WHERE ((Not ([(import) HGT Results].[Group Source])='" + numGroup + "' " _
& "And Not ([(import) HGT Results].[Group Source])='X') " _
& "AND (([(import) HGT Results].[Group Target])='" + numGroup + "'));"
Set qdfTmp = dbs.CreateQueryDef("", g_SQL)
Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

'Comptage du nombre d'enregistrements.
lenRecordInterGrp2 = 0
If Not rstTmp.recordCount = 0 Then
   rstTmp.MoveLast
   lenRecordInterGrp2 = rstTmp.recordCount
End If
qdfTmp.Close : rstTmp.Close

lenRecordInterGrp = lenRecordInterGrp + lenRecordInterGrp2

str2File = str2File + "      " + CStr(lenRecordIntraGrp) + "      " + CStr(lenRecordInterGrp) + "      " _
+ CStr(Group(ind, 2)) + "      " + CStr(Group(ind, 3))
Print #1, str2File

End Function
```

H.8 Module Nœuds Ancestraux Internes

Module MS VB permettant de reconstruire les nœuds internes (i.e. iNodes) de l'arbre d'espèces qui représentent les ancêtres des phages étudiés.

Nœuds Ancestraux Internes

```
'*************************************************************************************************
' DESCRIPTION :
'   L'arbre phylogénétique numéroté de noeuds internes est généré par la fonction iNodes() ci-dessous. La table "(import)
'    iNodes" est associée à l'arbre numéroté. Dans cette table, y figurent les informations sur le noeud interne (iNode), la
'    fonction protéique relative au VOG étutidé, ainsi que la séquence protéique ancestrale générée par le programme Ancestor.
'
" LISTE DE FONCTIONS :
'   [Principale] iNodes() : Fonction principale de création de l'arbre phylogénétique numéroté. Elle remplit également la table
'   "(import) iNodes" d'informations sur le numéro de noeud interne, la fonction protéique annotée associée au VOG considéré
'  et la séquence protéique ancestrale générée par Ancestor. Cette fonction fait appel aux fonctions secondaires
'   InternalNodeReading() et VOGReading().
'
'   [Secondaire] InternalNodeReading() As Integer : Fonction permettant de créer l'arbre numéroté interne (avec la liste des
'    sous-espèces à partir du noeud interne aux feuilles). Cette fonction retourne une valeur entière signifiant le nombre de
'    nœud interne effectif. Elle fait appel aux fonctions RemoveLastLevel() et ULevel().
'   [Secondaire] VOGReading(inode As Integer)  Assignation pour chaque noeud interne, de la fonction protéique annotée
'    associée au VOG considéré. Cette fonction fait appel à la fonction utilitaire SetVOGInfo().
'
'   [Utilitaire] RemoveLastLevel(str As String) As String : Fonction permettant de supprimer le dernier niveau de l'arbre
'    numéroté. Elle retourne une chaîne de caractères.
'   [Utilitaire] ULevel(str As String) As Integer : Fonction permettant de renseigner sur l'état Unaire ou non du noeud interne
'    considéré.
'   [Utilitaire] SetVOGInfo(indice As Integer) : Fonction permettant de renseigner les information sur les espèces, la fonction
'    protéique et les iNodes' dans la liste "VOGList".
'
'*************************************************************************************************

Option Compare Database

Dim bigStr As String

Dim iNodeSorted() As Variant
Const cst_iNode = 1
Const cst_NbOfSpecies = 2
Const cst_SpList = 3
Const cst_StartSorting = 3

Dim VOGList() As Variant
Dim VOGListSorted() As Variant
Dim VOGListSortedTmp() As Variant
Const cst_VOG = 1
'Const cst_NbOfSpecies = 2
'Const cst_SpList = 3
Const cst_Function = 4
Const cst_NbOfiNode = 5
```

```
Sub iNodes()
   Dim indI, iNodeNb As Integer
   Dim g_dbs As Database
   Dim num As Integer
   Dim func, vog As String
```

Nœuds Ancestraux Internes

```
    iNodeNb = InternalNodeReading
    VOGReading iNodeNb

    'Ouverture de fichier pour écriture.
    Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\ArbreNumerote.txt" For Output As #1
    Print #1, bigStr
    Close #1

    'MODE DEBUG   Ouverture de fichier pour écriture.
    'Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\NoeudsFonction.txt" For Output As #1
    'For indI = 1 To UBound(VOGListSorted)
        'Print #1, CStr(VOGListSorted(indI, cst_NbOfiNode)) + "#" + VOGListSorted(indI, cst_Function) + "#" +
VOGListSorted(indI, cst_VOG)
    'Next indI
    'Close #1

    Set g_dbs = CurrentDb
    For indI = 1 To UBound(VOGListSortedTmp)
        num = VOGListSortedTmp(indI, cst_NbOfiNode)
        func = VOGListSortedTmp(indI, cst_Function)
        vog = VOGListSortedTmp(indI, cst_VOG)

        g_dbs.Execute "INSERT INTO [(import) iNodes] VALUES ('" + CStr(num) + "', '" + func + "', '" + vog + "', '')"
    Next indI
    g_dbs.Close


    Debug.Print "iNodes() ---- >>>> TERMINÉ !!!!!"

End Sub

'#########################################################################################################
Private Function InternalNodeReading() As Integer
    Dim inode() As Variant
    Const cstFlag = 1
    Const cstLevels = 2
    Const cstUBinary = 3
    Const cstSpeciesList = 4
    Const cstNumberOfSpecies = 5

    Dim levelStr As String
    Dim aStr() As String
    Dim aLevels() As String
    Dim inCar, inElem, speciesList As String
    Dim NbiNode, iNodeNb, currentLevel As Integer
    Dim i, indI, indJ, ptrPOS As Integer
    Dim isInternalNode As Boolean

    'Ouverture de fichier pour écriture.
    Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\arbre.txt" For Input As #1

    'Lecture permettant de récupérer le nombre des noeuds internes.
    NbiNode = 0: bigStr = ""
    Do While Not EOF(1)            'Bouclage jusqu'à la fin du fichier.
        inCar = Input(1, #1)       'Récupération d'un caractère à la fois.
        Select Case inCar          'Évaluation du caractère.
            Case Chr(44)           'C'est un caractère ",".
```

Nœuds Ancestraux Internes

```
        NbiNode = NbiNode + 1
        inCar = " " + inCar + " "
      Case Chr(40)              'C'est un caractère "(".
        inCar = inCar + " "
      Case Chr(41)              'C'est un caractère ")".
        inCar = " " + inCar
      Case Chr(59)              'C'est un caractère ";".
        inCar = " " + inCar
    End Select
    bigStr = bigStr + inCar
Loop
Close #1    'Fermeture de fichier.

ReDim inode(0 To NbiNode - 1, 1 To 5)
aStr = Split(bigStr, " ")          'Transformation de la chaîne en lecture dans un tableau pour des traitements ultérieurs.

bigStr = "": levelStr = "": iNodeNb = -1: ptrPOS = -1: isInternalNode = False
For indI = 0 To UBound(aStr)
    inElem = aStr(indI)
    Select Case inElem              'Évaluation de Elemt.

      Case Chr(40)              'C'est un caractère "("
        ptrPOS = ptrPOS + 1                 'Mise du pointeur ptr de la positiob iNode à +1.
        iNodeNb = ptrPOS                    'Mise du nombre de iNode à ptrPOS.
        inode(ptrPOS, cstFlag) = 1           'Set the flag of iNode to '1'.
        inode(ptrPOS, cstFlag) = 1           'Mise du drapeau de iNode à '1'
        If ptrPOS = 0 Then
          levelStr = CStr(iNodeNb)
        Else
          levelStr = levelStr + " " + CStr(iNodeNb)
        End If

        inode(ptrPOS, cstLevels) = levelStr       'Mise la chaîne de niveau de iNode au niveau de lecture.
        inode(ptrPOS, cstUBinary) = "U"           'Mise du drapeau UBinary de iNode à la position Unary.

      Case Chr(41)              'C'est un caractère ")".
        inode(currentLevel, cstFlag) = 0
        If ptrPOS = iNodeNb Then
          inode(ptrPOS, cstUBinary) = "B"       'Mise du drapeau UBinary de iNode à la position Binaire.
        End If

        'C'est un noeud Unaire, i.e. node interne.
        If levelStr <> "" Then
          currentLevel = ULevel(levelStr)

          If inode(currentLevel, cstUBinary) = "B" Then   'Si c'est un noeud Bianire,
            'Suppression du dernier niveau.
            levelStr = inode(currentLevel, cstLevels)
            levelStr = RemoveLastLevel(levelStr)
            inode(currentLevel, cstLevels) = levelStr
          End If

          If inode(currentLevel, cstUBinary) = "U" Then   'Si c'est un noeud Unaire,
            inode(currentLevel, cstUBinary) = "B"         'Alors mettre à B, i.e. non final = "concatable"
            speciesList = inode(currentLevel, cstSpeciesList)
            For indJ = currentLevel + 1 To UBound(inode, 1)
              'La liste speciesList "concatable"
              If inode(indJ, cstLevels) = inode(currentLevel, cstLevels) And _
                inode(indJ, cstUBinary) = "B" _
              Then
                speciesList = LTrim(RTrim(speciesList + " " + inode(indJ, cstSpeciesList)))
```

Nœuds Ancestraux Internes

```
                    inode(indJ, cstUBinary) = "X"   'Alors mettre à X, i.e. final = non "concatable"

                    levelStr = inode(indJ, cstLevels)
                    levelStr = RemoveLastLevel(levelStr)
                    inode(indJ, cstLevels) = levelStr
                  End If
                Next indJ
                inode(currentLevel, cstSpeciesList) = speciesList

                inode(currentLevel, cstUBinary) = "B"
                'Suppression du dernier niveau.
                levelStr = inode(currentLevel, cstLevels)
                levelStr = RemoveLastLevel(levelStr)
                inode(currentLevel, cstLevels) = levelStr
              End If

              iNodeNb = iNodeNb - 1              'Mise le niveau de lecture à -1
              isInternalNode = True
            End If

          Case Chr(44)            'C'est un caractère ",".
          Case Chr(59)            'C'est un caractère ";".
            'Ne rien faire

          Case Else
            inode(ULevel(levelStr), cstSpeciesList) = RTrim(LTrim(" " + inode(ULevel(levelStr), cstSpeciesList) + " " +
inElem))

        End Select

        If isInternalNode Then
          bigStr = bigStr + inElem + CStr(currentLevel)   'Insertion du nombre des noeuds internes à la variable bigStr.
          isInternalNode = False
        Else
          bigStr = bigStr + inElem
        End If

      Next indI

      'Remplissage du nombre d'Espèces de chaque noeud interne...
      For indI = 0 To UBound(inode)
        aStr = Split(inode(indI, cstSpeciesList), " ")
        inode(indI, cstNumberOfSpecies) = UBound(aStr) + 1
      Next indI
      '... et tri ces nombres en fonction de l'ordre Ascendant.
      i = 0
      For indI = cst_StartSorting To (NbiNode + 1)          '(nbiNode + 1): est le nombre Total d'espèces étudiées.
        For indJ = 0 To UBound(inode)
          If inode(indJ, cstNumberOfSpecies) = indI Then
            i = i + 1                          'Sélection de seulement ceux qui ont au moins #cst_StartSorting d'espèces.
          End If
        Next indJ
      Next indI
      ReDim iNodeSorted(0 To i - 1, 1 To 3)
      i = 0
      For indI = cst_StartSorting To (NbiNode + 1)
        For indJ = 0 To UBound(inode)
          If inode(indJ, cstNumberOfSpecies) = indI Then
            iNodeSorted(i, cst_iNode) = indJ
            iNodeSorted(i, cst_NbOfSpecies) = inode(indJ, cstNumberOfSpecies)
            iNodeSorted(i, cst_SpList) = inode(indJ, cstSpeciesList)
```

Nœuds Ancestraux Internes

```
            i = i + 1
          End If
        Next indJ
    Next indI

    'MODE DEBUG   Ouverture de fichier pour écriture.
    Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\noeudsInternes.txt" For Output As #1
    For indI = 0 To UBound(inode)
        Print #1, CStr(indI) + "#" + CStr(inode(indI, cstNumberOfSpecies)) + "#" + inode(indI, cstSpeciesList)
    Next indI
    Close #1

    'MODE DEBUG : Ouverture de fichier pour écriture.
    Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\noeudsInternesTries.txt" For Output As #1
    For indI = 0 To UBound(iNodeSorted)
        Print #1, CStr(iNodeSorted(indI, cst_iNode)) + "#" + CStr(iNodeSorted(indI, cst_NbOfSpecies)) + "#" +
iNodeSorted(indI, cst_SpList)
    Next indI
    Close #1

    InternalNodeReading = NbiNode

End Function

'#############################################################################################################
Private Function RemoveLastLevel(str As String) As String
    Dim tmpStr As String
    Dim aLevels() As String

    aLevels = Split(str)
    tmpStr = ""
    For indK = 0 To UBound(aLevels) - 1
        tmpStr = tmpStr + " " + aLevels(indK)
    Next indK
    tmpStr = RTrim(LTrim(tmpStr))

    RemoveLastLevel = tmpStr
End Function

'#############################################################################################################
Private Function ULevel(str As String) As Integer
    Dim aLevels() As String

    aLevels = Split(str)
    ULevel = aLevels(UBound(aLevels))
End Function



'#############################################################################################################
Private Function VOGReading(inode As Integer)
    Dim qdfTmp As QueryDef
    Dim rstTmp As Recordset
    Dim lenRecord As Integer
    Dim indI, indJ, i, PreNumiNode, CurNumiNode As Integer
    Dim strVOG, strFunction As String

    Set g_dbs = CurrentDb
```

Nœuds Ancestraux Internes

```
'Cération temporaire de regroupement d'espèces.
g_SQL = _
"SELECT [(import) Analysed VOG].File " _
& "FROM [(import) Analysed VOG];"

Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
Set rstTmp = qdfTmp.OpenRecordset(dbOpenSnapshot)

'Comptage du nombre d'enregistrements, et remplissage des enregistrements dans la matrice "VOGList".
rstTmp.MoveLast
lenRecord = rstTmp.recordCount
ReDim VOGList(1 To lenRecord, 1 To 5)
rstTmp.MoveFirst
For indI = 1 To UBound(VOGList, 1)
   VOGList(indI, cst_VOG) = rstTmp!File
   rstTmp.MoveNext
Next indI
rstTmp.Close
qdfTmp.Close

'Remplissage d'information concernant les Espèces, la Fonction, et les iNodes dans la liste "VOGList".
For indI = 1 To UBound(VOGList, 1)
   SetVOGInfo (indI)
Next indI

'Tri de la liste "VOGList".
ReDim VOGListSortedTmp(1 To lenRecord, 1 To 5)
i = 1
For indI = 0 To inode
   For indJ = 1 To UBound(VOGList)
      If VOGList(indJ, cst_NbOfiNode) = indI Then
         VOGListSortedTmp(i, cst_VOG) = VOGList(indJ, cst_VOG)
         VOGListSortedTmp(i, cst_NbOfSpecies) = VOGList(indJ, cst_NbOfSpecies)
         VOGListSortedTmp(i, cst_SpList) = VOGList(indJ, cst_SpList)
         VOGListSortedTmp(i, cst_Function) = VOGList(indJ, cst_Function)
         VOGListSortedTmp(i, cst_NbOfiNode) = VOGList(indJ, cst_NbOfiNode)
         i = i + 1
      End If
   Next indJ
Next indI

i = 1
For indI = 2 To UBound(VOGListSortedTmp) - 1
   If VOGListSortedTmp(indI, cst_NbOfiNode) <> VOGListSortedTmp(indI - 1, cst_NbOfiNode) Then
      i = i + 1                        'Select only those are different in iNode
   End If
Next indI

'DEBUG MODE : Ouverture de fichier pour écriture.
Open "C:\Documents and Settings\Owner\Desktop\Sujet de
Recherche\TRAVAIL\Version1\WorkSpace\Ancestors\VOGListSortedTmp.txt" For Output As #1    'Open file for output.
For indI = 1 To UBound(VOGListSortedTmp)
   Print #1, CStr(VOGListSortedTmp(indI, cst_NbOfiNode)) + "#" + VOGListSortedTmp(indI, cst_Function) + "#" +
VOGListSortedTmp(indI, cst_VOG)
Next indI
Close #1

ReDim VOGListSorted(1 To i, 1 To 5)
i = 1
For indI = 1 To UBound(VOGListSorted)
   VOGListSorted(indI, cst_NbOfiNode) = VOGListSortedTmp(i, cst_NbOfiNode)
```

Nœuds Ancestraux Internes

```
    strVOG = VOGListSortedTmp(i, cst_VOG)
    strFunction = VOGListSortedTmp(i, cst_Function)
    For indJ = i To UBound(VOGListSortedTmp) - 1

      If VOGListSortedTmp(indJ + 1, cst_NbOfiNode) = VOGListSortedTmp(indJ, cst_NbOfiNode) Then
        strVOG = strVOG + ";" + VOGListSortedTmp(indJ + 1, cst_VOG)
        strFunction = strFunction + ";" + VOGListSortedTmp(indJ + 1, cst_Function)
      Else
        Exit For
      End If
    Next indJ
    i = indJ + 1

    VOGListSorted(indI, cst_VOG) = LTrim(RTrim(strVOG))
    VOGListSorted(indI, cst_Function) = LTrim(RTrim(strFunction))

  Next indI

End Function

'#########################################################################################################
Function SetVOGInfo(indice As Integer)
  Dim qdfTmp As QueryDef
  Dim rstTmp As Recordset
  Dim lenRecord As Integer
  Dim aSpeciesList() As String
  Dim indI, indJ, incr, SpeciesInVOG, SpeciesIniNode As Integer
  Dim speciesList As String

  Set g_dbs = CurrentDb
  g_SQL = _
  "SELECT [(import) Genes Description].[VOGp Cluster], [(import) Genes Description].[VOGp Cluster Name], [Organisms
List].ShortOrgName " _
  & "FROM [Organisms List] INNER JOIN ([(import) Analysed VOG] INNER JOIN [(import) Genes Description] ON
[(import) Analysed VOG].File=[(import) Genes Description].[VOGp Cluster]) ON [Organisms List].Organism=[(import)
Genes Description].Organism " _
  & "GROUP BY [(import) Genes Description].[VOGp Cluster], [(import) Genes Description].[VOGp Cluster Name],
[Organisms List].ShortOrgName " _
  & "HAVING ((([(import) Genes Description].[VOGp Cluster])='" + VOGList(indice, cst_VOG) + "'));"

  Set qdfTmp = g_dbs.CreateQueryDef("", g_SQL)
  Set rstTmp = qdfTmp.OpenRecordset(dbOpenDynaset)

  ' Modification de données dans l'enregistrement local.
  speciesList = ""
  SpeciesInVOG = 0
  Do While Not rstTmp.EOF
    VOGList(indice, cst_Function) = rstTmp![VOGp Cluster Name]
    speciesList = speciesList + " " + rstTmp!ShortOrgName
    SpeciesInVOG = SpeciesInVOG + 1
    rstTmp.MoveNext
  Loop
  rstTmp.Close
  qdfTmp.Close
  g_dbs.Close

  VOGList(indice, cst_NbOfSpecies) = SpeciesInVOG
  speciesList = LTrim(RTrim(speciesList))
  VOGList(indice, cst_SpList) = speciesList
```

Nœuds Ancestraux Internes

```
    aSpeciesList = Split(speciesList)
    For indI = 0 To UBound(iNodeSorted, 1)

        'VOGList instance is a sub set (i.e. "<=") of iNodeSorted instance, i.e. a sub tree
        'L'instance VOGList est un sous ensemble (i.e. "<=") d'instance de iNodeSorted, i.e. un sous arbre
        SpeciesIniNode = iNodeSorted(indI, cst_NbOfSpecies)
        If SpeciesInVOG <= SpeciesIniNode Then
            incr = 0
            For indJ = 0 To UBound(aSpeciesList)
                If InStr(iNodeSorted(indI, cst_SpList), aSpeciesList(indJ)) > 0 Then
                    incr = incr + 1
                End If
            Next indJ

            If incr = SpeciesInVOG Then
                VOGList(indice, cst_NbOfiNode) = iNodeSorted(indI, cst_iNode)
                Exit Function
            End If

        End If
    Next indI
    'Debug.Print "Si cette étape est atteinte, cela signifie qu'il y a ERREUR (i.e Pas de iNode pour le VOG considéré) !!"
End Function
```