UNIVERSITÉ DU QUÉBEC À MONTRÉAL

FLUX DE TRAVAUX ET LEURS APPLICATIONS EN BIOINFORMATIQUE

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE

PAR

ETIENNE LORD

MAI 2015

REMERCIEMENTS

Il va sans dire que ce projet ne serait pas devenu ce qu'il est sans le concours de Vladimir Makarenkov, professeur au département d'informatique de l'UQAM, qui m'a accompagné dans mon travail durant plus de cinq ans. Bien plus qu'un superviseur, j'ai trouvé en lui un mentor qui m'a appris la qualité du travail bien fait. Son optimisme et son enthousiasme ont porté fruit dans plusieurs projets attachés à ce doctorat.

J'aimerais remercier particulièrement Abdoulaye Baniré Diallo, qui m'a apporté une perspective différente de la recherche tant au niveau pratique que philosophique, me faisant découvrir la culture africaine. Il m'a permis de souvent voir une perspective plus grande dans l'exécution de certains projets, de ne pas avoir peur d'oser et de voir grand! Aussi, merci à mes collaborateurs et amis du laboratoire de bioinformatique à l'Université du Québec à Montréal : Alix Boc, pour le côté positif des choses, Dunarel Badescu pour la diversité des horizons de recherche, Mickael Leclercq qui m'a montré que plusieurs façons de voir les choses étaient possibles. Ainsi que Nadia, Alpha Boubacar et tous les autres collaborateurs que j'ai côtoyés durant ces belles années.

Merci à ma grande amie Myriam qui m'a appris à poursuivre les rêves les plus fous tout en gardant les pieds sur terre. Merci à Alassane, qui m'a supporté durant cette rédaction.

Finalement, et plus important encore, merci à ma famille, mon petit frère et ma petite sœur ainsi que mes chers parents qui m'ont permis de poursuivre mon rêve et m'ont inspiré tant de questions. Merci encore à Jocelyne, Jean-Philippe et Raphaële qui m'ont soutenu durant cette démarche aux nombreux détours. En conséquence, merci à tous ceux qui m'ont inspiré : Asimov, van der Aalst, Fry, Maeda et Wilson…

TABLE DES MATIÈRES

LISTE DES FIGURES VIII
LISTE DES TABLEAUX XI
LISTE DES ABRÉVIATIONS XIII
ABSTRACT XV
RÉSUMÉ XVI
INTRODUCTION 1
CHAPITRE I
NOTIONS DE BASE 4
1.1 Phylogénie 4
1.2 Phylogénies modernes : un problème de taille 5
1.2.1 Méthodes de distance 7
1.2.2 Maximum de parcimonie 11
1.2.3 Maximum de vraisemblance 12
1.2.4 Méthodes bayésiennes 12
1.3 L'inférence de grandes phylogénies 13
1.4 Comparaison des phylogénies : distance de Robinson et Foulds 14
1.5 La phylogénomique 14
1.5.1 Méthode de super-matrices 15
1.5.2 Méthode de super-arbres 15
1.5.3 Méthode de méga-phylogénie 16
1.6 Données bioinformatiques 16
CHAPITRE II
FLUX DE TRAVAUX BIOINFORMATIQUES : ÉTAT DE L'ART 18
2.1 Introduction 18
2.2 Définition formelle d'un flux de travaux (ou *workflow*) 18
2.3 Répertoire de flux de travaux 20
2.4 Les plates-formes de flux de travaux 20
2.4.1 L'architecture des systèmes de gestion de flux de travaux 22
2.4.2 Données versus variables partagées 22
2.4.3 Exécution concurrente versus exécution séquentielle 22

2.4.4 Sémantique individuelle versus sémantique collective 23
2.4.5 Modèles d'exécution des flux de travaux scientifiques 23
2.4.6 *Control-flows* et *data-flows* 23
2.5 Systèmes de gestion des flux de travaux scientifiques 24
2.6 Plates-formes de flux de travaux en bioinformatiques 24
2.6.1 Kepler 28
2.6.2 Galaxy 29
2.6.3 Taverna 30
2.6.4 LONI 31
2.6.5 Triana 32
2.6.6 Bio-Jeti 33
2.7 Réutilisation des flux de travaux 34
2.8 Comparaison des flux de travaux 35
2.9 Mesure de distance entre flux de travaux 36
2.9.1 Distances Euclidienne, cosine et autres 37
2.9.2 Distance de graphes 38
2.10 Méthodes de regroupement 40
2.10.1 Méthodes de partitionnement 41
2.10.2 Méthodes hiérarchiques 45
2.11 Détermination du nombre de groupes 46
2.11.1 Indice de Calinski-Harabasz 47
2.11.2 Indice Silhouette 48
2.11.3 Indice LogSS 49
2.11.4 Autres indices de regroupement. 49
2.12 Comparaison des partitions : l'indice Rand 50
2.13 Conclusions 50
CHAPITRE III
LA PLATE-FORME ARMADILLO 51
3.1 Préface 51
3.2 Introduction 52
3.3 Conception et implémentation 58

3.3.1 Description générale de *Armadillo* 58
3.3.2 Applications incluses dans la version 1.1 de *Armadillo* 62
3.4 Exemple d'utilisation : Inférer des arbres phylogénétiques à l'aide de *Armadillo* ... 67
3.4.1 Étape I. Créer un jeu de données de l'adiponectine 67
3.4.2 Étape II. Génération d'alignements de séquences protéiques 70
3.4.3 Étape III. Inférer l'arbre phylogénétique de l'adiponectine 71
3.5 Conclusions 72
3.6 Perspectives 73
CHAPITRE IV
CLASSIFICATION DE FLUX DE TRAVAUX PAR ALGORITHMES DE PARTITIONNEMENT ET REGROUPEMENT HIÉRARCHIQUES 78
4.1 Préface 78
4.2 Résumé 79
4.3 Introduction 79
4.4 Regroupement de flux de travaux - revue de la littérature 81
4.5 Méthodes de partitionnement pour la classification des flux de travaux 83
4.6 Méthodes de classification hiérarchique de flux de travaux 90
4.7 Stratégies d'encodage des flux de travaux 91
4.7.1 Encodage de type I 92
4.7.2 Encodage de type II 92
4.7.3 Encodage de type III 93
4.7.4 Encodage de type IV 93
4.8 Classification des encodages à l'aides des méthodes de partitionnement 96
4.9 Classification des encodages à l'aide des méthodes de classification hiérarchique 105
4.10 Nouvelle mesure de support du regroupement par paires 110
4.11 Conclusions 116
4.12 Perspectives 118
CHAPITRE V
UTILISATION PRATIQUE DE LA PLATE-FORME ARMADILLO 120
5.1 Introduction 120
5.2 Prédiction de micro-ARNs associés aux stress abiotiques chez le blé 120
5.2.1 Description de l'étude 121

5.2.2 Flux de travaux utilisés 122
5.3 Portail Web WMP: *Wheat micro-RNAs portal* 130
5.3.1 Préface 130
5.3.2 Introduction 130
5.3.3 Contenu et statistiques de la banque de données 131
5.3.4 Interface utilisateur 132
5.3.5 Étude de cas 136
5.3.6 Conclusions 137
5.3.7 Perspectives de l'étude sur le blé 138
5.4 Étude de la pression sélective du VIH chez les femmes enceintes 139
5.4.1 Description de l'étude 140
5.4.2 Flux de travaux utilisés 142
5.4.3 Résultats et conclusions de l'étude 151
5.5 Conclusions 152

CHAPITRE VI
CONCLUSIONS ET PERSPECTIVES 154
6.1 Principales contributions 156

ANNEXE A
AUTRES FLUX DE TRAVAUX 159

ANNEXE B
TABLEAUX SUPPLÉMENTAIRES DU CHAPITRE 4 165

ANNEXE C
LOGICIELS BIOINFORMATIQUES PARALLÉLISÉS 170

ANNEXE D
SUPPLÉMENT SUR LE NOUVEAU CRITÈRE DE SUPPORT 172

BIBLIOGRAPHIE 182

LISTE DES FIGURES

Figure		Page
2.1	Exemple de flux de travaux permettant l'inférence d'un arbre phylogénétique	19
2.2	Vues des principales plates-formes de flux de travaux bioinformatiques	26
2.3	Flux de travaux dans *Triana*.	33
2.4	Regroupement hiérarchique et regroupement par partitionnement	41
3.1	Comparaison de quatre manières différentes d'effectuer une recherche de séquences	54
3.2	Aperçu de l'interface graphique (*GUI*) de la plate-forme *Armadillo*	60
3.3	Vue d'un flux de travaux après exécution dans la plate-forme *Armadillo*	61
3.4	Un exemple d'une solution bioinformatique réalisée à l'aide de la plate-forme *Armadillo*	67
3.5	Aperçu des différentes étapes necessaires pour inférer un arbre phylogénétique	69
3.6	Total des téléchargements de la plate-forme *Armadillo* depuis son lancement en 2012 et régions géographiques des téléchargements	74
3.7	Évolution du « *langage applicatif* » de *Armadillo* entre 2012 et 2014	75
3.8	Évolution d'une expérience *in silico* avec la plate-forme *Armadillo*	76
3.9	Évolution de la vie d'un flux de travaux	77
4.1	Cinq flux de travaux bioinformatiques créés à l'aide de la plate-forme *Armadillo*.	85
4.2	Les résultats de simulations obtenus pour les quatre stratégies d'encodage des flux de travaux	99

4.3	Résultats de la simulation étudiant l'évolution de l'indice Rand moyen pour l'ensemble de données de *Armadillo*	101
4.4	Résultats de la simulation étudiant l'évolution de l'indice Rand moyen pour l'ensemble de données de *myExperiment*	103
4.5	Résultats de classification combinés pour l'ensemble de flux de travaux de *Armadillo* et de *myExperiment* obtenus en utilisant les quatre types d'encodage avec et sans poids	104
4.6	Classification hiérarchique des stratégies de regroupement hiérarchiques des flux de travaux pour l'ensemble de données de *Armadillo*.	106
4.7	Classification hiérarchique des stratégies de regroupement hiérarchiques des flux de travaux pour le jeu de données de *myExperiment*	108
4.8	Résultats combinés de la classification hiérarchique obtenue pour les ensembles de données de *Armadillo* et de *myExperiment*..	109
4.9	Les résultats des simulations évaluant le comportement des indices de support par paires définies dans cette thèse.	115
5.1	Vue d'ensemble de l'étude phylogénomique sur les micro-ARNs du blé	122
5.2	Flux de travaux conceptuel permettant la recherche de micro-ARNs à partir de données de séquençage	125
5.3	Flux de travaux conditionnel de recherche de l'ontologie des gènes cibles du blé	128
5.4	Flux de travaux ayant servi à l'identification des séquences cibles à l'aide de la méthode *BLAST*	129
5.5	Flux de travaux utilisé dans le portail *Wheat micro-RNAs portal*	134
5.6	Vue d'ensemble des caractéristiques principales du portail Web	135
5.7	Vue d'ensemble de l'étude sur le VIH de type I chez les femmes enceintes	140
5.8	Exemples d'arbres phylogénétiques du gène *env* du VIH de deux patientes montrant l'évolution à différents stades de leur grossesse	143
5.9	Flux de travaux servant à la séparation des données	144
5.10	Flux de travaux permettant l'évaluation de la pression sélective	146

LISTE DES TABLEAUX

Tableau		Page
1.1	Les 20 familles de protéines et domaines les plus représentés dans la banque de données PFAM.	6
1.2	Méthodes de reconstruction phylogénétiques et principaux logiciels	7
1.3	Quelques types de données utilisés en bioinformatique et phylogénomique..	17
2.1	Caractéristiques des plates-formes de flux de travaux	21
2.2	Architecture de référence des plates-formes de flux de travaux scientifiques	24
2.3	Comparaison des principaux systèmes de flux de travaux en bioinformatique	27
2.4	Pratiques favorisant la réutilisation des flux de travaux	35
2.5	Complexité algorithmique des méthodes de regroupement	46
2.6	Critères de regroupement les plus communs	47
3.1	Applications bioinformatiques incluses dans la plate-forme *Armadillo*	64
3.2	Comparaison des caractéristiques de la plate-forme *Armadillo* v1.1 par rapport aux autrs plates-formes de flux de travaux bioinformatiques : *Taverna*, *Galaxy*, *LONI*, *Ergatis* et *Kepler*	65
4.1	Les quatre propositions d'encodage des flux de travaux et leurs vecteurs de poids associés pour les cinq flux de travaux bioinformatiques présentés à la Figure 4.1	95
4.2	Principales caractéristiques des flux de travaux réels provenant des jeux de données de *Armadillo* et de *myExperiment* considérés dans cette étude	97
4.3	Valeurs du support global du regroupement des flux de travaux, obtenues pour l'ensemble de données de *Armadillo*	114

5.1 Librairies et total des micro-ARNs candidats identifiés chez le blé dans l'étude de Agharbaoui *et al.* (2015) 121

5.2 Logiciels d'inférence des micro-ARNs candidats utilisés dans l'étude de Agharbaoui *et al.* (2015) 124

A.1 Sélection de flux de travaux inclus dans la plate-forme *Armadillo* 159

B.1 Le jeu de données *Armadillo* comprenant 120 flux de travaux et leur classe respective. 166

B.2 Le jeu de données *myExperiment* comprenant 100 flux de travaux et leur classe respective.. 168

C.1 Survol de quelques logiciels bioinformatiques parallélisés 170

D.1 Support moyen individuel (*PSG*±SD) des différentes espèces d'iris de la Figure D.2 en fonction du type du regroupement et de l'indice d'optimisation.. 175

D.2 Classes empiriques du jeu de données de *Zoo*.. 177

LISTE DES ABRÉVIATIONS

Abréviation	Signification
AA	Acides Aminés
ADN	Acide DésoxyriboNucléique
API	*Application Programming Interface*
ARN	Acide RiboNucléique
BLAST	*Basic Local Alignment Search Tool*
BPEL	*Business Process Execution Language*
BPEL4WS	*Business Process Execution Language for Web Services*
CH	Critère de Calinski-Harabasz (indice)
CSV	*Comma Separated Value* (format de fichier de données)
DDBJ	*DNA Data Bank of Japan* ((base de données de séquences)
EBI	*European Bioinformatics Institute* (base de données de séquences)
EST	*Expressed Sequences Tags* (petite portion d'un gène exprimée permettant de l'identifier)
HTML	*HyperText Markup Language* (langage de programmation web)
HUGO	*Human Gene Nomenclature Committee* (Base de données génétique)
LogSS	Critère de partitionnement LogSS (indice)
miARN	Micro-ARN (ou miRNA *en anglais*)
ML	Maximum de vraisemblance (*Maximum Likelihood*)
MP	Maximum de parcimonie
NCBI	*National Center for Biotechnology Information*
NGS	*Next Generation Sequencing* (séquençage de nouvelle génération)
NJ	*Neighbor-Joining* (algorithme de regroupement hiérarchique)

OS	Système d'opération d'un ordinateur (*p.ex.* Linux)
OTU	*Operational Taxonomic Unit*
pb	Paires de bases (*p.ex.* Gbp ou Gb, Gigabases)
PFAM	*Protein Family database* (base de données de protéines)
PUBMED	*Public US National Library of Medicine National Institutes of Health*
RF	Distance topologique de Robinson et Foulds
RFAM	*RNA Family database* (base de données contenant des ARNs non-codants)
SQL	*Structured Query Language* (langage de description des bases de données)
THG	Transfert Horizontal de Gènes (*HGT en anglais*)
UPGMA	*Unweighted Pair Group Method with Arithmetic Mean* (algorithme de regroupement hiérarchique)
URL	*Uniform Resource Locator*
VIH	Virus de l'immunodéficience humaine de type I
WFCA	*WorkFlow Creation Area* (espace de création du flux de travaux dans une plate-forme de flux de travaux)
WSDL	*Web Services Description Language*
WTSI	*Wellcome Trust Sanger Institute*
XML	*eXtented Markup Language*

ABSTRACT

This thesis focuses on the use of workflows for the conduct of bioinformatics research related to phylogenomics. Current researches often involve the use of scripting languages, which limit their reproducibility and are out of reach of smaller laboratories and organizations. Workflows are, by definition, repeatable patterns of linked tasks. Thus, they can be used to reproduce experimental condition in *in silico* experiments. This thesis will propose solutions to the following questions: (1) What is the correct level of abstraction for a workflow platform dedicated to the study of bioinformatics? (2) How can we compare different bioinformatics protocols, used in research, by using workflows? (3) What are the practical uses of workflows in bioinformatics research and more specially in phylogenetic analysis?

To answer these questions, we will first present a new workflow platform, *Armadillo*, adapted to the domain of phylogeny. Second, we will introduce a new classification procedure for workflows which use weighted *k*-means and *k*-medoids algorithms. A novel support criterion was also introduced to validate the membership of each workflow in its respective partition. Third, a number of experimental workflows developed and executed from the *Armadillo* platform will serve to display the use of this methodology when dealing with real phylogenomics data.

Keywords: Bioinformatics, workflow, clustering, *k*-means, *k*-medoids, workflow support criteria, phylogenetic analysis.

RÉSUMÉ

Cette thèse servira de modèle de démonstration de l'utilisation de flux de travaux (*workflows*) pour la conduite de recherche en bioinformatique et plus particulièrement en phylogénomique. La plupart des études actuelles utilisent des langages de programmation de type *scripts* pour la réalisation de ces études de grande envergure. Cependant, l'utilisation de cette approche limite la reproductibilité de ces études, en les mettant souvent hors d'atteinte pour les laboratoires n'étant pas spécialisés en bioinformatique. Les flux de travaux consistent en des patrons de tâches pouvant être répétés, permettant ainsi une reproduction des conditions expérimentales pour les expériences *in silico*. Cette thèse proposera des solutions aux questions suivantes : (1) Quel est le niveau d'abstraction requis pour une plate-forme de flux de travaux en bioinformatique? (2) Comment comparer et classifier des flux de travaux? (3) Quelles sont les applications pratiques des flux de travaux en bioinformatique, et particulièrement dans les études des phylogénies?

Pour répondre à ces questions, nous présenterons dans cette thèse une nouvelle plate-forme de flux de travaux, *Armadillo*, adaptée à l'analyse phylogénétique. Deuxièmement, une nouvelle stratégie de comparaison de flux de travaux et de leur classification à l'aide d'algorithmes de type *k*-means et *k*-medoids pondérés sera introduite. Un nouveau critère de support de chacun des flux de travaux dans cette classification a aussi été développé. Troisièmement, l'application de flux de travaux phylogénétiques conçus et exécutés à l'aide de la plate-forme *Armadillo* servira à illustrer l'utilité d'une telle plate-forme pour la recherche en phylogénomique.

Mots-clés : Analyse phylogénétique, bioinformatique, classification, critère de support de flux de travaux, flux de travaux, *k*-means, *k*-medoids.

INTRODUCTION

Nous traiterons dans cette thèse du problème grandissant de l'utilisation et de la comparaison des flux de travaux en bioinformatique. Cette étude de la similarité des flux de travaux nous amènera à la recherche de leur classification permettant : 1) leur réutilisation et 2) la dispersion de ceux-ci sur des grappes d'ordinateurs. Au chapitre 1, nous présenterons une brève introduction à l'analyse phylogénomique. Au chapitre 2, nous nous attarderons à présenter une revue de la littérature sur les flux de travaux et leur comparaison. Par la suite, nous présenterons les résultats de cette thèse. Nous répondrons alors aux questions suivantes :

1) Est-ce qu'une plate-forme de flux de travaux est pertinente pour la réalisation d'études phylogénomiques et bioinformatiques?

Pour répondre à cette question, nous présenterons au chapitre 3, une plate-forme de création de flux de travaux qui a été développée spécialement pour les études et les simulations phylogénomiques et a déjà fait l'objet d'une publication (Lord *et al.*, 2012).

2) Peut-on comparer des méthodologies de recherche en utilisant les flux de travaux? Pour ce faire, une étude de la classification des flux de travaux par différentes méthodes de partitionnement et méthodes hiérarchiques sera présentée au chapitre 4.

3) Quels sont les exemples pratiques de recherche pouvant être menées grâce aux flux de travaux? Comme démonstration, nous présenterons au chapitre 5, deux études dans lesquelles la plate-forme *Armadillo* fut utilisée. La première traitera du séquençage du génome du blé (*Triticum aestivum L.*) et de l'analyse de ce séquençage à l'aide de flux de travaux. Dans le deuxième cas, une étude du virus de l'immunodéficience humaine de type I (VIH) chez des femmes enceintes montrera l'utilisation de la plate-forme *Armadillo* dans le traitement de

génome du VIH et l'analyse de la pression sélective dans l'évolution de cette population de virus à travers les différentes phases de la grossesse.

Une conclusion de la thèse sera ensuite présentée au chapitre 6.

Cette thèse inclut ainsi le texte partiel ou complet de publications présentant la nouvelle plate-forme de flux de travaux ou d'études réalisées en utilisant celle-ci.

Chapitre 3

Lord, E., Leclercq, M., Boc, A., Diallo, A. B, Makarenkov. V. (2012). Armadillo 1.1: an original workflow platform for designing and conducting phylogenetic analysis and simulations. *PLoS One*, *7*(1):e29903.

J'ai contribué, avec Mickael Leclercq, à la conception du logiciel Armadillo. Ma contribution est de 75 % alors que celle de Mickael est de 25 %. Alix Boc a contribué à l'élaboration du concept et au soutien technique lors de la réalisation. Tout le travail a été réalisé sous la supervision de mes directeurs de recherche.

Chapitre 4

Lord, E., Diallo, A. B., et Makarenkov, V. (2015). Classification of bioinformatics workflows using weighted versions of partitioning and hierarchical clustering algorithms. *BMC Bioinformatics*, *16*(1), 68.

J'ai réalisé l'ensemble de la recherche, sous la supervision de mes directeurs de recherche.

Chapitre 5

Lord, E., Remita, M. A., Agharbaoui, Z., Leclercq, M., Badawi, M. A., Makarenkov, V., Sarhan, F., et Diallo, A. B. (2015b). WMP: Wheat MiRNA web-Portal. A novel comprehensive wheat miRNA database, including related bioinformatics software (soumis).

J'ai réalisé 25 % du travail d'analyse (flux de travaux) et 50 % de la mise en place du site web présentant les données et fournissant des services web. L'ensemble de la recherche a été réalisée sous la supervision de mes directeurs de recherche.

Une partie des résultats des articles suivants sera aussi présentée :

Agharbaoui, Z., Leclercq, M., Remita, M. A., Badawi, M. A., Lord, E., Houde, M., Danyluk, J., Diallo, A. B. et Sarhan, F. (2015). An integrative approach to identify hexaploid wheat miRNAome associated with development and tolerance to abiotic stress. *BMC Genomics 16*(1), 339.

Des flux de travaux réalisés par moi et Mickael Leclercq et la plate-forme Armadillo ont été utilisés dans cette étude. L'ensemble de la recherche a été réalisée sous la supervision de mes directeurs de recherche.

Ransy, D.G., Lord, E., Caty, M., Lapointe, N., Boucher, M., Diallo, A.B., Soudeyns, H. (2014). Subtle Differences in Selective Pressures Applied on the Envelope Gene of HIV-1 in Pregnant Versus Non-Pregnant Women. (soumis).

Le travail de recherche et l'analyse des données ont été réalisés par Doris Ransy. J'ai participé à l'analyse, la recherche des données et à leur présentation à l'aide de la plate-forme Armadillo. L'ensemble de la recherche a été réalisée sous la supervision de mes directeurs de recherche.

CHAPITRE I

NOTIONS DE BASE

«*Blast away at it! Get it done!*» – Williams J. Mitchell, MIT Media Lab

1.1 Phylogénie

Une phylogénie est une classification des espèces selon leurs liens de parenté (Darwin, 1857). L'étude de l'information génétique de différentes espèces pour former cette phylogénie est appelée l'analyse phylogénétique (Felsenstein, 2004). Les phylogénies modernes sont généralement représentées par des arbres phylogénétiques, et illustrent une hypothèse de l'évolution verticale des espèces. Dans un arbre phylogénétique, les feuilles terminales représentent les espèces contemporaines et les nœuds internes des ancêtres hypothétiques. La filiation est ainsi représentée par la connexion des nœuds par des arcs dont la longueur est fonction soit d'un temps de divergence évolutive (phylogramme), et/ou d'une relation de similarité entre les espèces (cladogramme) (Lecointre et Guyader, 2001). De plus, cette filiation est, par le degré de granularité, représentée comme binaire, *c.-à-d.* qu'un nœud ancestral pourra seulement être lié à deux descendants (Maddison, 1997; Dagan, 2011). En pratique, cette représentation de l'histoire évolutive d'organismes apparentés permet la recherche sur l'évolution de maladies infectieuses telle que le VIH (Romero-Severson *et al.*, 2014), les transferts génétiques entre espèces (Boc et Makarenkov, 2003) ou encore, de connaître la fonction de gènes ou de protéines, par homologie entre les espèces (Dinsdale *et al.*, 2008; Baughman *et al.*, 2011).

1.2 Phylogénies modernes : un problème de taille

Il est maintenant possible de séquencer un génome humain pour quelques milliers de dollars[1] (Pagani *et al.,* 2012). Ainsi, le prix du séquençage de 1 Megabase (1 000 000 pb) est passé de 5 200 $ à 6 cents entre 2001 et 2013. Ainsi, on peut maintenant séquencer en trois jours seize génomes humains, soit environ 1.8 terabases, pour un coût de 797$ par génome en excluant le coût du personnel (Hayden, 2014). Par conséquent, il est estimé que 15 petaoctets de nouvelles données génomiques sont générés chaque année. Présentement, l'information génomique recueillie représente 50 500 génomes partiellement séquencés[2] incluant 926 archéobactéries, 36 959 procaryotes et 8 624 eucaryotes (Pagani *et al.,* 2012). Sur *GenBank*, 260 000 séquences d'espèces uniques sont présentement recensées (Benson *et al.,* 2013), mais on estime que seulement 14 % des espèces ont été découvertes jusqu'à présent (May 1988; Mora *et al.,* 2011).

Cette diversité se traduit par d'énormes familles de protéines provenant d'espèces différentes. Ainsi, 19 familles et domaines protéiques contiennent plus de 80 000 séquences sur *PFAM* (Tableau 1.1), une base de données contenant 13 000 protéines. De plus, 9 familles contenant plus de 50 000 séquences d'ARN sont répertoriées sur *RFAM* (Burge *et al.,* 2013; Punta *et al.,* 2013).

Plus de 390 logiciels permettant l'étude phylogénomique et la reconstruction phylogénétique[3] sur ces génomes et séquences sont présentement disponibles. Pour ce faire, quatre approches principales (Tableau 1.2) sont employées en fonction de l'hypothèse évolutive, de la composition des données et de leur taille: les méthodes de distances (ou approche phénétique ne tenant pas compte des relations temporelles entre les espèces), les méthodes de maximum de parcimonie (approche cladistique basée sur la généalogie des espèces), les méthodes de maximum de vraisemblance, ainsi que et les méthodes bayésiennes (Philippe *et al.,* 2011).

---

[1] http://www.genome.gov/SequencingCosts/
[2] http://www.genomesonline.org/
[3] http://evolution.genetics.washington.edu/phylip/software.html - Joseph Felseinstein, Université de Washington, USA, Juillet 2014.

Tableau 1.1 Les 20 familles de protéines et domaines les plus représentés dans la banque de données *PFAM* (Punta *et al.*, 2013; vérifié le 15 Août 2014).

Descriptions	No. Accession *PFAM*	Types	Nombre de séquences	Longueur moyenne (*AA*)
ABC transporter	PF00005	Domaine	363 409	147.80
Cytochrome C and Quinol oxidase polypeptide I	PF00115	Famille	254 351	227.90
Zinc-finger double Domain	PF13465	Domaine	227 898	25.80
WD Domaine, G-beta repeat	PF00400	Répétition	193 252	38.20
Major Facilitator Superfamily	PF07690	Famille	181 668	295.20
Reverse transcriptase (RNA-dependent DNA polymerase)	PF00078	Famille	172 360	171.90
Binding-protein-dependent transport system inner membrane component	PF00528	Famille	156 339	195.90
Response regulator receiver domain	PF00072	Domaine	151 337	111.60
Envelope glycoprotein GP120	PF00516	Famille	146 453	228.30
Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	PF02518	Domaine	129 386	113.50
Retroviral aspartyl protease	PF00077	Domaine	124 336	94.30
Protein kinase domain	PF00069	Domaine	114 309	237.60
Ankyrin repeats (3 copies)	PF12796	Famille	110 723	89.50
Cytochrome b(N-terminal)/b6/petB	PF13631	Domaine	102 102	142.10
NADH-Ubiquinone/plastoquinone (complex I), various chains	PF00361	Famille	91 875	225.60
Helicase conserved C-terminal domain	PF00271	Famille	89 228	81.30
Bacterial regulatory helix-turn-helix protein, lysR family	PF00126	Domaine	86 700	59.70
LysR substrate binding domain	PF03466	Famille	85 941	202.80
short chain dehydrogenase	PF00106	Domaine	80 936	163.30
His Kinase A (phospho-acceptor) domain	PF00512	Domaine	79 834	67.20

Tableau 1.2 Méthodes de reconstruction phylogénétiques et principaux logiciels associés. Le nombre taxa pouvant être traités par ceux-ci est indiqué entre parenthèses[4].

	Méthodes basées sur les caractères	**Méthodes sans caractère**
Méthodes avec modèles d'évolution	*Maximum de vraisemblance :* fastDNAml (~100 taxa) PhyML (~200 taxa) RAxML (~10 000 taxa) FastTree2 (>237 000 taxa) *Méthodes bayésiennes* : BEAST (~100 taxa) MrBayes (~200 taxa)	*Méthodes de distance :* Neighbor (~1000 taxa) BioNJ (> 1000 taxa) RapidNJ (~13 000 taxa) Ninja (~100 000 taxa)
Méthodes sans modèle d'évolution	*Maximum de parcimonie* : DNAPars (~500 taxa) PROTPars (~500 taxa) TNT (~70 000 taxa) Oblong (>1 000 000 taxa)	

1.2.1 Méthodes de distance

Les méthodes par mesure des distances calculent la similarité entre des paires de séquences (Cavalli-Sforza et Edwards, 1967; Fitch et Margoliash, 1967). Si ces distances sont suffisamment proches du nombre d'évènements évolutionnaires entre les espèces, *c.-à-d.* les substitutions (transversions et transitions entre les différents nucléotides), l'arbre inféré est alors représentatif (Kim et Warnow, 1999). Cependant, les substitutions multiples se produisant au même site ne sont pas prises en compte lors du calcul des paires distances. Ceci peut être corrigé à l'aide de modèles d'évolution qui représentent la probabilité de

[4] Les références pour chacun des logiciels sont données dans les sections suivantes.

substitution entre les différents nucléotides. Parmi les modèles d'évolution les plus populaires, on retrouve le modèle de Jukes-Cantor (JC69) présentant des fréquences de changements égales entre toutes les paires de bases (Jukes et Cantor, 1969), le modèle d'Hasegawa-Kishino-Yano (HKY85) présentant des taux de transitions et transversions pondérés en fonction de la fréquence des nucléotides (Hasegawa *et al.*, 1985) et, finalement, le modèle *General Time Reversive* (GTR) prenant en compte que les taux de substitutions sont différents ainsi qu'une fréquence inégale des différents nucléotides (Lanave *et al.*, 1984). Dans le cas de séquences protéiques, des modèles de substitutions empiriques ont générés en se basant sur les fréquences retrouvées dans des banques de séquences tels que les modèles PAM (Dayhoff *et al.*, 1972) et BLOSUM (Henikoff et Henikoff, 1992). Le modèle le plus récent, le modèle Jones, Taylor, et Thornton (JTT), a été élaboré en employant la méthode de Dayhoff sur des alignements de séquences plus volumineux (Jones *et al.*, 1992; voir Whelan et Goldman, 2001).

Une fois la matrice de distances obtenue, des techniques de regroupement telles que Unweighted Pair Group Method with Arithmetic Mean (UPGMA; Sokal et Michener, 1958), Neighbor-Joining (NJ; Saitou et Nei, 1987), FITCH et KITSCH (Fitch et Margoliash, 1967) sont utilisées pour reconstruire une hiérarchie des espèces. Les méthodes UPGMA et KITSCH infèrent une classification ultramétrique des espèces, tandis que les algorithmes NJ et FITCH entraînent la création d'arbres additifs (Felsenstein, 2004). Nous détaillerons certaines de ces techniques, car elles seront utilisées lors de la classification des flux de travaux.

1.2.1.1 UPGMA

La méthode UPGMA (Sokal et Michener, 1958) est un algorithme agglomératif qui produit un arbre ultramétique *c.-à-d.* que toutes les distances entre les feuilles et la racine sont égales. Une horloge moléculaire avec un temps constant est alors assumée (Felsenstein, 2004). Dans cette méthode, les deux groupes de séquences les plus proches sont amalgamés à chaque itération de l'algorithme, créant un nouveau nœud dans la hiérarchie. Ainsi, cette méthode construit l'arbre de bas en haut, en calculant la distance d_{ij} entre deux groupes C_i et C_j comme étant la moyenne pondérée des distances entre les paires de séquences de chaque groupe :

$$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} d_{pq}. \qquad (1)$$

Dans cette dernière équation (Équation 1), $|C_i|$ et $|C_j|$ représentent le nombre de séquences dans les groupes i et j respectivement. Le nouveau groupe créé par l'union de C_i et C_j est alors appelé C_k et sa distance jusqu'au groupe C est la suivante (Équation 2) :

$$d_{kl} = \frac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}. \qquad (2)$$

L'algorithme est alors comme suit :

Algorithme UPGMA

Initialisation :

Assigner chaque séquence i à son propre groupe C_i

Définir une feuille à l'arbre T pour chaque séquence et la placer à la hauteur de 0

Itération :

Déterminer les deux groupes C_i et C_j où la distance d_{ij} est minimale

Définir un nouveau groupe k et calculer la distance d_{kl} pour tous les autres groupes

Définir un nouveau nœud k de hauteur $d_{ij}/2$ avec comme enfants les nœuds i et j

Ajouter k au nouveau groupe C_k et enlever i et j de la liste

Terminaison :

Lorsqu'il ne reste que deux groupes, placer la racine de l'arbre à la distance $d_{ij}/2$

1.2.1.2 Neighbor-Joining

L'algorithme de Neighbor-Joining, contrairement à l'algorithme UPGMA, n'assume pas une horloge moléculaire. Il suit le principe d'évolution minimum, menant à une minimisation de la taille totale de l'arbre additif (Saitou et Nei, 1987). Dans cet algorithme, la distance d_{ij} entre deux nœuds i et j est définie comme étant la distance minimale entre ceux-ci et les autres feuilles de l'arbre (Durbin *et al.*, 2006). Soit $|L|$ le nombre de feuilles dans l'ensemble de feuilles L, on retrouve (Équations 3 et 4) :

$$D_{ij} = d_{ij} - (r_i + r_j), \qquad (3)$$

$$r_i = \frac{1}{|L|-2}\sum_{k\in L} d_{ik} \,. \qquad (4)$$

L'algorithme est alors comme suit :

Algorithme Neighbor-Joining

Initialisation:

Définir T comme étant un arbre contenant comme feuilles (*nœuds*) toutes les séquences et définir $L = T$

Itération :

Prendre une paire i et j dans L où la distance D_{ij} est minimale

Définir un nouveau nœud k de distance $d_{km} = ½ (d_{im} + d_{jm} - d_{ij})$ pour tous les m dans L

Ajouter k à T avec des longueurs de branches : $d_{km} = ½ (d_{im} + d_{jm} - d_{ij})$, joignant ainsi k à i et j respectivement

Enlever i et j de L et y ajouter k

Terminaison :

Lorsque L contient seulement deux feuilles i et j, ajouter une branche entre i et j de longueur d_{ij}.

Les implémentations les plus connues de ce principe sont le logiciel Neighbor du *package* PHYLIP (Felsenstein, 2006) et BioNJ dont l'algorithme est différent mais utilise le même principe (Gascuel, 2007). Ce dernier logiciel tenant compte dans l'estimation des distances de différents modèles évolutifs. De même, des logiciels tels que RapidNJ (Simonsen *et al.*, 2008) et NINJA (Wheeler, 2009), utilisant des heuristiques de recherche, permettent maintenant l'utilisation de cette méthode sur des jeux de données de plus de 100 000 séquences.

1.2.1.3 FITCH et KITSCH

Les algorithmes de Fitch et Kitsch, selon la méthode de Fitch et Margoliash (1967), utilisent une fonction objective des moindres carrés pour trouver l'arbre présentant la distance minimale globale dans l'arbre inféré. Dans cette méthode, une matrice M^T contenant les $(\frac{n(n-1)}{2} \times 2n-3)$ chemins incidents (*path edge incidence matrix*) entre les n feuilles considérées (ou espèces), est construite. Ces chemins sont permutés lors de la recherche

heuristique du meilleur arbre. Par la suite, cette matrice est jumelée à un vecteur de poids $\vec{e}$ contenant $2n$-3 poids e_i correspondant à ces chemins (*c.-à-d.* représentant les longueurs de arêtes (i = 1, ..., $2n$-3) de l'arbre). Alors, le vecteur $\vec{d}^T$ contenant les distances entre les feuilles de l'arbre T est défini tel que $\vec{d}^T = M^T \vec{e}$. La topologie d'arbre recherchée (Fitch et Margoliash, 1967) correspond alors à la topologie dans laquelle l'erreur E (Équation 5), représentant l'erreur relative au carré entre le vecteur de distances $\vec{d}^M$ de la matrice M^T et le vecteur de distances $\vec{d}^T$, est minimisée (Équation 5) :

$$E = \left\| \vec{d}^T - \vec{d}^M \right\|^2 = \sum_{i<j} \frac{| d_{ij}^T - d_{ij}^M |^2}{(d_{ij}^M)^p} \rightarrow \min . \qquad (5)$$

Dans le cas des méthodes de Fitch et Kitsch, la valeur de l'exposant p est ainsi de 2 (Felsenstein, 1984).

L'algorithme peut alors être défini comme suit :

Algorithme Fitch

Initialisation :

Assigner chaque séquence i à son propre groupe C_i

Définir une feuille à l'arbre T pour chaque séquence et la placer à la hauteur de 0

Itération :

Joindre les séquences i et j dont la distance d_{ij} est minimale

Pour toutes les séquences n'étant pas i et j, calculer la distance des arrêtes de l'arbre $T((i,j), x)$

Recalculer les distances moyennes pour toutes les paires de groupes, telles que $d_u = i \cup j, k$ (pour chaque k)

Trouver les nouvelles longueurs de branches minimisant l'erreur E (Équation 5)

Terminaison :

Arrêter lorsqu'il ne reste qu'un seul groupe

1.2.2 Maximum de parcimonie

Les méthodes de maximum de parcimonie recherchent l'arbre phylogénétique qui correspond au nombre minimum de substitutions requises pour expliquer les différences dans

l'alignement de séquences. La méthode de Fitch (ci-haut, Fitch et Margoliash, 1967) permet la réversibilité dans ces substitutions nucléotidiques, tandis que des méthodes telles que Dollo (Farris, 1977) ne le permettent pas. Les logiciels les plus connus utilisant cette approche sont DNAPars et PROTPars du *package* PHYLIP (Felsenstein, 2006). De plus, les logiciels TNT (Goloboff *et al.,* 2008) et Oblong (Goloboff, 2014) permettent l'utilisation de cette méthode sur des jeux de données volumineux.

1.2.3 Maximum de vraisemblance

Les méthodes de maximum de vraisemblance, introduites par Felsenstein, assignent une probabilité aux différents évènements de mutation alors que le calcul de ceux-ci est comptabilisé de manière similaire aux méthodes de maximum de parcimonie. Le concept de cette approche est de générer tous les arbres phylogénétiques possibles en se basant sur un alignement de séquence original, et de calculer par la suite leur habileté à prédire la séquence observée (Felsenstein, 2004). L'arbre présentant la plus grande probabilité est celui conservé. Les techniques de maximum de vraisemblance ont une complexité algorithmique élevée due au nombre d'arbres qui doivent être générés. Par exemple, pour 23 espèces, il y a $1.32x10^{25}$ arbres phylogénétiques non enracinés possibles (Felsenstein, 2004), ce qui représente plus de dix fois la quantité d'étoiles dans l'univers (Dao *et al.,* 2013). Cette réalité limite le nombre de taxa ou de sites pouvant être utilisés dans l'inférence phylogénétique (Goloboff, 2014). Les logiciels PAUP (Swofford, 2002), PhyML (Guindon *et al.,* 2010), RAxML (Stamatakis, 2006), FastTree2 (Price *et al.,* 2010) sont présentement les plus utilisés et permettent la recherche de milliers de taxa. De même, les logiciels DNAML et PROTML, du *package* PHYLIP (Felsenstein, 2006), ainsi que le logiciel fastDNAml (Olsen *et al.,* 1994) incluent des heuristiques permettant une recherche pour des centaines de taxa, ce qui les rend comparables aux méthodes de maximum de parcimonie.

1.2.4 Méthodes bayésiennes

Les méthodes bayésiennes (Rannala et Yang, 1996) sont des méthodes conceptuellement différentes des méthodes de maximum de vraisemblance et de maximum de parcimonie. Celles-ci ne recherchent pas seulement le meilleur arbre, mais plutôt une distribution d'arbres. On maximise la probabilité *a posteriori* qui est proportionnelle à la vraissemblance

multipliée par la probabilité *a priori* de cette hypothèse de départ (Guindon et Gascuel, 2003). Cette probabilité postérieure représente un taux de confiance de la phylogénie inférée. Ces méthodes requièrent pour fonctionner un *a priori* sous la forme d'un modèle d'évolution ou encore une topologie de départ (Ronquist *et al.*, 2009). Contrairement aux méthodes de maximum de parcimonie ou de vraisemblance, les méthodes bayésiennes reposent sur un algorithme qui ne cherche pas forcément à trouver le point le plus haut dans l'espace des valeurs des paramètres (*optimum global*). Ainsi, elles sont plus rapides, mais requièrent plus d'itérations pour obtenir un résultat statistiquement intéressant (Guindon et Gascuel, 2003). Les logiciels implémentant cette approche incluent MrBayes (Huelsenbeck et Ronquist, 2001; Ronquist *et al.*, 2012) et BEAST (Drummond et Rambaut, 2007).

1.3 L'inférence de grandes phylogénies

Peu d'études ont porté sur l'inférence de phylogénies de plus de mille espèces (Goloboff *et al.*, 2009). Deux types de grandes phylogénies s'opposent : 1) la recherche du nombre maximum d'espèces dans une phylogénie (Goloboff *et al.*, 2009); 2) l'utilisation d'un nombre maximal de caractères (*c.-à-d.* de sites) pour augmenter la précision de l'arbre phylogénétique inféré (Dunn *et al.*, 2008; Philippe *et al.*, 2011).

Les logiciels actuels de maximum de vraisemblance RAxML (Stamatakis, 2006) et de méthodes bayésiennes MrBayes (Ronquist *et al.*, 2012) sont capables de prendre en compte entre 100 et 10 000 taxa (Dao *et al.*, 2013). Les méthodes de maximum de parcimonie implémentées dans les logiciels TNT (Goloboff *et al.*, 2008) et Oblong (Goloboff, 2014) peuvent prendre en compte des phylogénies comptant plus de 70 000 taxa (Goloboff *et al.*, 2009) et 30 millions de sites distincts (pour 50 taxa).

Des exemples de grandes phylogénies incluent la classification d'environ 13 000 espèces de plantes par Smith *et al.* (2009), en considérant le gène de la ribulose-bisphosphate carboxylase (rbcL). En 2009, Wheeler a utilisé NINJA pour construire des arbres de plus de 50 000 séquences de l'ADN polymérase ARN-dépendante et du transporteur ABC (Wheeler, 2009). La même année, Goloboff a analysé la phylogénie de 73 060 eucaryotes en considérant une combinaison de caractères issue d'ARN ribosomal, de protéines et d'ADN.

Plus récemment, 120 000 séquences d'hyménoptères (*insectes*) ont été analysées en utilisant un pipeline personnalisé (Peters *et al.*, 2011).

1.4 Comparaison des phylogénies : distance de Robinson et Foulds

Pour comparer deux phylogénies, une méthode de comparaison des topologies d'arbres est généralement utilisée (Robinson et Foulds, 1981). Cette méthode calcule une distance d'édition d'arbres permettant de mesurer le nombre de fusions et de fissions de nœuds nécessaires pour transformer un arbre T_1 en arbre T_2. Plus formellement, la distance des Robinson et Foulds (RF) entre deux arbres contenant les mêmes espèces est définie comme une mesure normalisée du nombre de bipartitions induites présentes dans un arbre mais absentes dans l'autre arbre (Équation 6). Dans l'équation 6, la notation *B(T)* représente le jeu de bipartitions non-triviales d'un jeu de données induit par un arbre (T_1) mais non par l'autre arbre (T_2) (Pattengale *et al.*, 2007). Cette mesure de dissimilarité est une métrique (Robinson et Foulds, 1981) et peut être estimée par une heuristique en temps sub-linéaire (Pattengale *et al.*, 2007), ou calculée en temps linéaire ou quadratique avec préordonnancement des données (Day, 1985) ou non (Makarenkov et Leclerc, 2000).

$$d_{RF}(T_1,T_2)=\frac{1}{2}(\left|B(T_1)-B(T_2)\right|+\left|B(T_2)-B(T_1)\right|) \qquad (6)$$

1.5 La phylogénomique

La phylogénomique est un champ de recherche qui débuta dans les années 90 (Pagel 1999). Utilisant les principes de la phylogénétique, on se base cette fois sur la totalité de l'information génomique (Rokas et Holland, 2000). La phylogénomique a ainsi pour objectif d'augmenter la *prédiction* fonctionnelle des gènes en utilisant la totalité de l'information de plusieurs génomes (Eisen, 1998). On émet, dans ce cas, l'hypothèse que la distribution des gènes dans ces génomes provient de pressions sélectives multiples et que cette méthodologie de recherche permettra une meilleure résolution de l'arbre final d'espèces (Leigh *et al.*, 2011; Swofford *et al.*, 1996). Aujourd'hui, cette augmentation des données biologiques (Stamatakis *et al.*, 2007; Zhang *et al.*, 2011) rend possible l'inférence d'arbres phylogénétiques de milliers de taxa (Stamatakis, 2006). La reconstruction phylogénomique repose actuellement

sur trois méthodologies : la méthode de super-matrices, la méthode de super-arbres et la méthode de méga-phylogénies.

1.5.1 Méthode de super-matrices

Premièrement, la méthode de *super-matrices*, implique la concaténation de séquences en un long alignement où les éléments non présents, par exemple un gène non séquencé, sont traités comme des données manquantes. Dans ce cas, des *gaps* sont ajoutés dans l'alignement final pour prendre en compte ces données manquantes. Les méthodes de reconstruction par maximum de vraisemblance sont adaptées à ce genre de données puisqu'elles considèrent l'hétérogénéité entre les différentes parties de l'alignement si un modèle de partition est utilisé (qui prend en compte des taux d'évolution hétérogènes dans les différentes parties de l'alignement). Cette approche est assez robuste et peut prendre en considération de 12.5 à 25 % de données manquantes (revue pas Philippe *et al.*, 2011). L'approche de super-matrices est ainsi souvent utilisée, car elle permet d'assembler de larges jeux de données et permet aussi de ne pas attendre la fin du séquençage d'un génome pour analyser la phylogénie d'une espèce.

1.5.2 Méthode de super-arbres

La deuxième approche, l'approche de *super-arbres* fait plutôt appel à la génération de plusieurs arbres (un par gène ou partie de gène) qui sont combinés avec une certaine valeur de support lors de l'inférence de l'arbre phylogénétique final. L'approche de combinaison la plus fréquente est l'addition par parcimonie de chacun des arbres (aussi appelée approche par matrice de parcimonie) (Bininda-Emonds, 2004; Dao *et al.*, 2003). D'autres approches de combinaisons, telles que l'évaluation du support par la distance de Robinson et Foulds (Robinson et Foulds, 1981), sont aussi possibles lors du choix de conservation de la topologie des différents taxa (Bansal *et al.*, 2010). Cette méthode a d'ailleurs été utilisée pour l'étude des mammifères placentaires (Liu *et al.*, 2001). Cependant, pour l'instant, cette technique a majoritairement été utilisée sur des données provenant de procaryotes (Philippe *et al.*, 2011), principalement parce que ces organismes ne contiennent qu'un faible nombre de gènes et que ces gènes sont de petite taille. Elle a, de plus, le désavantage de devoir évaluer un modèle d'évolution propre pour chacun des arbres de gènes inférés.

1.5.3 Méthode de méga-phylogénie

La troisième approche est une approche plus récente nommée *méga-phylogénie* (Smith *et al.*, 2009). Cette approche repose sur la création d'une super-matrice de sites, mais en incluant seulement certaines portions des gènes ou protéines orthologues (Smith *et al.*, 2008). Ces régions sont choisies par une méthode d'analyse de la saturation des sites, et diffèrent des méthodes d'alignement par le fait que les régions difficilement réconciliables sont conservées en utilisant une technique de réalignement progressive des sites, plutôt que l'exclusion automatique de ces régions. Un exemple de cette approche est la comparaison de 4 657 espèces de plantes comprenant 22 391 sites (Smith et Donoghue, 2008).

En conclusion, il existe encore une controverse sur la méthode permettant de réaliser les études phylogénomiques les plus robustes (Ren *et al.*, 2009; Philippe *et al.*, 2011; Lapointe et Rissler, 2005).

1.6 Données bioinformatiques

Les données bioinformatiques actuellement accessibles proviennent de sources multiples et se retrouvent sous plusieurs formats selon leur type et leur provenance (Goderis *et al.*, 2005; Goodman *et al.*, 2014). La plupart des logiciels permettant le décodage et l'utilisation de ces données sont des logiciels spécialisés, créés par des chercheurs, dont le support des données n'est pas la priorité (Sharma *et al.*, 2013).

Un aperçu des différents formats et des tailles de données requis pour la conduite d'une expérimentation *in silico* en phylogénomique est présenté au Tableau 1.3. On remarque la très grande disparité entre les différentes tailles de données nécessaires à une analyse.

Tableau 1.3 Quelques types de données utilisés en bioinformatique et phylogénomique.

Principaux types de données	**Formats***	**Tailles approximatives lors d'une étude phylogénomique**
Séquences	Fasta, Phylip, PSL	1 Kb ~ 10 Mb
Génomes	HAL, MAF, .2bit, .nib,	10 Mb ~ 1 Tb
Arbres phylogénétiques	Nexus, Newick, PhyloXML	1 Kb ~ 1 Mb
Annotations des séquences	BED, GFF, GTF, WIG, GenePred, Personal Genome SNP, VCF	1 Mb ~ 1 Tb
Séquençage nouvelle génération (*NGS*)	BAM, bigWIG, Fastq, ENCODE	0.5 Gb ~ 1 Pb

* N'inclus pas les variantes individuelles de formats propres à chaque logiciel.
Kb : Kilooctet, Mb : Megaoctet, Gb : Gigaoctet; Tb : Teraoctet; Pb : Petaoctet;

1.7 Conclusions

Le problématique sous-jacente aux méthodes phylogénétiques implique : 1) la surabondance de méthodes et de formats, 2) l'application de ces méthodes de phylogénomique à plusieurs jeux de données, 3) la reprise d'une analyse lorsque de nouveaux taxa (espèces, gènes, protéines) sont disponibles. Dans le prochain chapitre, nous explorerons cette problématique avec l'introduction des flux de travaux permettant une sérialisation des protocoles de recherche en vue de leur exécution.

CHAPITRE II

FLUX DE TRAVAUX BIOINFORMATIQUES : ÉTAT DE L'ART

> «*It should be trivial for a young Ph.D. researcher in chemistry to ask a computer: Locate 100,000 molecules that are similar to the known HIV protease inhibitors, compute their electronic properties, and dock them into viral escape mutant.*» – Savas Parastatidis, Microsoft

2.1 Introduction

Les flux de travaux et les plates-formes de flux de travaux, permettant l'orchestration et l'exécution de ceux-ci, sont d'abord des intégrateurs de plusieurs outils facilitant la communication entre les différentes applications, en encapsulant la conversion de différents formats de données. Ils automatisent, en plus, l'accès à des bases de données biologiques distantes permettant aux chercheurs une abstraction sur la provenance des données tout en en conservant, cependant, la trace d'exécution (Goderis, 2008). Finalement, ils permettent la répétition des expériences *in silico* (Ioannidis *et al.*, 2009; Zhao *et al.*, 2012)

2.2 Définition formelle d'un flux de travaux (ou *workflow*)

Un flux de travaux, aussi appelé gestion de flux ou « *workflow* », est une représentation formelle de l'ordre d'exécution d'un ensemble de tâches définies, permettant de mieux comprendre leur interdépendance et d'obtenir un résultat (van der Aalst et van Hee, 2004). L'objectif de cette représentation est de faire abstraction de la complexité des tâches sous-jacentes, et de permettre ainsi la conception de patrons d'analyse par et pour des non-spécialistes, ou encore de bien comprendre le processus complet menant à la production d'un résultat (Weerawarana *et al.*, 2005).

Plus spécifiquement, un flux de travaux (Figure 2.1) est défini comme un graphe orienté acyclique (*DAG*) (van der Aalst et van Hee, 2002). Ce graphe $W(T, E)$ contient une série de tâches $T=\{t_1, \ldots, t_i\}$ et comprend une source $(s) \in W$, un puits $(p) \in W$, et un ensemble de transitions $E = \{e_1, ..., e_j\}$ de tel sorte que que chaque transition e se retrouve dans un chemin de s à p (Rahman *et al.*, 2013). Dans le cas de flux de travaux devant être exécutés, on peut ajouter à cette définition un ensemble de ressources computationnelles disponibles $R = \{r_1, \ldots, r_n\}$ et une série de propriétés $M = \{m_{i,1}, \ldots, m_{i,k}\}$ pour chacune des tâches t_i (van der Aalst et Hee, 2004). Cependant, contrairement aux graphes, un flux de travaux peut contenir plusieurs types de tâches soit : 1) des tâches représentant des opérations ou 2) des tâches représentant des entrées et sorties. Brièvement, un flux de travaux est un patron de tâches ordonnées pouvant être exécutées de manière répétitive. En informatique, différents langages ont utilisé les flux de travaux pour simplifier la programmation sous une forme visuelle, tels que: Fabrik, InterCons, LabVIEW, viz, DataVis, etc. (voir Hils, 1992).

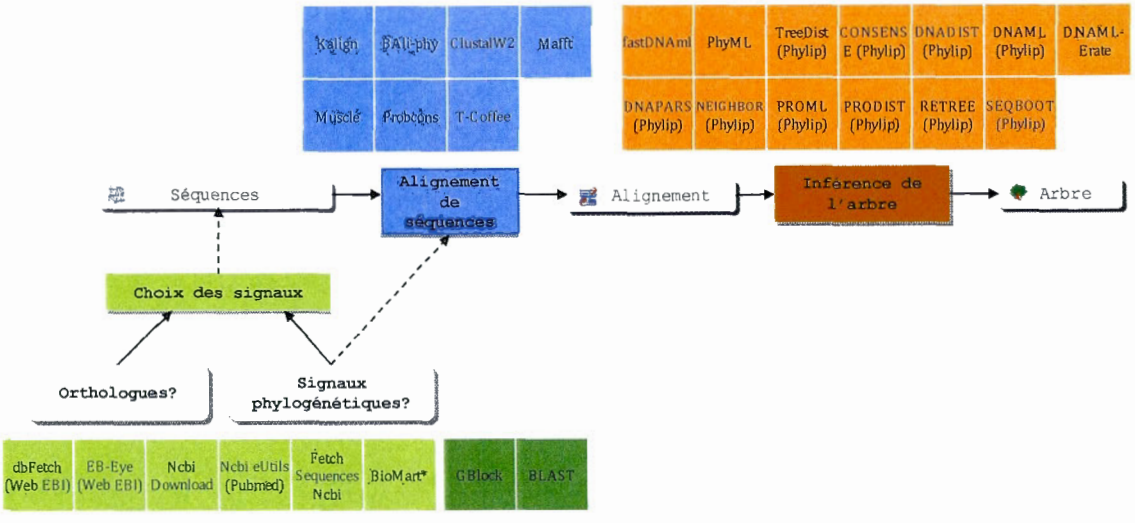


Figure 2.1 Exemple de flux de travaux permettant l'inférence d'un arbre phylogénétique. Différentes méthodes et sources de données sont disponibles à chaque étape.

2.3 Répertoire de flux de travaux

Le portail Web *myExperiment*[5] est une banque de données de flux de travaux orientée vers la bioinformatique (Goble *et al.*, 2010). Il contient présentement 2 230 flux de travaux publics (Mai 2014) provenant de 14 plates-formes de flux de travaux telles que Taverna, LONI, Kepler et BioExtract (Lushbough *et al.*, 2008 et 2011). Pour l'inférence des phylogénies, seulement 11 flux de travaux sont répertoriés sur le site *myExperiment*. De plus, peu de publications font état de l'utilisation de flux de travaux dans la recherche. Plus récemment, un nouveau portail de flux de travaux a été créé pour supporter les recherches en biodiversité. Le portail privé www.*biovel.eu* (DonVito *et al.*, 2012) permet à des utilisateurs enregistrés de consulter des flux de travaux portant sur les domaines de la phylogénétique, de l'évolution des populations et de la métagénomique (Vicario *et al.*, 2011). Actuellement, 71 flux de travaux sont disponibles et ont été évalués en fonction de leur structure dans une publication (Cohen-Boulakia *et al.*, 2014).

2.4 Les plates-formes de flux de travaux

Il existe une grande confusion au niveau des plates-formes de flux de travaux et sur les flux de travaux dans la littérature. On appelle présentement flux de travaux, dans la littérature, tout ensemble de *scripts* ou logiciels exécutés en *pipelines* permettant de prendre en entrée un certain nombre de fichiers et de sortir finalement des résultats sous forme graphique ou de fichiers. En bioinformatique, par exemple, on peut parler du logiciel *ESTPiper* (Tang *et al.*, 2009), permettant l'annotation de séquences, de *Phylogena* (Hanekamp *et al.*, 2007), de *Phylemon* (Sánchez *et al.*, 2011), de *BioExtract server* (Lushbough *et al.*, 2008 et 2011), de *Phylogeny.fr* (Dereeper *et al.*, 2008), et de *UGENE* (Okonechnikov *et al.*, 2012), qui permettent la création automatisée de plusieurs phylogénies comprenant un ou plusieurs patrons d'analyse. La plate-forme *Anvaya* (Limaye *et al.*, 2012), avec une implémentation de type client-serveur permettant de distribuer des tâches en utilisant un serveur *Torque*, peut, elle aussi, traiter des données génomiques et effectuer une série de tâches reliées à la bioinformatique. De même, *Agalma* (Dunn *et al.*, 2013), est un système d'exécution de

[5] http://www.myexperiment.org

scripts pour la phylogénétique, programmé en *Java*. Ce système est bâti comme une couche superposée au moteur *BioLite* (Howison *et al.*, 2012). Ce dernier correspond à une suite de *scripts* en langages *Python* et *C++* permettant l'exécution de tâches distribuées.

Ces plates-formes et applications répondent à la définition de la *Workflow Management Consortium* (*WfMC*) pour ce qu'est une plate-forme de flux de travaux (Hollingsworth, 1995). On parle ainsi de plate-forme de gestion de flux de travaux pour désigner les systèmes de gestion de flux de travaux permettant la création et la mise en œuvre des flux de travaux. Ces plates-formes permettent d'avoir un environnement qui définit et contrôle la coordination des processus et des applications (Beaulah *et al.*, 2008). De plus, ces plates-formes facilitent l'application ou l'exécution du flux de travaux sur une plus grande échelle (plate-forme distribuée) (Callaghan *et al.*, 2010). Ainsi, Woollard *et al.* (2008) ont identifié cinq objectifs que devraient rencontrer les systèmes de gestion de flux de travaux pour encourager leur utilisation (voir ci-bas, Tableau 2.1).

Tableau 2.1 Caractéristiques des plates-formes de flux de travaux.

Caractéristiques (adapté de Woollard *et al.*, 2008)
• Permettent l'automatisation des tâches pouvant produire des erreurs
• Permettent l'intégration de l'analyse et la visualisation des données
• Permettent la collection, l'organisation et la réorganisation des données (transformation vers d'autres formats)
• Facilitent le déploiement et le passage à une plus grande échelle de la procédure
• Simplifient la compréhension et la cognition de la procédure

Cependant, avec le temps, les plates-formes de flux de travaux sont devenues, à partir de simples plates-formes de gestion de processus, des systèmes permettant la distribution des tâches et l'exécution conditionnelle de celles-ci en fonction des données (Oinn *et al.*, 2006, Giardine *et al.*, 2005). Ce genre de plate-forme permet aussi la simplification de l'automatisation de tâches générant des erreurs fréquentes (*error-prone*), la collection des données incluant l'organisation des entrées/sorties, le reformatage (*refactoring*) des flux de

travaux, l'analyse des données de sortie et la visualisation des résultats (Oinn *et al.*, 2007, Beaulah *et al.*, 2008). Dans cette thèse, nous garderons donc comme définition d'une plate-forme de flux de travaux : une plate-forme permettant la création de patrons de tâches pouvant être modifiés, l'automatisation des tâches de conversion entre les entrées et sorties et l'exécution sur plusieurs systèmes d'exploitation.

2.4.1 L'architecture des systèmes de gestion de flux de travaux

L'architecture de référence des flux de travaux d'affaires n'est pas souhaitable pour les flux de travaux scientifiques (Hollingsworth, 1995). Les deux architectures ou modèles d'exécution ont le même objectif. Par contre, dans les flux de travaux d'affaires, on cherche à réduire le nombre de ressources humaines (ainsi que les coûts) et à augmenter les revenus, alors que dans les flux de travaux scientifiques, on veut réduire l'action humaine et diminuer le temps de calcul de manière à augmenter le nombre de résultats (van der Aalst et Stahl, 2011). Toutefois, les flux de travaux d'affaires sont orientés vers la coordination d'un travail impliquant plusieurs partenaires, alors que les flux de travaux scientifiques sont conçus pour la répétition d'une tâche sans apport extérieur. Ainsi, trois grandes différences dans l'architecture différencient les flux de travaux d'affaires et les flux de travaux scientifiques: la représentation des données, le modèle d'exécution et la sémantique d'utilisation des données (voir ci-bas).

2.4.2 Données versus variables partagées

Dans les flux de travaux d'affaires, les variables sont partagées. À l'inverse, dans les flux de travaux scientifiques, les données représentent en même temps leur disponibilité et les données elles-mêmes. Ainsi, chaque tâche reçoit sa propre copie des données. Dans les flux de travaux scientifiques, ce paradigme est résolu à l'aide d'une queue, de façon à ce que si une tâche A requiert des valeurs produites par B et C et que C produit plusieurs valeurs, plusieurs instances de A seront créées par les plates-formes de flux de travaux.

2.4.3 Exécution concurrente versus exécution séquentielle

Dans les flux de travaux scientifiques, chaque tâche peut s'exécuter de manière concurrente. Par exemple, on peut retrouver de multiples instances d'une tâche en cours, ce qui entraine un mécanisme de synchronisation. Ce mécanisme implique qu'une sémantique soit convenue pour faire la concaténation des sorties sous la forme d'un tableau (voir par exemple Sroka *et al.*, 2009).

2.4.4 Sémantique individuelle versus sémantique collective

Dans les flux de travaux scientifiques, les données sont utilisées dans l'ordre de leur arrivée et ne sont pas interchangeables. Par contre, dans les flux de travaux d'affaires, les données représentent un contrôle sur l'exécution et l'ordre d'exécution n'est pas formellement défini (van der Aalst et Stahl, 2011).

2.4.5 Modèles d'exécution des flux de travaux scientifiques

Les flux de travaux scientifiques impliquent normalement une indépendance lors de l'exécution, *c.-à-d.* une seule personne accède au flux de travaux. De plus, les *data-flows*, qui ne peuvent utiliser la logique de contrôle du *control-flow*, utilisent, pour contrôler le flux des données, des constructions externes qui ne sont pas propres à la logique du *data-flow* : telles que des boucles (*p.ex. ForLoop*) ou des branchements conditionnels (*p.ex. If*) (Migliorini *et al.*, 2011; Sroka, *et al.*, 2009). De plus, alors que les flux de travaux d'affaires ne peuvent suspendre temporairement une tâche en cours d'exécution, les flux de travaux scientifiques peuvent allouer des tâches et les suspendre en attendant de disposer des entrées de celles-ci. Ainsi, on définit généralement les flux de travaux d'affaires comme étant des *control-flows*, et les flux de travaux scientifiques généralement comme des *data-flows* en relation à leur modèle de contrôle (Migliorini *et al.*, 2011).

2.4.6 *Control-flows* et *data-flows*

Dans les flux de travaux de type *control-flow*, la logique temporelle est définie telle qu'une tâche doit attendre sa complétion avant l'exécution d'une autre. La répétition de tâches et l'exécution de celles-ci en parallèle doivent être définies lors de la conception du flux de travaux. Ce type de représentation en *control-flow* rend plus facile leur cognition (van der Aalst et van Hee, 2004). À l'inverse, dans les flux de travaux de type *data-flow*,

l'ordonnancement des tâches est déterminé par la disponibilité des données. Une problématique de ce modèle d'exécution (*data-flow*) est la difficulté de suivre l'exécution du flux de travaux et de savoir quelle partie est en cours d'exécution (Migliorini *et al.*, 2011).

2.5 Systèmes de gestion des flux de travaux scientifiques

Les systèmes de gestion de flux de travaux scientifiques permettent une formalisation du processus scientifique (Lin *et al.*, 2001, 2009). Dans ce domaine, ils sont principalement utilisés pour faciliter l'exécution de tâches répétitives et pour lier des applications spécialisées (Bowers *et al.*, 2008; Parastatidis, 2009). Une architecture de référence a été décrite pour les flux de travaux scientifiques par Lin *et al.* (2009). Cette architecture prône sept points suivants (ci-bas, Tableau 2.2):

Tableau 2.2 Architecture de référence des plates-formes de flux de travaux scientifiques.

Caractéristiques (adapté de Lin *et al.*, 2009)
• L'interface utilisateur permet l'interaction et peut être adaptée par l'utilisateur (personnalisation de l'interface utilisateur par l'usager).
• Supporte la reproductibilité dans les résultats.
• Intègre des services (*p.e.* services Web) et des logiciels hétérogènes.
• Supporte la distribution et la gestion de données hétérogènes.
• Offre le support pour les plates-formes de hautes performances.
• Permet la surveillance pendant le fonctionnement et permet la reprise, même dans le cas d'erreurs.
• Permet une interconnexion avec d'autres systèmes (*p.ex.* de flux de travaux, d'ordinateurs en réseau)

2.6 Plates-formes de flux de travaux en bioinformatiques

Dès 2006, Seibel et ses collaborateurs (Seibel *et al.*, 2006) ont reconnu le besoin de réunir différents logiciels de bioinformatique en une chaîne de commandes. Ils ont ainsi créé une librairie en *Java* (*BioDOM*), implémentant un langage dérivé du langage hiérarchique

Extensible Markup Language (*XML*), pour permettre le passage de données (séquences, alignements multiples de séquences, etc.) d'une application à une autre.

Présentement, il y a un engouement certain pour les flux de travaux dans la recherche scientifique. Ainsi, les articles présentant les deux plates-formes de flux de travaux les plus connues, Galaxy (Giardine *et al.*, 2005) et Taverna (Oinn *et al.*, 2007), ont été respectivement citées 1489 fois et 2016 fois (*Google Scholar*, Juin 2014). Cependant, peu de ces citations portent sur des recherches utilisant ces systèmes de flux de travaux dans le domaine de la phylogénétique et de la phylogénomique. Plusieurs autres systèmes de flux de travaux ont été développés pour des applications en bioinformatique tels que Kepler (Bowers *et al.*, 2008), LONI (Dinov *et al.*, 2011) et Triana (Taylor *et al.*, 2007). D'autres systèmes de flux de travaux scientifiques conçus pour la bioinformatique sont aussi disponibles et sont répertoriés sur une page Web de Wikipedia[6]. La plupart des systèmes de flux de travaux actuels en bioinformatique sont très similaires dans leur utilisation. Ils sont principalement orientés vers une plus grande utilisation du pouvoir computationnel, l'intégration des services en ligne (*Web Services*) et un accès de plus en plus important au « nuage » (*cloud computing*) (Dinov *et al.*, 2011). Une brève compilation de leurs particularités est décrite au Tableau 2.3 et plus en détails dans les sections suivantes.

[6] http://en.wikipedia.org/w/index.php?title=Bioinformatics_workflow_management_systems

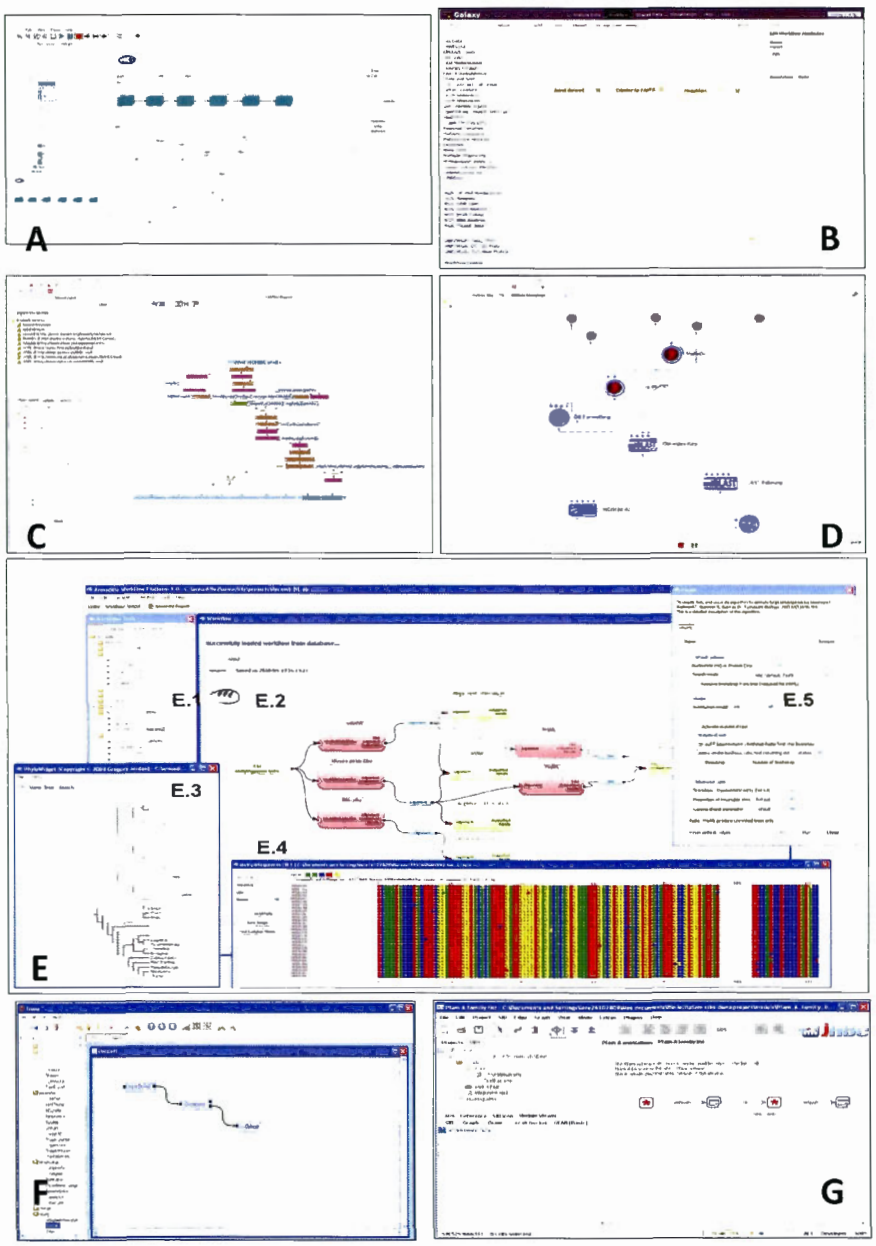


Figure 2.2 Vues des principales plates-formes de flux de travaux bioinformatiques : A) Kepler, B) Galaxy, C) Taverna, D) LONI, E) *Armadillo* (voir la Section 3), F) Triana, G) Bio-Jeti. La nouvelle plate-forme *Armadillo* comprend un navigateur de méthodes (E1), une fenêtre de création (E2), un outil de visualisation des arbres phylogénétiques (E3), une vue des alignements de séquences (E4) et une fenêtre d'édition propre à chaque méthode (E5).

Tableau 2.3 Comparaison des principaux systèmes de flux de travaux en bioinformatique.

Plate-formes	Classes	Recompilation des outils	Flux de travaux	Langages de programmation	Modèle Client-Serveur	Accès au nuage	Langage de description	Accès au « nuage »	Application	Site Web
Kepler	**Local**	Oui (via API)	*Data-flow*	Java	Oui	Oui (*ecogrid*)	MoML	Oui (*ecogrid*)	Général	**kepler-project.org**
Galaxy	**Web**	Non	*Data-flow*	Ruby, Python, Perl, Bash	Oui	Oui (*Amazon EC2*)	Propriétaire	Oui (*Amazon EC2*)	Bioinformatique	**galaxy.psu.edu**
Taverna	**Local, services**	Oui (via API)	*Data-flow*	Java	Oui	Oui (*mygrid*)	Scufl Scufl2	Oui (*mygrid*)	Bioinformatique	**taverna.org.uk**
LONI	**Local, services Web**	Non	*Data-flow*	Java	Oui	Oui	XML	Oui	Bioinformatique	**pipeline.loni.ucla.edu**
Bio-Jeti	**Local, services Web**	Non	*Control-flow*	Java	Oui	Oui	XML	Oui	Bioinformatique	**biojeti.cs.tu-dortmund.de**
Triana	**Client-Serveur**	Oui (via API)	*Data-flow/ Control-flow*	Java	Non	Non	*TaskGraph*	Oui	Général	**trianacode.org**
Armadillo *	**Local**	**Oui (via API)**	*Data-flow/ Control-flow*	Java	Non	Non	Clé-valeurs	Non	**Phylogénomique**	**bioinfo.uqam.ca/armadillo**

* Nouvelle plate-forme introduite dans cette thèse au chapitre 3.

2.6.1 Kepler

La plate-forme Kepler *version 2.2* (Figure 2.2A) est un système de gestion de flux de travaux scientifique « open source » implémentée dans le langage de programmation *Java*. Elle a été développée par les membres du *Science Environment for Ecological Knowledge* (SEEK) et du *Scientific Data Management* (*SDM*). Ce système est conçu pour la modélisation, la simulation et la conception de tâches concurrentes ou de gestion en temps réel (Altintas *et al.*, 2004). Développé à UC Berkeley, elle est l'une des rares plates-formes de flux de travaux qui supporte différents modèles d'exécution des tâches. Ces différents modèles sont : *Synchronous Data-flow* (SDF) permettant l'exécution séquentielle simple mais continuelle du flux de travaux dans un seul fil d'exécution (*thread*), *Dynamic Data-flow* (DDF) où l'ordre d'exécution est déterminé par les données, *Process Network* (PN) dans lequel plusieurs files d'exécution peuvent être créées, et finalement les modèles *Continuous Time* (CT) et *Discrete Event* (DE). L'utilisation de ces différents modèles se fait en assignant des *directors*, des objets insérés directement dans les flux de travaux (Ludäscher *et al.*, 2009). Les tâches sont, quant à elles, décrites comme des *actors* qui permettent des activités dans différents domaines incluant la bioinformatique (Bowers *et al.*, 2008; Migliorini *et al.*, 2011). Le flux de travaux final est décrit dans le langage *MoML*[7], dérivé du *XML*, qui présente la topologie du flux de travaux mais aussi les différents objets et leurs ports d'entrée et sortie.

Le projet pPOD (*processing PhylOData*; *phylodata.org*) a utilisé le système de flux de travaux Kepler pour uniformiser la création d'arbres phylogénétiques à l'intérieur du projet AToL (*A Tree of Life, http://tolweb.org/*) du National Science Fondation des États-Unis (Bowers *et al.*, 2008). Les objectifs d'utilisation de Kepler dans ce projet étaient de développer un modèle de données encapsulant tous les types de données menant à la création d'arbres phylogénétiques, tout en permettant la collecte de toutes les données créées lors de cette inférence. De plus, le projet pPOD fait partie intégrante du CIPRES (Cyberinfrastructure for Phylogenic Research) (Ludäscher *et al.*, 2009). Un des désavantages de cette plate-forme est de requérir la recompilation des outils, la nécessité de mettre en place

---

[7] A Modeling Markup Language in XML
(ptolemy.eecs.berkeley.edu/publications/papers/00/moml/moml_erl_memo.pdf)

un serveur d'applications et la programmation des objets ou l'utilisation de services Web pour leur utilisation. De plus, il n'y a pas de vérification des types de données lors de la création des flux de travaux.

2.6.2 Galaxy

Galaxy (Figure 2.2B) est un système de flux de travaux en ligne programmé en *Python* et *Ruby* à l'Université Pennsylvania State et à l'Université Emory (Giardine. *et al.,* 2005). Celui-ci peut cependant être installé dans un environnement Linux ou Mac, ou encore être exécuté sur une instance du Amazon EC2 Cloud[8] (Afgan *et al.,* 2010). Orienté vers l'étude des données de séquençage à haut débit, il est principalement un engin qui permet l'exécution de logiciels ou de lignes de commandes sous la forme d'un *data-flow*. Dans celui-ci, les données sont définies comme des tables dont on peut choisir certaines colonnes comme données à analyser (Schatz *et al.,* 2010; Woollard, 2010). Galaxy permet également l'utilisation de données telles que des alignements de séquences (Blankenberg *et al.,* 2011). Il a permis plusieurs études incluant l'analyse de microbiomes (Kosakovsky Pond *et al.,* 2009), l'analyse de gènes non-codants (Hinchcliffe et Webster, 2011) et l'étude de l'hétéroplasmie (la présence de plusieurs variantes de génomes mitochondriaux) chez les eucaryotes (Goto *et al.,* 2010).

Bien que récemment mise à la disposition de la communauté scientifique, la plate-forme Galaxy a comme avantages en phylogénétique d'être mature et facile à utiliser pour des biologistes. De plus, son utilisation en ligne permet de préserver les données et donne accès à des ressources computationnelles importantes. Cependant, cette utilisation en ligne ne permet pas la manipulation de grands jeux de données s'ils ne se retrouvent pas dans les grandes banques de séquences (*p.ex. NCBI*). De plus, l'installation locale est supportée sous Linux, mais n'est pas automatisée et n'inclut qu'une seule méthode d'analyse phylogénétique dans sa version de base.

Pour remédier à ces limitations, différentes extensions ont été apportées à la plate-forme. Par exemple, il est maintenant possible de lier la plate-forme Galaxy à la plate-forme Taverna,

grâce à l'application Tavaxy (Abouelhoda *et al.*, 2012). Cette extension permet ainsi : 1) l'exécution de programmes locaux, 2) le support d'utilitaires permettant la manipulation des séquences, 3) la distribution de tâches sur des ressources computationnelles supplémentaires et 4) l'accès à des structures de contrôle (*if-else* et *loops*). Une autre extension développée récemment est Osiris (Oakley *et al.*, 2014). Cette dernière permet de mener des études phylogénomiques directement sur Galaxy, en incluant des outils tels que l'alignement de milliers de séquences avec MAFFT (Katoh et Standley, 2013) et l'inférence d'arbres phylogénétiques avec les logiciels RAxML (Stamatakis, 2006) et BEAST (Drummond et Rambaut, 2007).

2.6.3 Taverna

Probablement le plus connu et utilisé des systèmes de flux de travaux en bioinformatique, Taverna version 2.5 (Figure 2.2C) est un logiciel *open source* permettant la création et l'exécution de flux de travaux basés sur des services Web ou l'exécution de tâches locales, par exemple des *scripts* ou des applications *Java*. Créé dans le cadre du projet *myGrid* (Stevens *et al.*, 2003), Taverna est présentement intégré au projet *myExperiment* (Goble *et al.*, 2010), un site Web cataloguant et permettant la recherche et la diffusion de flux de travaux, principalement dans le domaine de la bioinformatique. Supportant seulement un modèle d'exécution, les flux de travaux y sont définis dans un langage dérivé du *XML*, le *Scufl* (Turi *et al.*, 2007), langage qui est présentement à sa deuxième version (*t2flow/Scufl2*) (Sroka *et al.*, 2009; Missier *et al.*, 2010).

Contrairement au *BPEL4WS* (Weerawarana *et al.*, 2005), qui est un langage orienté vers le *control-flow* ayant inspiré sa création (Weerawarana, *et al.*, 2005), le Scufl2 (Sroka, *et al.*, 2009) est un langage de *data-flow* qui permet la construction d'un graphe d'exécution de services distants et locaux. Dans celui-ci, les différentes tâches sont appelées des *processors* qui possèdent des ports d'entrée (*input*) et de sortie (*output*). Chaque *processor* peut être lié à d'autres par des *datalinks*. D'autres types de liens, tels que les *coordination links*, permettent de spécifier une ordination des tâches sans avoir besoin d'être lié à des entrées/sorties. De

[8] http://aws.amazon.com/fr/ec2/

plus, chaque flux de travaux peut avoir des entrées formelles définies comme sources et sorties (*sinks*). L'exécution se produit alors lorsque toutes les entrées de la source sont disponibles et se termine lorsque les sorties sont produites ou lorsqu'il y a des erreurs. Taverna fait grand usage des services Web qui forment la base de la description des tâches. Cependant, du code *Java* peut aussi être compilé et exécuté à l'exécution du flux de travaux. De fait, les utilisateurs peuvent spécifier une liste d'alternative (logiciels et services Web) pour chacun des *processors*. Dans sa définition, un processor peut avoir d'autres paramètres tels que le nombre d'essais et le temps entre les essais dans le cas d'échecs des services Web. Finalement, aucune structure de contrôle (*control-flow*) n'est incluse dans Taverna de prime abord. L'utilisateur peut, par contre, définir une structure de choix en utilisant des composantes du langage de programmation *Java* pouvant être incorporées au flux de travaux (Migliorini *et al.*, 2011).

Un avantage de Taverna est l'utilitaire *CalcTav* qui permet de définir et d'exécuter à partir d'une feuille de données *OpenOffice*, des flux de travaux à partir des données s'y retrouvant (Sroka *et al.*, 2011). De plus, beaucoup d'exemples de flux de travaux sont disponibles en ligne. Cependant, un important désavantage de l'utilisation de Taverna est le faible niveau d'abstraction dans sa description des types de données. Ainsi, ces types sont simples tels que des chaînes de caractères, des entiers, des booléens (Missier, *et al.*, 2010). De fait, ce faible niveau d'abstraction complexifie la création de flux de travaux et les rend plus complexes. De plus, les services Web ne sont pas toujours idéaux pour l'analyse phylogénétique ou phylogénomique car il y a souvent une limite sur le nombre de séquences ou de données pouvant être traitées par ces derniers. Ainsi, l'utilisateur doit effectuer lui-même la division de ces données avant de les soumettre à ces services pour gérer ces limitations.

2.6.4 LONI

Développé en *Java* pour l'analyse de l'imagerie médicale, LONI *version 5.0* (Figure 2.2D) (Rex *et al.*, 2003; Dinov *et al.*, 2009 et 2011), ce système permet l'exécution de flux de travaux locaux (suite à la création d'un serveur) ou distants, décrits dans un langage *XML*. Le système LONI utilise une logique d'exécution de type *data-flow* lors de son exécution, utilisant le *Oracle Grid Engine* (Dinov *et al.*, 2010). Plusieurs exemples de flux de travaux en bioinformatique incluant des exemples de recherche de similarité de séquences BLAST et

l'analyse des données de séquençage à haut débit sont disponibles et sont aussi répertoriés sur le site de *myExperiment* (voir Section 2.3). L'intégration d'applications pour l'inférence phylogénétique telles que PHYLIP (Felsenstein, 2005) et PAUP (Swofford, 2002) est prévue pour la prochaine mise à jour.

Cette plate-forme a pour avantage en phylogénomique d'avoir une bonne documentation en ligne, des outils variés ainsi que plusieurs publications dans le domaine médical. Cependant, peu d'outils spécifiques sont présentement disponibles en phylogénétique. En plus, la nécessité de l'installation d'un serveur d'applications et/ou l'utilisation d'outils en ligne limitent son application en phylogénomique.

2.6.5 Triana

Triana *version 4.0* (Figure 2.2F) est un logiciel *open source* développé en *Java*, utilisant le *Java Remote Method Invocation*[9] comme engin d'exécution. Triana a été développé à l'Université Cardiff comme un projet de physique pour la détection de champs de gravitation (voir Majithia *et al.,* 2004; Taylor et Schutz, 1997). Les composantes de Triana sont implémentées comme des services (*proxy*) pouvant être locaux ou distribués. Ces services peuvent être: des objets *Java*, des exécutables locaux, des services Web, des ressources Web, le lancement d'une tâche partagée, ou un fichier local ou distant. Nommées *components*, ces composantes acceptent, traitent et produisent des données via des *ports* définis comme des interfaces (Churches *et al.,* 2006). Les *components* peuvent avoir plusieurs paramètres tels que des adresses *proxy* alternatives et des répertoires de référence. Le flux de travaux est défini dans un langage *XML* proche du *WSDL*[10] (un format de description des services Web) (Weerawarana *et al.,* 2005), alors que les *components* sont des fichiers *Java* pouvant être modifiés par l'utilisateur (Migliorini *et al.,* 2011). Triana supporte à la base un modèle d'exécution basé sur un *data-flow*. Cependant, quelques composantes permettent un *control-flow* via des messages internes (*trigger*). Des exemples de ces composantes incluent des constructions telles que des *Loop*, *If*, *Duplicator* et *Merge*.

---

[9] http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html
[10] http://www.w3.org/TR/wsdl

Peu d'applications bioinformatiques ont été développées avec Triana. Un seul flux de travaux est disponible dans le répertoire *myExperiment* (voir Figure 2.3), principalement à cause de la nécessité de connaître le langage de programmation *Java* pour la création des flux de travaux. De plus, il est difficile de connaître les différents types de données de toutes les composantes lors de la création ou de l'utilisation du flux.

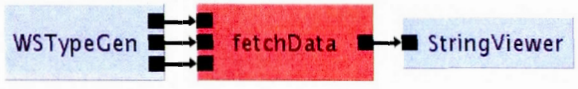


Figure 2.3 Flux de travaux dans Triana[11] permettant l'accès à des données provenant de la European Bioinformatics Institute (EBI).

Ainsi, un avantage de l'utilisation de Triana en phylogénomique est le fait que celle-ci permet des exécutions en parallèle. Cependant, le peu d'exploitation de cette plate-forme de flux de travaux en bioinformatique, ainsi que la nécessité de connaître le langage de programmation *Java* pour créer les objets du flux de travaux, rendent son application limitée.

2.6.6 Bio-Jeti

La plate-forme Bio-Jeti (Figure 2.2G; Lamprecht *et al.*, 2009b) est orientée vers l'application des principes de la sémantique et de l'exploration de modèle (*model checking*) à la conception de flux de travaux. Plus spécifiquement, cette plate-forme de l'Université de Potsdam est bâtie sur l'architecture de jABC et jETI (*electronic tool integration platform*) (Margaria *et al.*, 2005). Elle sert à développer des graphes, qui peuvent ensuite être convertis en diverses spécifications (*p.ex.* en code *C++* ou *Java*) et formats (*p.ex. BPEL*) via le générateur de code GeneSys (Lamprech *et al.*, 2009a). De plus, Bio-Jeti vérifie que les graphes créés répondent aux critères des différents modèles. Elle peut aussi compléter certains flux de travaux en suggérant des outils de transformation (Lamprech *et al.*, 2009a).

Présentement, cette plate-forme inclut différents types de données définis tels que : *Entier*, *Chaîne de caractères*, *Accession*, *Alignement de séquences*, *Séquence*, *Arbre*, *Description*,

[11] http://www.myexperiment.org/workflows/894.html

Liste itérative et *Compteur*. De plus, cette plateforme inclut des structures de contrôle telles que des itérations et des boucles (*ForLoop*). Cependant, bien que cette plate-forme compte des outils permettant l'alignement de séquences, la recherche dans des bases de données biologiques (*DDBJ*[12], *BioMoby*[13], *EBI*) et certaines méthodes d'inférence phylogénétiques via des services Web, elle ne comprend pas d'applications exécutées localement.

2.7 Réutilisation des flux de travaux

Bien que les plates-formes de flux de travaux existent depuis des années, plusieurs défis restent à combler en ce qui concerne les flux de travaux scientifiques tels que: l'abstraction limitée des types de données (Goderis, 2008), une plus grande facilité d'accès programmatique aux données, une meilleure performance des plates-formes de création, des ressources (*applications*) utilisables plus importantes, une meilleure sémantique et une plus grande facilité à conserver les traces des exécutions (Romano, 2008). De plus, bien qu'il soit à l'avantage des systèmes de gestion de flux scientifiques de pouvoir accommoder une diversité de tâches en permettant l'incorporation de plusieurs types de services (services Web, exécutables locaux, exécution de scripts), cette approche fait que peu de flux de travaux sont réutilisés et réutilisables en pratique (Goderis *et al.*, 2005; Goderis, 2008; Zhao *et al.*, 2012). D'ailleurs, l'étude de Zhao *et al.* (2012) a démontré que plus de 80 % des flux de travaux trouvés sur le site de *myExperiment* ne pouvaient être exécutés tels quels. Ainsi, pour faciliter cette réutilisation de flux de travaux, Hettne et ses collaborateurs (Hettne *et al.*, 2012) ont présenté sept pratiques qui favoriseraient cette réutilisation (voir le Tableau 2.4).

---

[12] http://www.ddbj.nig.ac.jp/
[13] http://biomoby.open-bio.org/

Tableau 2.4 Pratiques favorisant la réutilisation des flux de travaux.

Pratiques (adapté de Hettne *et al.*, 2012)
• Créer des flux de travaux abstraits dans lesquels plusieurs modules peuvent être utilisés si l'un devient inutilisable. Choisir de fait des services provenant de distributeurs fiables (*p.ex. EBI, NCBI*)
• Proposer plusieurs données en sortie, faisant en sorte que ce flux de travaux puisse être réutilisé
• Proposer des exemples d'entrées et de sorties
• Annoter le flux de travaux pour montrer toutes les étapes et utiliser des standards
• Faire en sorte que le flux de travaux ne soit pas spécifique à l'environnement ou système d'exploitation
• Réutiliser des sections de flux de travaux si possible
• Valider les flux de travaux par des procédures de vérification

2.8 Comparaison des flux de travaux

La classification spécifique des flux de travaux est un domaine peu étudié. On traite plutôt de la comparaison de graphes les représentant (Santos *et al.*, 2008). Néanmoins, la classification des flux de travaux, et par extension, leur comparaison et l'établissement d'une mesure de distances sont essentielles à plusieurs de leurs usages soit : 1) la réutilisation de flux de travaux ou de parties de ceux-ci (Goderis, 2008; Hettne *et al.*, 2012), 2) la prédiction de leurs temps d'exécution (Smith *et al.*, 1999; Nadeem et Fahringer, 2009), 3) la dispersion de leurs tâches respectives sur de plus grands environnements computationnels (*architecture du nuage*) (Da Silva *et al.*, 2013; Singh *et al.*, 2008; Wieczorek *et al.*, 2009) et l'optimisation de leur exécution (Lee *et al.*, 2009; Vairavanathan *et al.*, 2012).

Toutes les données non-uniformes contiennent une structure due à l'hétérogénéité des tâches. Le processus permettant de retrouver cette structure est le regroupement ou la classification des éléments en groupes (Schaeffer, 2007). Ce regroupement fait normalement appel à une mesure de similarité entre les objets. Ce problème est NP-complet, même en ne considérant que deux classes (Drineas *et al.*, 1999) car il y a 2^{N-1}-1 bipartitions possibles pour répartir un

ensemble de N éléments (Everitt *et al.,* 2001). De fait, les algorithmes de regroupement sont normalement des heuristiques.

On qualifie normalement ce regroupement de non-supervisé (*p.ex.* par l'algorithme k-means) si l'on ne considère pas d'*a priori* sur les différentes classes représentées par les objets (Conte *et al.,* 2004). De plus, ce regroupement peut être qualifié de fort (*hard clustering*), si chaque élément ne peut être assigné qu'à une seule classe, tel que dans les algorithmes de k-means ou UPGMA. À l'inverse le regroupement doux (*soft clustering* ou *fuzzy clustering*) permet l'assignation d'un élément à plusieurs classes en fonction d'une probabilité (Vesanto et Alhoniemi, 2000). On peut citer, par exemple, l'algorithme c-means permettant cette classification (Bezdek, 1981; Hathaway *et al.,* 2000). Cependant, dans le cadre de ce travail, nous considèrerons uniquement le regroupement fort.

2.9 Mesure de distance entre flux de travaux

Une mesure de distance permettant le calcul de la similarité ou de la dissimilarité entre deux flux de travaux doit posséder les quatre propriétés suivantes (Schaeffer, 2007; voir aussi Johnson et Wichern, 1992, pour la définition d'une distance) :

1. La distance entre deux flux de travaux ne peut pas être négative.
2. La distance entre un flux de travaux et lui-même est de 0.
3. Une distance de 0 entre deux flux de travaux indique qu'ils sont identiques.
4. La distance entre les flux de travaux W_1 et W_2 est équivalente à la distance entre les flux de travaux W_2 et W_1.

Ainsi, on omet normalement la propriété de l'inégalité triangulaire lorsque l'on compare des flux de travaux (Wormbacher et Li, 2010), bien qu'elle soit requise pour satisfaire la définition d'une distance (Johnson et Wichern, 1992). On peut, dès lors, appliquer des mesures de similarité ne requérant pas d'être strictement dans l'espace Euclidien en comparant, par exemple, des représentations vectorielles (Schaeffer, 2007). Ainsi, en considérant des données quantitatives, différent algorithmes de regroupement prennent en

compte soit : la distance de Minkowski, la distance Euclidienne, la distance cosine, la distance de Manhattan, la distance *Sup*, la corrélation de Pearson, ou encore la distance symétrique des points ou la corrélation de rangs de Spearman (revue par Xu et Wunsch, 2005).

2.9.1 Distances Euclidienne, cosine et autres

Dans le cas où un espace Euclidien à *n*-dimensions est considéré, la distance Euclidienne ou distance norme L^2 (ou distance de Pythagore) est la plus utilisée (Équation 7). Elle représente la distance entre deux éléments dans un espace à *n* dimensions (ou variables) ou encore sur un plan (pour lequel *n*=2). Dans le cas de *n* variables décrivant un même élément, soit $P = \{x_1, \ldots, x_n\}$ et $Q = \{y_1,\ldots,y_n\}$, on définit cette distance telle que :

$$d(P,Q) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2} \,. \qquad (7)$$

Un vecteur de poids *w*, tel que défini par Makarenkov et Legendre (2001), peut aussi être ajouté à la distance Euclidienne (Équation 8) :

$$d(P,Q) = \sqrt{\sum_{k=1}^{n} w_k (x_k - y_k)^2} \,. \qquad (8)$$

Ce vecteur de poids *w*, contenant des valeurs supérieures à zéro ($w \geq 0$), permet la normalisation de chacun des éléments et permet de ne pas tenir compte de certaines valeurs non pertinentes (Milligan, 1989). Une autre distance est la distance de Manhattan (Équation 9), ou norme L^1:

$$d(P,Q) = (\sum_{k=1}^{n} |x_k - y_k|^2) \,. \qquad (9)$$

Ou encore la distance de Minkowski (Équation 10) qui fait un lien entre ces dernières. Dans ce cas, si *p* est égal à deux, on a la distance Euclidienne, et si *p* est égal à 1, on a la distance de Manhattan :

$$d(P,Q) = (\sum_{k=1}^{n} | x_k - y_k |^2)^{1/p}. \quad (10)$$

Dans le cas où l'on prend en compte un espace non-Euclidien, on peut considérer des distances telles que la distance cosine (ou distance *cosinus*, Équation 11) :

$$d(P,Q) = 1 - \cos\theta = 1 - \frac{\sum_{k=1}^{n}(x_k \times y_k)}{\sqrt{\sum_{k=1}^{n} x_k^2} \times \sqrt{\sum_{k=1}^{n} y_k^2}}. \quad (11)$$

Ou encore la distance cosine pondérée :

$$d(P,Q) = 1 - \cos\theta = 1 - \frac{\sum_{k=1}^{n} w_k(x_k \times y_k)}{\sqrt{\sum_{k=1}^{n} w_k x_k^2} \times \sqrt{\sum_{k=1}^{n} w_p y_k^2}}. \quad (12)$$

La mesure de Tversky (1977), dans laquelle la similarité est déterminée par le nombre d'éléments ayant des caractéristiques similaires, a aussi été utilisée par Goderis (2008) pour comparer des flux de travaux. Dans d'autres mesures, telle que la distance de Jaccard (Jaccard, 1901), on considère une représentation binaire des données (revue par Choi *et al.*, 2010). Dans la distance de Tanimoto (Tanimoto, 1957), qui en est une extension, on respecte l'inégalité triangulaire (Lipkus, 1999). Cette dernière distance a été utilisée par certains auteurs pour comparer des structures sous forme de graphes (Wilkens *et al.*, 2005).

2.9.2 Distance de graphes

D'autres mesures d'évaluation de la similarité, appliquées aux flux de travaux (Santos *et al.*, 2008), sont les méthodes de graphes (Conte *et al.*, 2004; Dijkman *et al.*, 2009). Des mesures de similarité basées sur le *maximum common induced subgraph* (*MCIS*) ou *maximum common supergraph* (Bunke et Shearer, 1998; Fernández et Valiente, 2001; Wallis *et al.*, 2001) ont été proposées pour les flux de travaux (Goderis *et al.*, 2008; Santos *et al.*, 2008) (Équation 13) :

$$dMCIS(W_1,W_2) = 1 - \frac{|MCIS(W_1,W_2)|}{\max(|W_1|,|W_2|)} . \quad (13)$$

Dans cette dernière équation, présentée par Santos *et al.* (2008), deux graphes sont dits isomorphiques s'il y a correspondance exacte entre leurs nœuds et qu'une arrête présente entre deux nœuds est aussi présente dans les deux flux de travaux. Un sous-graphe commun de deux flux de travaux W_1 et W_2 consiste alors aux sous-graphes H_1 de W_1 et H_2 de W_2 tel que H_1 est isomorphique à H_2 (McGregor, 1982). Un sous-graphe commun maximum (*MCS*) des deux flux de travaux est alors le sous-graphe contenant le plus grand nombre d'arcs communs aux deux flux de travaux W_1 et W_2. Un sous-graphe H de W_1 et W_2 est appelé induit si tous les arcs présents dans W_1 et W_2 sont aussi présents dans H. Le sous-graphe induit maximum (*MCIS*) est alors le graphe contenant le plus grand nombre de ces arcs (Raymond et Willet, 2002). La notation $|W_1|$ indique la cardinalité du flux de travaux W_1. Cependant, un problème persiste dans ce type de comparaison puisqu'elle requiert un nombre suffisant de structures et d'annotations communes aux deux graphes pour permettre une bonne comparaison (Santos *et al.*, 2008).

De plus, les flux de travaux n'ont pas de contraintes fixes sur la dimensionnalité, *c.-à-d.* le nombre de nœuds et d'arrêtes n'est pas fixé *a priori* (Cordella *et al.*, 2004; Conte *et al.*, 2004). Un désavantage de cette représentation graphique des flux de travaux, lors de leur classification, est alors la complexité algorithmique résultante de la comparaison des différents chemins dans les graphes (Schaeffer, 2007). Ainsi, alors que la comparaison de deux vecteurs se réalise en temps linéaire (Conte *et al.*, 2004; Xu et Wunsch, 2005), la comparaison de deux graphes par des algorithmes exploitant l'isomorphisme des graphes peut résulter en une complexité factorielle (Cordella *et al.*, 2004). Il faut cependant noter que des algorithmes de complexité polynomiale sont disponibles pour les graphes sous différentes contraintes telles que les arbres et les graphes planaires (Luks, 1982). Cependant, l'algorithme de référence de similarité de graphes par isomorphismes des sous-graphes (Ullmann, 1976) a une complexité algorithmique de $O(N^2N!)$. Encore, l'algorithme de *maximum common subgraph* de Bunke *et al.* (2002) et l'algorithme *VF2* (Cordella *et al.*, 2004) ont une complexité dans le temps de $O(N!N)$ pour vérifier la similarité entre deux graphes.

Ainsi, des approches permettant des erreurs ont été introduites pour la comparaison de graphes, incluant l'utilisation de distance d'édition (Robles-Kelly et Hancock, 2005; Bunke et Allermann, 1983). La distance d'édition des graphes (*graph edit distance*) est une évaluation du nombre minimal d'opérations (insertions, délétions, substitutions des nœuds ou vertices dans le graphe) permettant de transformer un graphe en un autre (Riesen et Bunke, 2009; Sanfeliu et Fu 1983). On peut aussi permettre la fusion et la dé-fusion de certains nœuds lors de ce calcul (Wallis *et al.*, 2001; Duda *et al.*, 2011). La difficulté dans ce cas est que la série d'opérations transformant un graphe à un autre peut-être multiple (Riesen et Bunke, 2009). On utilise alors encore des heuristiques pour estimer le nombre d'opérations nécessaires (Conte *et al.*, 2004), résultant en des complexités algorithmiques de $O(N^3)$ (Riesen et Bunke, 2009; Zeng *et al.*, 2009), pour la comparaison de deux graphes.

2.10 Méthodes de regroupement

Les méthodes de regroupement sont divisées en deux catégories (Figure 2.4): les méthodes de partitionnement cherchant à trouver des groupes d'éléments, et les méthodes hiérarchiques cherchant en plus à établir des liens de parenté entre les différents éléments.

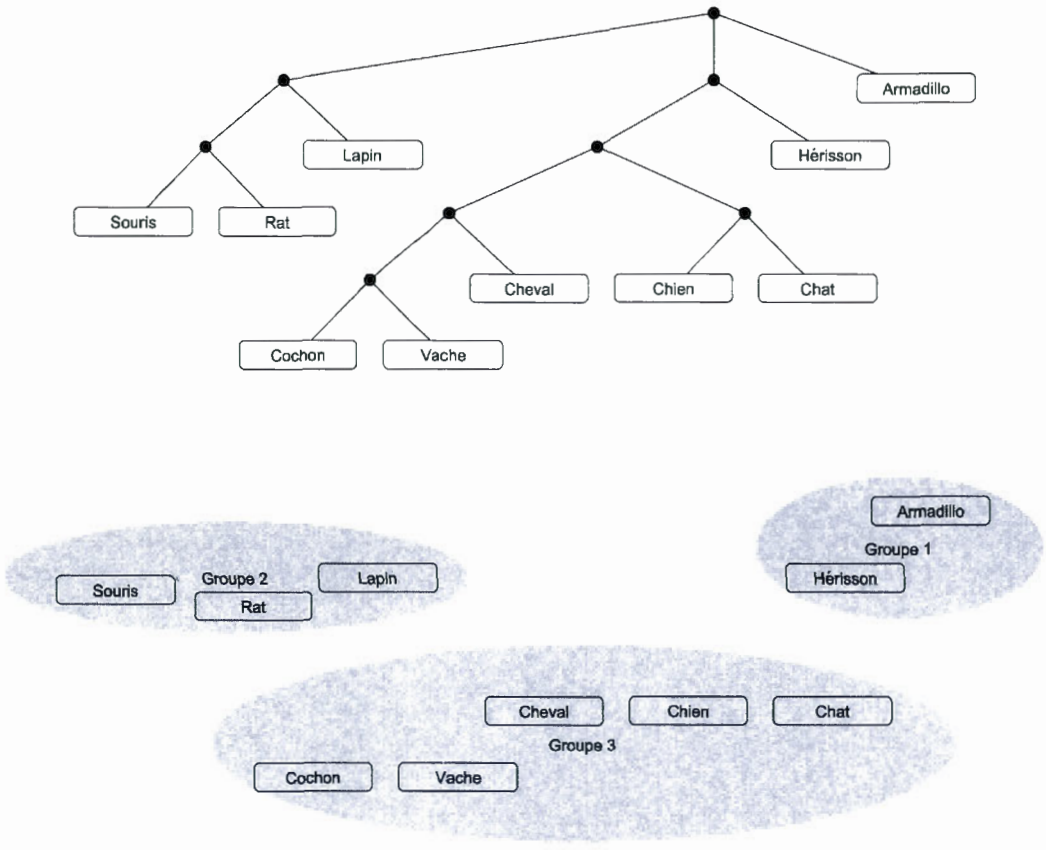


Figure 2.4 Regroupement hiérarchique (*haut*) et regroupement par partitionnement (*bas*).

2.10.1 Méthodes de partitionnement

Les algorithmes de partitionnement cherchent à trouver la meilleure partition de N éléments en K classes (ou groupes). Généralement, ces méthodes produisent des résultats de meilleure qualité que les méthodes hiérarchiques (Ng et Han, 2002). Les méthodes de k-means (MacQueen, 1967) et de k-medoids (Kaufman et Rousseeuw, 1990) sont deux méthodes représentatives de ce type de regroupement qui divisent un ensemble d'éléments en k classes prédéfinies. Cependant, étant des heuristiques, l'application de ces méthodes ne conduit pas toujours au même regroupement. De plus, elles ne permettent pas de détecter les groupes de formes irrégulières (Berkhin, 2006). La méthodologie de k-means compte plusieurs variations (Revue par Jain, 2010) tout comme la méthode k-medoids (Reynolds *et al.*, 2004). D'autres méthodes de partitionnement, telles que les SOM (*Self-Organising Map*), sont capables de

prendre en compte des données de formes irrégulières tout en étant moins sensibles aux points aberrants (Vesanto, et Alhoniemi, 2000). Ces méthodes peuvent aussi être appliquées directement aux graphes en prenant comme distance le chemin le plus court entre deux tâches (Yamakawa *et al.*, 2006; Günter et Bunke, 2002).

2.10.1.1 *k*-means

L'algorithme *k*-means (MacQueen, 1967) est l'un des algorithmes les plus utilisés dans le domaine de la classification (Wu *et al.*, 2008). L'un des avantages de l'algorithme *k*-means et de ses variations est sa simplicité. Une recherche locale, appelée itération de Lloyd, s'effectue itérativement jusqu'à ce qu'il n'y ait plus de changements dans l'assignation des différents éléments à leur groupe (Lloyd, 1957). Dans l'algorithme original, on partitionne automatiquement n éléments en K groupes avec une complexité algorithmique de $O(K \times n \times m \times i)$ où m représente le nombre de variables et le nombre d'itérations (i) est généralement compris entre 50 et 500 (Frahling et Sohler, 2008). Pendant ces étapes, on utilise normalement la distance Euclidienne pour évaluer le centre de chacun des groupes, le *centroïde*, qui représente le point milieu évalué après chaque itération. Effectivement, cette méthode minimise la distance intragroupe au carré et maximise en même temps la somme des distances intergroupes au carré (Witten et Tibshirani, 2010). Une description plus complète de cet algorithme sera présentée au chapitre 4.

Parmi les améliorations apportées à la méthode originale, Bradley et collaborateurs (2000) ont ajouté à l'algorithme une contrainte de taille minimum pour éviter les groupes ne contenant aucun élément. Cette modification fait suite à l'observation que dans le cas d'un regroupement contenant plus de 10 variables décrivant les éléments et un nombre de groupe supérieur à 20, l'algorithme converge souvent vers des minimums locaux et amenant des groupes contenant peu de données ou étant vides. Wagstaff et collaborateurs (2001) ont, par la suite, ajouté des contraintes permettant de spécifier l'inclusion ou l'exclusion d'éléments de certains groupes (*p.ex.* l'élément *X* ne doit pas se retrouver avec l'élément *Y*). D'autres versions hybrides incluent la réduction de l'espace de recherche en utilisant la technique de l'analyse des composants principaux (*PCA*, Ding et He, 2004), considérant des *coresets* (données représentatives) (Frahling et Sohler, 2008), ou encore incluant des algorithmes

génétiques lors de la recherche des centroids (Lu *et al.*, 2004). Finalement, puisque le temps d'exécution de l'algorithme peut devenir exponentiel (Arthur et Vassilvitskii, 2006), plusieurs versions exploitant le parallélisme, telles que *k*-means++ et *k*-means sur des processeurs graphiques, ont été implémentées (Hong-Tao *et al.*, 2008, Bahmani *et al.*, 2012).

2.10.1.2 *k*-medoids

L'algorithme *k*-medoids (Kauffman et Rousseeuw, 1990) est une modification de l'algorithme de *k*-means dans laquelle le centre du regroupement, maintenant appelé *médoïde*, est un objet représentatif du groupe. L'algorithme choisit, dans ce cas, l'élément minimisant la distance intragroupe. Les avantages de la méthode *k*-medoids sur la méthode *k*-means est que cette méthode peut prendre en entrée une matrice de distances et qu'il n'y a pas de recalcul de nouvelles distances à chacune des itérations. Une description détaillée de cet algorithme est présentée au chapitre 4. La complexité algorithmique de cette méthode est ainsi de $O(K \times (n-K)^2 \times m \times i)$.

Plusieurs modifications ont été apportées à l'algorithme original pour réduire cette complexité. Reynolds *et al.* (2004) ont optimisé la recherche des médoïdes en les ordonnant et en utilisant une technique permettant de réduire le calcul de la distance intragroupe. De même, Zhang et Couloigner (2005) ont suggéré l'utilisation d'un réseau triangulaire pour calculer le coût d'utilisation d'un nouveau médoïde durant la phase de recherche pour une application en géographie.

2.10.1.3 Autres méthodes

L'algorithme *PAM* (*Partition Around Medoids*; Kaufman et Rousseeuw, 1990) est une modification de l'algorithme *k*-medoids, utilisant une méthode de *hill-climbing* pour trouver des médoïdes potentiels à chaque itération en interchangeant seulement des éléments.

La méthode *PAM* peut être résumée telle que :

1. Prendre *K* éléments aléatoirement pour être les médoïdes originaux et assigner chacun des non-médoïdes au groupe le plus proche.

2. Échanger un médoïde à un non-médoïde si la distance intragroupe est minimisée.

3. Recalculer les positions des médoïdes.

4. Assigner chaque non-médoïde au médoïde le plus proche.

5. Répéter les étapes 2 à 4 jusqu'à ce que la position des médoïdes soit fixe, *c.-à-d.* que la distance totale de chacun des éléments à son médoïde soit égale à celle de l'itération précédente.

Ce faisant, chaque itération requiert le calcul de $K(n\text{-}K)$ paires de distances résultant dans une complexité de équivalente de $O(K \times (n-K)^2 \times m \times i)$ (Ng et Han, 2002).

L'algorithme *CLARA* est une modification de l'algorithme PAM, dans lequel ont effectue un tirage aléatoire des éléments :

1. Pour i variant de 1 à 5 répéter les étapes suivantes :

2. Prendre un échantillon de $40 + 2K$ éléments et appliquer l'algorithme PAM.

3. Assigner chaque élément dans le jeu de données au médoïde le plus rapproché.

4. Calculer le coût global et retenir les médoïdes si le coût est plus faible que à l'itération précédente.

5. Retourner à l'étape 1.

Cette formulation, qui requiert seulement l'évaluation de $40+2K$ éléments, résulte en une complexité de $O((K \times (40+K)^2 + K \times (n-K)) \times m \times i)$ soit $O(K^3 + n \times K)$ (Ng et Han, 2002). Pour réduire cette complexité, une version optimisée de l'algorithme, *CLARANS*, a été développée (Ng et Han, 2002). Dans celle-ci, les médoïdes sont choisis à partir d'une liste ordonnée minimisant la recherche de solution, de concert à un nombre maximum d'éléments voisins recherchés, résultant en une complexité de $O(N^2)$. Cependant, dans ce cas, tout comme l'algorithme CLARA, tous les points ne sont pas considérés. Par ailleurs,

l'algorithme de *DBSCAN* utilise une fonction de densité, *c.-à-d.* une distance maximale considérée entre deux points, le rendant capable de travailler sur des formes irrégulières, à l'inverse des méthodes basées sur des heuristiques telles que *k*-means et *k*-medoids. De plus, la recherche heuristique par l'algorithme DBSCAN amène une complexité algorithmique de $O(N \log N)$ lorsqu'un pré-classement des données est considéré, ou de $O(N^2)$ sans ce pré-classement (Ester *et al.*, 1996). Cependant, cet algorithme ne peut être utilisé avec des jeux de données présentant de fortes différences de densité (tels que les flux de travaux). Il permet aussi que certains éléments soient omis ou présents dans plusieurs groupes.

2.10.2 Méthodes hiérarchiques

Les approches hiérarchiques génèrent une hiérarchie de groupes imbriqués (Berkhin, 2006). Elles convergent plus rapidement que les méthodes de partitionnement en ne visitant qu'une fois chaque élément durant leur exécution (Xu et Wunsch, 2005). Deux approches sont alors considérées: l'approche *agglomérative* ou l'approche par *division*. L'approche agglomérative comprend des algorithmes tels que UPGMA (voir Chapitre 1) qui relient les éléments les plus proches jusqu'à ce qu'il ne reste plus d'éléments. L'approche par division va plutôt diviser un groupe initial d'éléments en sous-groupes en se basant encore sur la distance entre ceux-ci. Un algorithme représentatif de cette méthode est l'algorithme de Neighbor-Joining. Les approches agglomératives ont généralement une complexité algorithmique minimale de $O(N^2)$, due au calcul de la matrice de distance inhérente au regroupement (Karypis, 2002; Zhao *et al.*, 2005). Cependant, l'algorithme *CLUTO* (Karypis, 2002), utilisant une distance cosine et une combinaison hybride d'algorithmes agglomératifs et de partitionnements, permet de réduire cette complexité algorithmique à $O(N^{2/3} \log N)$ en divisant, tout au long des itérations, le nombre de groupes à évaluer et le nombre d'éléments dans ces derniers.

Tableau 2.5 Complexités algorithmiques des méthodes de regroupement.

Algorithme	Complexité algorithmique (temps)
k-means	$O(K \times N \times m \times i)$
k-medoids	$O(K \times (n-K)^2 \times m \times i)$
Fuzzy *c*-means	$O(N)$
CLARA	$O((K \times (40+K)^2 + K \times (N-K)) \times m \times i)$
CLARANS	$O(N^2)$
Général hiérarchique*	$O(N^2)$
UPGMA	$O(N^2)$
FITCH	$O(N^4)$
KITSCH	$O(N^4)$
Neighbor-Joining	$O(N^3)$
CLUTO	$O(N^{2/3} \log N)$
Méthodes d'isomorphisme des graphes**	$O(N!)$
Méthode d'édition de graphes**	$O(N^3)$

* Basé sur l'analyse algorithmique de Xu et Wunsch (2005).

** Voir également Riesen et Bunke (2009), Schaeffer (2007) et Zeng *et al.* (2009).

2.11 Détermination du nombre de groupes

Différents critères aident à la détermination du nombre optimal de classes durant la classification par partitionnement. Soit Q_k, la qualité d'un partitionnement correspondant à la

valeur de ce critère de regroupement comprenant k groupes. Certains de ces critères recherchent une valeur maximale de cette valeur Q ou minimale (Tableau 2.5). Par ailleurs, on recherche plutôt une différence maximale, ou minimale, entre différentes valeurs de Q pour certains critères (*p.ex.* LogSS) Cette règle correspond à la différence maximale entre deux valeurs de pentes V (Équation 14). Cette valeur de pente V_i correspond à la différence entre deux valeurs de Q successives (Q_{i-1} et Q_i) (Équation 15) :

$$k = \arg\max(V_i - V_{i-1}), \text{ et} \tag{14}$$

$$V_i = Q_{i+1} - Q_i \ (pentes). \tag{15}$$

Une analyse de 30 différents critères a été réalisée par Milligan et Cooper (1985) qui ont trouvé que le critère de Calinski-Harabasz était le plus représentatif pour définir le nombre de classes optimal. Depuis, de nouveaux critères et des jeux de données plus conséquents ont été étudiés par Arbelaitz et collaborateurs (2013) qui ont déterminé que le critère Silhouette utilisé avec l'algorithme *k*-means permettait une meilleure détermination du nombre de groupes.

Tableau 2.6 Critères de regroupement les plus communs (Arbelaitz *et al.*, 2013).

Critères	Règle	Références
Calinski-Harabasz	Maximum	Calinski et Harabasz (1974)
Silhouette	Maximum	Rousseeuw (1987)
Davies-Bouldin (DB)	Minimum	Davies et Bouldin (1979)
LogSS	Différence minimale	Hartigan (1975)
Ball-Hall	Différence maximale	Ball et Hall (1965)

2.11.1 Indice de Calinski-Harabasz

L'indice Calinski-Harabasz (*CH*, Calinski et Harabasz 1974) est un critère considérant à la fois la distance intergroupe (SS_B) et la variance intragroupe (SS_W) (Équation 16). Ici, K est définie comme le nombre de groupes et n le nombre d'éléments :

$$CH(K) = \frac{SS_B}{SS_W} \times \frac{(n-K)}{(K-1)}. \tag{16}$$

Le calcul du coefficient SS_B (Équation 17) est évalué en calculant la norme L^2 (distance Euclidienne) entre le vecteur représentant les centroids des groupes $mean_k$ et le vecteur *mean*, représentant la moyenne de tous les éléments. Aussi, le coefficient SS_W (Équation 18) peut être calculé de la même façon, où y_{ik} est un vecteur représentant l'élément i dans le groupe k, et n_k le nombre d'éléments dans le groupe k.

$$SS_B = \sum_{k=1}^{K} n_k \left\| mean_k - mean \right\|^2, \text{ et} \tag{17}$$

$$SS_W = \sum_{k=1}^{K} \sum_{i=1_k}^{n_k} \left\| y - mean_k \right\|^2. \tag{18}$$

2.11.2 Indice Silhouette

La distance Silhouette (Rousseeuw 1987) est une mesure de l'appartenance d'un objet à son groupe, plutôt qu'au groupe le plus proche (Kaufman et Rousseeuw 1990). Pour chaque élément i, $a(i)$ est la distance moyenne entre i et tous les autres éléments du groupe C_i. Ainsi, pour chaque groupe sauf C_i, la distance $d(i,C)$ est la distance moyenne entre i et les éléments dans le groupe C. Alors, $b(i)$ est la plus petite de ces distances. L'indice Silhouette est défini comme suit (Équation 19) :

$$s(k) = \left[\sum_{i=1}^{n_k} \frac{b(i)-a(i)}{\max(a(i),b(i))} \right] / n_k. \tag{19}$$

Et l'indice global pour une classe K pour un ensemble de groupes est la moyenne de tous les indices Silhouette individuels (Équation 20) :

$$\bar{s}(K) = \sum_{k=1}^{K} [s(k)] / K. \tag{20}$$

2.11.3 Indice LogSS

L'indice LogSS (Hartigan, 1975) estime la variance entre les groupes (Équation 21) sans correction sur la taille de ces groupes, à l'inverse de l'indice de Calinski-Harabasz. Les variances intergroupes et intragroupe se calculent de la même manière dans les Équations 17 et 18.

$$\log SS(K) = \log \frac{SS_B}{SS_W}. \tag{21}$$

2.11.4 Autres indices de regroupement

Plus de 40 indices ont été répertoriés jusqu'à maintenant (Arbelaitz *et al.,* 2013; Desgraupes, 2013; Milligan et Cooper, 1985). Parmi les autres indices employés, l'indice de Ball-Hall (Ball et Hall, 1965) mesure la dispersion moyenne des groupes en fonction de la racine carrée moyenne des points, soit la distance des éléments par rapport à leurs centres respectifs (distance intragroupe). En ajoutant la distance intergroupe, on obtient l'indice de Davies-Douldin (Davies et Bouldin, 1979), qui est une méthode mesurant la séparation intergroupe et la distance intragroupe. Dans cet indice, la moyenne de la somme des distances entre chaque paire de points composant deux groupes, en fonction de leurs centres respectifs est prise en compte. On mesure donc la dispersion des deux groupes divisée par la distance de leurs deux centres. Finalement, l'indice Dunn (Dunn, 1973) est basé sur la racine carrée de la distance minimale entre deux groupes (mesure de la séparation intergroupe) qui est divisée par la racine carrée de la distance maximale entre deux points du même groupe (mesure de l'encombrement intragroupes). Ainsi, cette mesure de la séparation est fonction des deux points les plus rapprochés entre les deux groupes (Arbelaitz *et al.,* 2013). Cependant, puisque l'on ne tient pas compte de la moyenne des données, l'indice de Dunn est très susceptible au bruit aux points aberrants (*outliers*). De plus, si les groupes sont très rapprochés, la représentativité de l'indice sera très faible. De fait, on recommande normalement de comparer plusieurs indices pour trouver le nombre de groupes optimal puisque aucun indice ne convient à tous les jeux de données (Everitt *et al.,* 2001; Kaufman et Rousseeuw, 1990; Xu et Wunsch, 2005).

2.12 Comparaison des partitions : l'indice Rand

Une fois le nombre de groupes déterminé et le regroupement réalisé, on peut utiliser un critère pour comparer ce résultat à une partition de référence. On utilise alors différentes mesures de distance telles que l'indice non paramétrique de McNemar (1947), l'indice *F* ou indice de Czekanowski-Dice équivalent (*F-measure*; van Rijsbergen, 1979), ou encore l'indice de Folkes-Mallows (Fowlkes et Mallows, 1983) (revue par Desgraupes *et al.*, 2013).

Cependant, l'indice le plus utilisé est l'indice Rand (Rand, 1971). Dans celui-ci (Équation 22), a est le nombre de paires d'éléments se retrouvant dans le même groupe dans les partitions U et V, b est le nombre de paires d'éléments qui sont ensemble dans U mais non dans V, c est le nombre d'éléments qui sont ensemble dans V mais pas dans U, et d est le nombre d'éléments qui sont dans des groupes différents dans les deux partitions (Rand, 1971) :

$$RI = \frac{a+d}{a+b+c+d}. \tag{22}$$

2.13 Conclusion

Dans ce chapitre, nous avons exploré les environnements de création des flux de travaux et différentes approches et algorithmes permettant de les comparer. Nous introduirons, dans le prochain chapitre, une nouvelle plate-forme locale de flux de travaux, *Armadillo*. Cette plate-forme permet la recherche d'hypothèses en facilitant l'utilisation de plusieurs tâches et méthodes dans la même expérimentation. Cette plate-forme intègre des structures de contrôle (boucles *For* et fonctions conditionnelles *If*) et comprend différents éléments pouvant aider à la comparaison des flux de travaux (*p.ex.* types définis, temps d'exécution conservés).

CHAPITRE III

LA PLATE-FORME ARMADILLO

> L'Armadillo ou tatou *(Dasypus novemcinctus)* est un animal des Amériques présentant 9 bandes (du Latin *novemcinctus*) sur sa carapace. Le nombre de ces bandes peut varier de 7 à 11 et permet à celui-ci de se protéger en se mettant en boule. Pratiquement omnivore, il est aussi extraordinaire par son mode de reproduction dans lequel quatre jumeaux génétiquement identiques sont mis au monde après division d'un seul ovule fécondé. – Lecointre et Guyader (2001)

3.1 Préface

Dans ce chapitre, nous introduirons une nouvelle plate-forme de flux de travaux, développée durant ce projet doctoral. Cette plate-forme, dédiée à la recherche en phylogénétique et phylogénomique, prône l'exécution locale des tâches et de programmes bioinformatiques populaires (Tableau 1.1). Contrairement à d'autres plates-formes telles que Taverna et Galaxy, la plate-forme *Armadillo* incorpore directement des structures de contrôle (*control-flow*) dans les flux de travaux. Un certain nombre d'outils bioinformatiques et phylogénétiques généraux a été inclus dans la première version du logiciel. Comme *Armadillo* est un projet « open-source », il permet aussi aux scientifiques de développer leurs propres modules ainsi qu'intégrer de nouvelles applications informatiques. Grâce à cette plate-forme de flux de travaux, une série de tâches phylogénétiques complexes peut être modélisée sans aucune connaissance préalable de la programmation. La première version de *Armadillo* a été utilisée avec succès par des professeurs de bioinformatique de l'Université du Québec à Montréal dans le cadre des cours de biologie computationnelle offerts aux cycles supérieurs en 2010-2011. Le programme et son code source sont disponibles gratuitement à l'adresse url : <http://www.bioinfo.uqam.ca/armadillo>. Une partie du texte présenté a fait l'objet d'une publication dans la revue PloS One (Lord *et al.*, 2012).

3.2 Introduction

La bioinformatique est un domaine en rapide évolution qui englobe la biologie moléculaire, la biochimie, la science de l'informatique et des statistiques (Oinn *et al.,* 2006). Elle a émergé en raison de l'augmentation dramatique et de la complexité des données génomiques disponibles (Hoon *et al.,* 2003). La phylogénétique, qui est un sous-champ de la bioinformatique et de la biologie moléculaire, qui étudie les relations évolutives entre les organismes en fonction de leur proximité moléculaire, ou encore de leurs différences morphologiques. Elle présente ces relations à travers des représentations appelées arbres phylogénétiques (ou phylogénies) (Felsenstein, 2004).

Le développement d'une variété d'algorithmes phylogénétiques a conduit à la conception de nombreuses applications informatiques générant souvent des résultats différents lors de la résolution du même problème de biologie computationnelle (Hoon *et al.,* 2003; Stevens *et al.,* 2007). Ainsi, la modélisation et la conduite de simulations bioinformatiques *in silico* peut être une tâche très difficile en raison de cette quantité et cette diversité des outils disponibles, mais encore du nombre de banques de données regroupant les informations génomiques qu'un chercheur doit consulter. Souvent, des tutoriels et des exemples sont distribués avec les applications phylogénétiques et bioinformatiques, tandis que des cours de formation sont offerts sur le Web et peuvent être trouvés dans différents annuaires, tel que le *Bioinformatique Link Directory* (Brazas *et al.,* 2010). Cependant, aucune « pratique standard » pour la bioinformatique et l'analyse phylogénétique n'a encore été strictement définie à l'exception de domaines très spécifiques, voir Yang (2007) ou Swofford (2002). Chaque étape d'une analyse peut ainsi être effectuée en utilisant une variété de méthodes et d'outils (Hoon *et al.,* 2003). De plus, tout en réalisant leurs expériences et leurs simulations, les bioinformaticiens doivent faire face aux limitations des différents logiciels et à des questions d'intégration de données (Oinn *et al.,* 2006). En outre, des résultats erronés de l'analyse de données biologiques peuvent survenir lorsque des outils et des modèles accessibles, mais non-adéquats, sont utilisés pour le traitement des données (Wong *et al.,* 2008).

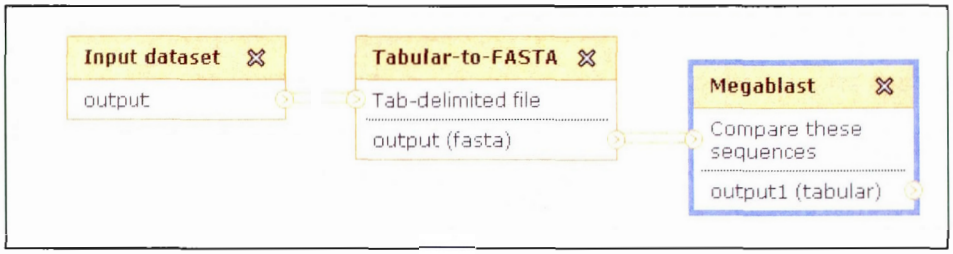
De même, une augmentation dramatique des données génomiques et phylogénétiques cause une demande accrue pour des logiciels permettant la création et la gestion de pipelines d'analyse traitant ces données d'une manière automatisée. Par exemple, Cicarelli et al. (2006) ont développé une procédure automatique pour reconstruire un arbre phylogénétique avec des longueurs de branches incluant des espèces des trois règnes de la vie. Cette procédure peut être représentée comme un flux de travaux bioinformatique typique englobant les principales tâches suivantes (voir également la Figure 1 de Ciccarelli et al., (2006)): (1) sélection et préparation des familles de gènes marqueurs, (2) production de l'alignement de séquences, (3) concaténation des séquences (en une super-matrice), (4) détection systématique et l'élimination des transferts horizontaux de gènes, (5) reconstruction de l'arbre phylogénétique, et (6) évaluation de la phylogénie inférée. Philippe et collaborateurs (2001) avertissent toutefois que l'analyse phylogénétique automatique a ses propres limitations et mettent ainsi en garde les utilisateurs de tels systèmes contre les pièges les plus fréquents. En conséquence, toutes les données générées automatiquement devraient être systématiquement vérifiées « manuellement » et corrigées si nécessaires, avant de procéder à la poursuite de leur traitement ou de leur interprétation.

Ainsi, une tâche typique de la bioinformatique peut être décrite comme un pipeline dans lequel les ressources ou les données sont traitées successivement par une série d'outils dédiés (Oinn *et al.,* 2006). Un pipeline ou *flux de travaux* comprend généralement trois étapes: (1) l'acquisition des données, (2) l'analyse des données et (3) une étape consistant en la génération d'un rapport traitant des résultats (Stevens *et al.,* 2007). La Figure 3.1 présente un exemple d'une telle tâche bioinformatique, soit la recherche de séquences d'ADN semblables à une séquence cible. Une telle tâche requiert une combinaison de trois étapes: (1) l'entrée d'une séquence d'ADN donnée dans un format de données particulier, (2) sa conversion vers un format compatible en fonction des outils utilisés, et enfin, (3) l'exécution d'une requête vers une base de données en ligne ou une base de données locale appropriée à l'aide d'un algorithme de comparaison de séquences tel que l'algorithme BLAST (Johnson *et al.,* 2008).

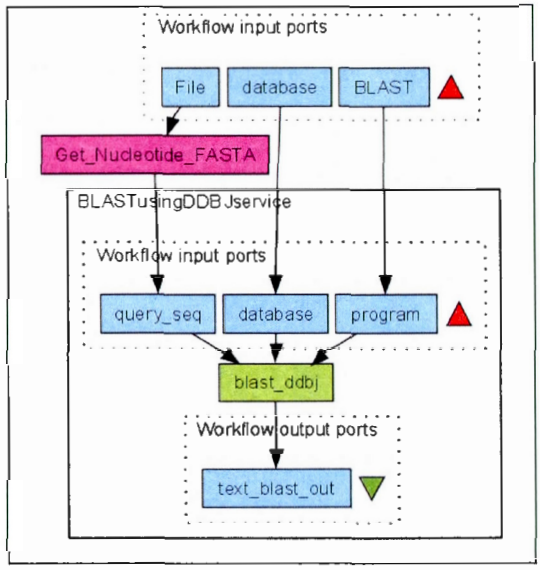
A

```
load_fasta_sequence.pl < new.dna | megablast.pl
```

B



C



D

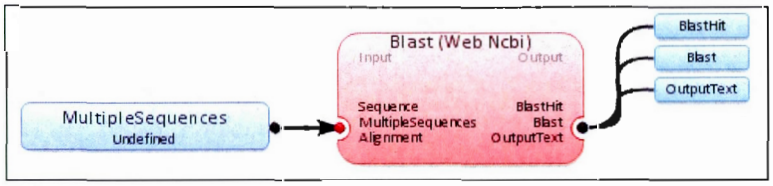


Figure 3.1 Comparaison de quatre manières différentes d'effectuer une recherche de séquences (*c.-à-d.* avec l'utilisation de l'algorithme BLAST dans chacune). Le panneau (A) représente un pipeline standard utilisant un script en langage *Perl*; Ces panneaux (B, C, D) montrent différents flux de travaux représentant la même opération dans les plates-formes Galaxy (B) (Giardine *et al.*, 2005), Taverna (C) (Oinn *et al.*, 2007) et la nouvelle plate-forme *Armadillo* (D).

Par conséquent, la construction d'études bioinformatiques rigoureuses nécessite l'intégration de plusieurs outils informatiques, qui peuvent inclure des applications publiques ou commerciales. Les principaux problèmes peuvent alors survenir lors de leur intégration dans un seul pipeline: des protocoles de communication incompatibles entre les applications (*p.ex.* des formats de fichiers ou des options de ligne de commande incompatibles), les conditions matérielles requises pour les algorithmes de chacune des applications sont non-spécifiées, l'accès électronique limité ou inexistant à des banques de données biologiques publiques ou privées, les logiciels fonctionnant sous différents systèmes d'exploitation (*OS*) et des rapports de résultats non normalisés qui sont générés par les différentes applications (Stevens *et al.*, 2007). Les scientifiques étudiant les sciences de la vie, qui n'ont aucune connaissance préalable en informatique, sont alors soit limités à une utilisation de base des outils existants, ou soit doivent apprendre un langage *script*, tel que *Perl* (Figure 3.1A), ou un langage de programmation, tels que le langage *C* ou le langage *Java* (Dudley et Butte, 2009), afin de mettre en œuvre leurs expériences.

Quelques outils informatiques ont été mis en place pour aider les chercheurs en bioinformatique à effectuer ces tâches de plus en plus complexes: (1) des applications Web telles que Galaxy (Giardine, 2005) (Figure 3.1B) ou la plate-forme ENSEMBL de la *European Bioinformatics Institute platform* (EBI) (Goujon *et al.*, 2010) qui permettent aux utilisateurs l'accès à des grappes de calculs multiprocesseurs à l'aide d'une interface conviviale. De plus, des portails disponibles uniquement sur le Web tels que *Phylogeny.fr* (Dereeper *et al.*, 2008), *Phylemon* (Sànchez *et al.*, 2011) et *Bioextract.org* (Lushbough *et al.*, 2011), permettent la création de pipelines d'exécution simples destinés à l'inférence phylogénétique (*c.-à-d.* les boucles et les opérations conditionnelles ne sont pas permises dans ces pipelines) ou encore le serveur Web AIR (Kumar *et al.*, 2009) destiné à la réalisation d'analyses phylogénomiques utilisant une méthode de *super-matrices*; (2) des bibliothèques ou *API*[14] de programmation (*BioPerl* (Stajich *et al.*, 2002), *Biojava* (Holland *et al.*, 2008), etc) écrites en langages *scripts* ou de programmations populaires pour faciliter les

[14] Interface de programmation d'applications

tâches de programmation; (3) des applications multitâches (*c.-à-d.* des applications fusionnant différents programmes) ou progiciels, combinant une variété d'algorithmes bioinformatiques généraux, tels que MEGA (Tamura *et al.,* 2011), Geneious (Drummond *et al.,* 2009) et Mesquite (Maddisson et Madisson, 2011). Les exemples d'analyses bioinformatiques mis à la disposition des chercheurs via les trois dernières applications comprennent : l'alignement de séquences, l'inférence phylogénétique, l'assemblage des *shorts reads*[15], l'analyse de la recombinaison, la reconstruction des états ancestraux des séquences, la simulation de l'évolution des caractères et la détection de la coalescence profonde. De la même manière, PAML (Yang, 2007), PAUP (Swofford, 2002), PHYLIP (Felsenstein, 1989) et T-REX (Boc *et al.*, 2012) sont des applications multifonctionnelles spécialisées axées sur l'analyse phylogénétique.

Alors que le premier et le troisième types d'outils s'appuient sur la manipulation de formats de données diverses et l'exécution manuelle d'applications différentes pour répondre aux questions biologiques, le deuxième type d'outil, *c.-à-d.* les API de programmation, nécessitent une connaissance approfondie des langages de programmation. Ainsi, une approche émergente dans la communauté des bioinformaticiens est le développement de plates-formes de *workflows* ou flux de travaux (Figures 3.1B-3.1D) (Romano, 2008), qui peuvent être utilisées pour la conception et l'exécution d'études de simulations, l'analyse d'échantillons ou à des fins éducatives.

Ces plates-formes de flux de travaux sont constituées de processus ou d'applications connectées. Elles ont été initialement mises en place et utilisées dans le domaine financier et les environnements d'affaires (Oinn *et al.,* 2007; Woollard *et al.,* 2008). Les flux de travaux ont également été employés activement pour relier en chaine des logiciels spécialisés dans le but de créer un flux de données (*data-flow*) intégré au processus de développement de certains logiciels, et aussi d'applications scientifiques (Oinn *et al.,* 2007). Ils fournissent ainsi un environnement propice à la définition, à la gestion et la coordination des activités de traitement de données (Lin *et al.,* 2001). Les plates-formes de gestion de flux de travaux ont

[15] Courts segments d'ADN produits par le séquençage de nouvelle génération.

ainsi évoluées de simples gestionnaires d'exécution à des systèmes complexes permettant l'exécution conditionnelle de flux de données et la répartition de tâches en fonction d'une logique temporelle ou en fonction des ressources disponibles (Oinn *et al.,* 2006; Giardine *et al.,* 2005). Ces plates-formes permettent également de simplifier l'automatisation de tâches sujettes à l'erreur, la collecte de données, la *refactorisation* et l'organisation des entrées / sorties, le traitement des résultats et une représentation graphique de ceux-ci (Oinn *et al.,* 2006; Beaulah *et al.,* 2008). Les exemples de plates-formes de flux de travaux dédiés au domaine de la bioinformatique comprennent la plate-forme Web Galaxy (Giardine *et al.,* 2005) (Figure 3.1B) et la plate-forme locale *Taverna* (Oinn *et al.,* 2007) (Figure 3.1C). Ces dernières contiennent un langage de descriptions des flux de travaux spécifique ainsi qu'un modèle d'automatisation spécifique. Toutefois, tandis que la plate-forme Galaxy est basée sur une architecture de serveurs privés, Taverna s'appuie sur des services Web développés et maintenus par des tiers.

Des études antérieures ont démontré que les plates-formes de flux de travaux peuvent également être utiles dans la conceptualisation de solutions, tout en facilitant l'apprentissage « *just-in-time* » par leur pouvoir de démonstration (Ma *et al.,* 2010). Ainsi, plusieurs plates-formes de flux de travaux ont été consacrées au domaine de l'éducation (Lin *et al.,* 2001; Ma *et al.,* 2010; Deelman *et al.,* 2005; Gil *et al,.* 2011; van der Veen *et al.,* 2000). Elles fournissent aux enseignants différents protocoles, les aidant à créer du contenu pédagogique en ligne (*e-learning*, l'apprentissage à distance) ou local (Lin *et al.,* 2001; Vouk *et al.,* 1999), tout en gardant la trace des résultats, des échecs et des tentatives de résolution de problèmes des étudiants (Vouk *et al.,* 1999). La plupart des flux de travaux éducationnels sont ainsi conçus pour améliorer la cognition des élèves, soit vérifier la qualité du processus d'apprentissage (Zhang et Li, 2009). Au meilleur de notre connaissance, aucune plate-forme de flux de travaux consacrée à l'éducation de la bioinformatique n'a encore été proposée.

Ainsi, le développement d'une plate-forme de flux de travaux encapsulant des algorithmes bioinformatiques et des formats de données populaires dans cette discipline, serait très important pour ce domaine en expansion. Une telle plate-forme permettrait aux étudiants d'ignorer la programmation et « *la cuisine* », nécessaires à cette science, pour se concentrer directement sur les objectifs réels de leurs projets. Cette plate-forme devrait idéalement

satisfaire à un principe du *WYSIWYG* (*What You See Is What You Get*), ou plus précisément dans notre cas, à un principe du *WYPIWYG* (*What You Pipe Is What You Get*) (Hoon *et al.*, 2003).

Nous avons donc développé *Armadillo*, une plate-forme de flux de travaux originale dédiée à la modélisation phylogénétique complexe et aux expériences générales en bioinformatique. *Armadillo* permet une exécution locale d'applications phylogénétiques populaires. Ce faisant, elle permet aux utilisateurs de concevoir rapidement et de mettre à une plus grande échelle (*scale up*) des expériences de biologie computationnelle, de faciliter la gestion de différents formats de données, d'effectuer des transactions électroniques avec les principales banques de données biologiques et la conversion automatique des entrées et sorties de plusieurs logiciels du domaine. De plus, la plate-forme proposée présente une interface utilisateur graphique (*GUI*), dans laquelle les applications disponibles sont représentées de manière classique sous forme de boîtes simples reliées en pipeline par des boîtes d'interconnexions. Ces interconnexions représentent des évènements d'acquisition de données et permettent la conception d'un flux impliquant différentes applications phylogénétiques (voir Figure 3.1D). Dans les sections suivantes, nous allons décrire les caractéristiques distinctes de la plate-forme *Armadillo* qui peuvent être utiles pour effectuer des simulations, réaliser des inférences phylogénétiques et faciliter apprentissage de la bioinformatique.

3.3 Conception et implémentation

3.3.1 Description générale de *Armadillo*

La plate-forme de flux de travaux *Armadillo* (*version 1.1*) a été développée en langage de programmation *Java* à l'aide de la librairie *Processing*[16] (Fry, 2004) pour la gestion des opérations graphiques (Figure 3.2). De même que pour les autres plates-formes de flux de travaux bioinformatiques existantes, telles que Taverna ou Galaxy, les composantes représentant soit des ensembles de données, des méthodes, ou des logiciels bioinformatiques peuvent être liées entre elles pour créer un flux de données en effectuant des opérations de

connexion « *drag-and-drop* » entre les boîtes (Figures 3.2A et 3.2B). La plate-forme *Armadillo* intègre une visualisation des séquences d'ADN, d'ARN et de protéines (Figure 3.2D) et permet d'accéder à des logiciels d'interférence d'arbres phylogénétiques et à des applications permettant la manipulation de ces arbres (Figures 3.2A et 3.2C). La configuration de chaque application s'effectue via une boîte de dialogue personnalisée (Figure 3.2E) qui permet d'accéder aux fonctions les plus couramment utilisées. *Armadillo* ne nécessite pas un accès à Internet pour la plupart de ses opérations. Elle fonctionne nativement sous le système d'exploitation Windows et Mac OS X Leopard et Lion[17] et comprend plusieurs applications pré-compilées pour ces systèmes. Les exigences matérielles minimales du système sont indiquées sur le site de l'application (pour plus de détails, voir: http://www.bioinfo.uqam.ca/armadillo). L'installation d'*Armadillo* peut être effectuée en une seule étape qui comprend la mise en place des nombreux logiciels phylogénétiques (voir le Tableau 3.1). Certes, l'utilisateur doit faire face aux paramètres de mémoire particuliers et aux limites imposées par les applications bioinformatiques incluses dans la plate-forme, tout comme un manque de mémoire vive (*overflow*) qui peut être causé par le chargement ou l'exécution de grands ensembles de données par ces applications. En utilisant l'option *Preferences* → *Advanced*, l'utilisateur peut spécifier la quantité de mémoire *RAM* qui peut être utilisée par *Java* lors du chargement de la plate-forme.

De même, une fois exécuté, plusieurs informations sont recueillies par la plate-forme *Armadillo* (Figure 3.3) telles que la taille des données, le temps d'exécution des méthodes, et l'ordre d'exécution des différentes tâches. Cet ordre d'exécution se fait présentement à l'aide d'un tri topologique suivi d'une exécution selon un modèle *myopic* (Wieczorek *et al.* 2005). Par contre, des modèles d'exécution tels que l'algorithme *Heterogeneous Earliest Finish Time* (HEFT) (Zhao et Sakellariou, 2003; Wieczorek *et al.* 2005) ou *l'algorithme dynamic critical path* (DCP; Kwok et Ahmad, 1996) pourraient être implémentés.

---

[16] http://processing.org
[17] Et plus récent tel que *Yosemite*.

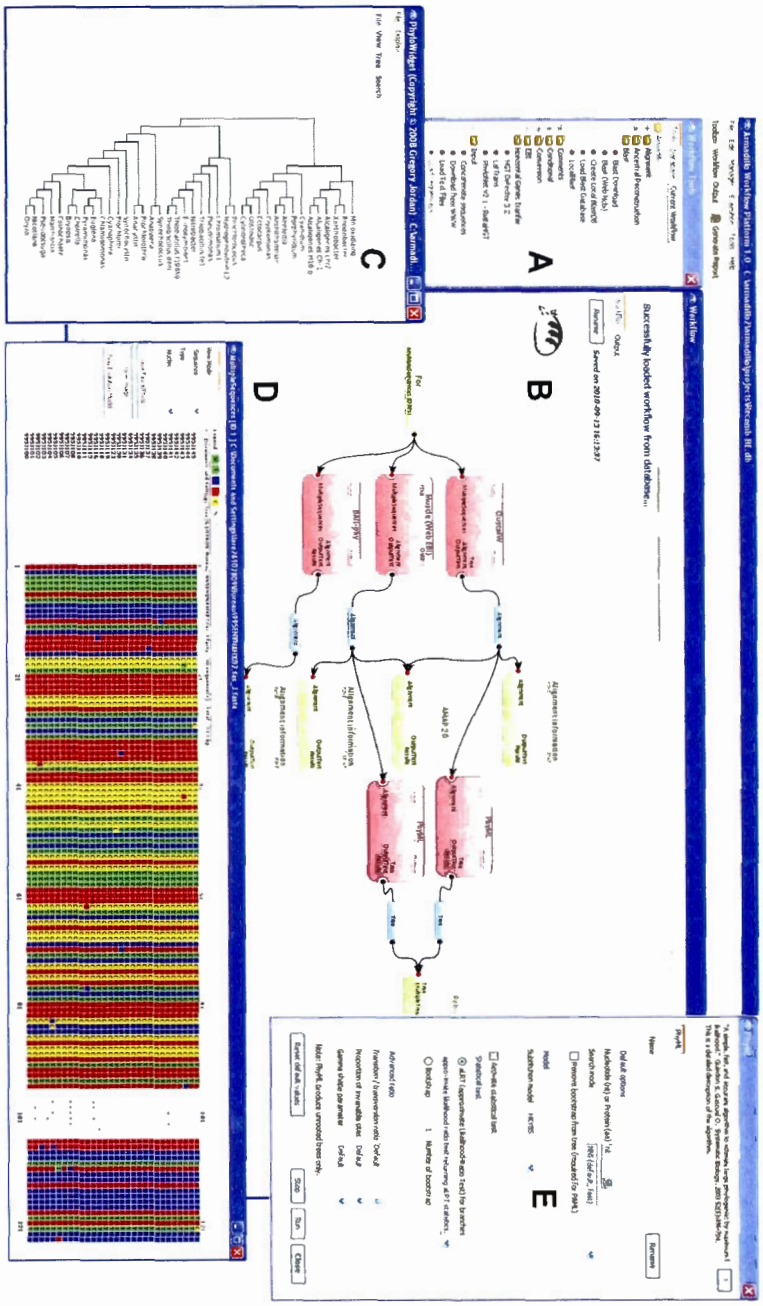


Figure 3.2 Aperçu de l'interface graphique (GUI) de la plate-forme Armadillo. Le panneau (A) présente les outils disponibles. Tous les outils peuvent être utilisés comme des composants pouvant être ajoutés par *drag-and-drop* dans les flux de travaux. Le panneau (B) présente la vue de création d'un flux de travaux (*WFCA*). Le panneau (C) présente une

représentation d'un arbre phylogénétique en utilisant le logiciel PhyloWidget (Jordan et Piel, 2008). Le panneau (D) présente une visualisation possible des séquences. Le panneau (E) présente une boîte de dialogue affichant des options de configuration pour un composant du flux de travaux.

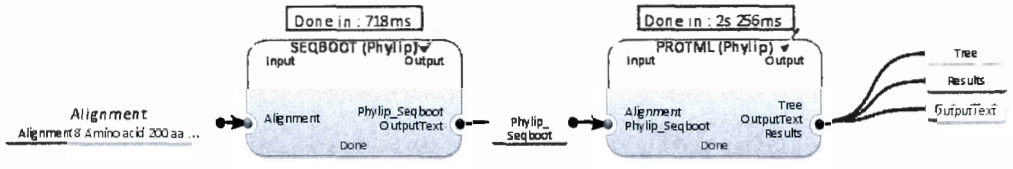


Figure 3.3 Vue d'un flux de travaux après exécution dans la plate-forme Armadillo.

Tel que décrit par Stevens *et al.* (2007), le cycle de vie des expériences *in silico* consiste en différentes étapes clés qui doivent être abordées : (1) la conception expérimentale initiale, (2) l'exécution, (3) l'interprétation des résultats obtenus, et (4) la publication de rapports pouvant être intégrés à des publications. La plate-forme Armadillo a été construite et conçue de manière à faciliter un bon nombre de ces tâches. Toutes les entrées et sorties des applications sont sauvegardées dans un seul fichier de stockage du « projet ». Ce fichier de stockage est mis en œuvre sous la forme d'une base de données SQLite[18] utilisant le langage de base de données : *Structured Query Language* (SQL). L'utilisation de celle-ci s'effectue via la librairie en *Java Xerial* (Hipp et Kennedy, 2003). Cette base de données peut également être affichée directement sur un site Web pour permettre aux utilisateurs de partager leurs résultats, ou encore de comparer différentes stratégies de conception des flux de travaux. Cependant, le fait que cette plate-forme locale ne prend pas en charge directement de bases de données plus conséquentes (*p.ex.* MySQL[19]) peut sembler étrange à notre époque où l'informatique en nuage est de plus en plus présente (Yuan *et al.*, 2010). Néanmoins, le but principal de ce projet était de faciliter la conception de flux de travaux phylogénétiques sans avoir à créer des utilisateurs, de faire la gestion de groupes d'utilisateurs, ou d'assurer un

[18] http://sqlite.org
[19] http://www.oracle.com/us/products/mysql/overview/index.html

accès sécurisé aux données. Toutefois, nous considérons que toutes ces options pourraient faire l'objet d'extensions possibles de notre plate-forme de flux de travaux. Un seul système de fichiers, utilisé dans la plate-forme *Armadillo*, facilite également l'organisation du matériel de cours en permettant une création rapide d'exercices, de jeux de données et de schémas de travail. Des fichiers textes et *HTML* peuvent être ajoutés directement à n'importe quel flux de travaux créé. La plate-forme prend également en charge l'annotation des composants individuels, et peut ainsi permettre aux enseignants, étudiants et chercheurs de commenter les résultats de leurs analyses. Enfin, l'architecture de *Armadillo* peut également être utilisée lors d'un cours portant sur le langage *SQL* puisque toutes les requêtes sont affichées et exécutées directement dans la plate-forme.

3.3.2 Applications incluses dans la version 1.1 de *Armadillo*

La première version de la plate-forme *Armadillo* supporte de nombreux formats de données de séquences via l'inclusion de l'application *ReadSeq,* développée par Gilbert (2010). La visualisation d'arbres phylogénétiques peut être effectuée en utilisant soit l'application PhyloWidget (Jordan et Piel, 2008) soit l'application Archaeopteryx (Zmasek et Eddy, 2001) capables de lire les fichiers d'arbres aux formats *XML*, *phyloXML*, *Newick* et *Nexus*. Plusieurs des outils les plus populaires pour résoudre les problèmes fondamentaux en phylogénétique, tels que la reconstruction de l'évolution des espèces à partir de séquences moléculaires, plusieurs algorithmes d'alignement de séquences et des comparaisons de séquences par BLAST constituent les applications de base incluses dans Armadillo (voir le Tableau 3.1 pour la liste complète des applications incluses). Les logiciels PAML (Yang, 2007), DNAML et PROTML (du *package* PHYLIP; Felsenstein, 1989), permettant des analyses avec différents modèles d'évolution, sont fournis directement dans la plate-forme *Armadillo*. De même, le logiciel Gblocks (Talavera et Castresane, 2007; Castresana, 2000) peut être utilisé pour améliorer la qualité des alignements de séquences en supprimant les blocs alignés divergents ou ambigüs. Des fonctions personnalisées peuvent également être ajoutées au flux de travaux via la compilation et l'exécution de code source *Java*. Cette fonctionnalité est disponible via le menu *Tools → Your program →Custom*, de la boîte à outils. Une description des fonctions de base de chaque application incluse dans Armadillo

est accessible en cliquant sur le bouton *Information* de l'application (ce bouton est représenté par le symbole « ? »).

Les résultats générés par une application, après l'exécution, sont accessibles et peuvent être vérifiés directement à partir de la vue principale. *Armadillo* procède par la validation des résultats de chaque application lors de l'exécution du flux de travaux. Une fois l'exécution du flux de travaux terminée, un rapport complet est généré pour présenter la description détaillée des différentes étapes, ainsi que les résultats correspondants (Figure 3.3 et 3.4). Ce rapport comprend tous les résultats obtenus, l'état des applications, ou les erreurs rencontrées (le cas échéant) et les sorties d'applications obtenues à toutes les étapes intermédiaires de l'analyse. Les rapports sont présentés à travers des hyperliens vers des fichiers *HTML* associés. Un exemple d'une tâche complexe, que nous avons utilisé dans nos simulations, est présenté sur la Figure 3.4. Dans ce flux de travaux, des applications permettant des alignements de séquences (*c.-à-d.*, Muscle (Edgar, 2004) et ProbCons (Do *et al.*, 2005) fournissent les alignements de séquences à l'entrée pour l'algorithme de reconstruction d'arbres phylogénétiques PhyML (Guindon et Gascuel, 2003). Par la suite, des applications de validation ou de comparaison des topologies d'arbres obtenues peuvent être ajoutées au flux de travaux.

Tableau 3.1 Applications bioinformatiques incluses dans la plate-forme Armadillo.

Tâches bioinformatiques	Applications et services
Bases de données distantes *National Center for Biotechnology Information (NCBI)*	Accès à la recherche et au téléchargement de données en utilisant le service Web EUtils[b]
ENSEMBL - European Bioinformatics Institute (EBI)	Accès à la recherche en utilisant EB-Eye (Valentin *et al.*, 2010) et au téléchargement de données avec dbFetch (Goujon *et al.*, 2010)
HUGO Gene Nomenclature Committee	Accès à la recherche et au téléchargement de données du génome humain
Alignement de séquences multiples	BAli-phy (Suchard et Redelings, 2006), ClustalW (Thompson *et al.*, 1994), ClustalW2 (Larkin *et al.*, 2007), Kalign (Lassmann et Sonnhammer, 2005), Muscle (Edgar, 2004), ProbCons (Do *et al.*, 2005)
Détection des transferts horizontaux de gènes	HGT Detection (Boc *et al.*, 2010), PhyloNet – RIATA_HGT (Than *et al.*, 2008), LatTrans (Addario-Berry *et al.*, 2003)
Inférence d'arbres phylogénétiques	fastDNAml (Olsen *et al.*, 1994), DNAml-Erate (Rivas et Eddy, 2008), Garli (Zwickl, 2006), MrBayes (Ronquist et Huelsenbeck, 2009), PhyML (Guindon et Gascuel, 2003), PHYLIP (11 applications incluses) (Felsenstein, 1989), génération d'arbres phylogénétiques aléatoires (Makarenkov, 2001)
Visualisation d'arbres phylogénétiques	PhyloWidget (Jordan et Piel, 2008), Archaeopteryx (Zmasek et Eddy, 2001), ScripTree (Chevenet *et al.*, 2010)
Sélection de modèles d'évolution des séquences	jModelTest (Posada, 2008), ProtTest (Abascal *et al.*, 2005)
Analyse de la pression sélective	PAML v4.4 (Yang, 2007)
Alignement de séquences et recherche par BLAST	BLAST (Local et Web via *EBI* et *NCBI* (Johnson *et al.*, 2008))

[a] La liste des applications présentement incluses est disponible à l'adresse: http://adn.bioinfo.uqam.ca/armadillo/included.html.
[b] NCBI EUtil est disponile au : http://www.ncbi.nlm.nih.gov/books/NBK55693/

Il est à noter que les flux de travaux scientifiques gèrent généralement de grandes quantités de données (Yuan *et al.*, 2010). Ils peuvent ainsi profiter des fonctions du Web 2.0 soit pour l'acquisition de ces données par l'accès à des bases de données spécialisées, par

l'intermédiaire de services informatiques distribués. Cela permet aux utilisateurs de libérer leurs postes de travail pour exécuter d'autres analyses. Nous avons décidé d'inclure, dans cette première version, l'accès aux trois principaux fournisseurs de données et de services Web: le National Center for Biotechnology Information (*NCBI*), l'Institut Européen de Bioinformatique ENSEMBL (*EBI*), et le Wellcome Trust Sanger Institute (*WTSI*) (voir le Tableau 3.1).

Dans le Tableau 3.2 (voir également le Tableau 2.3), nous comparons les principales caractéristiques de la plate-forme *Armadillo* avec d'autres plates-formes bioinformatiques populaires de flux de travaux, y compris Ergatis (Orvis *et al.*, 2010), Galaxy (Giardine *et al.*, 2005), Kepler (Altintas *et al.*, 2004), LONI (Dinov *et al.*, 2011) et Taverna (Oinn *et al.*, 2007). Spécifiquement, nous comparons le processus de conception d'un flux de travaux, l'organisation des données expérimentales et la possibilité d'ajouter de nouvelles applications.

Tableau 3.2 Comparaison des caractéristiques de la plate-forme *Armadillo* v1.1 aux autres plates-formes de flux de travaux bioinformatiques : Taverna, Galaxy, LONI, Ergatis et Kepler.

	Design des flux de travaux		**Organisation des données**				**Ajouts à la plate-forme**	
Plate-formes	*Drag and drop*	Expression conditionnelle/ Itération	Modèle client-serveur	Organisation des données	Fonction de recherche	Répétition d'expériences	Plate-forme ouverte	Ajouts de nouvelles applications
Armadillo	x	x		x	x	x	x	x*
Taverna		x				x	x	x*
Galaxy	x		x	x	x	x		
LONI	x	x	x	x	x	x		x
Ergatis	x	x	x			x	x	x
Kepler	x	x				x	x	x

* Ajout par l'utilisation de services Web ou par la programmation en langage *Java*. Un x indique la présence de la caractéristique pour cette plate-forme.

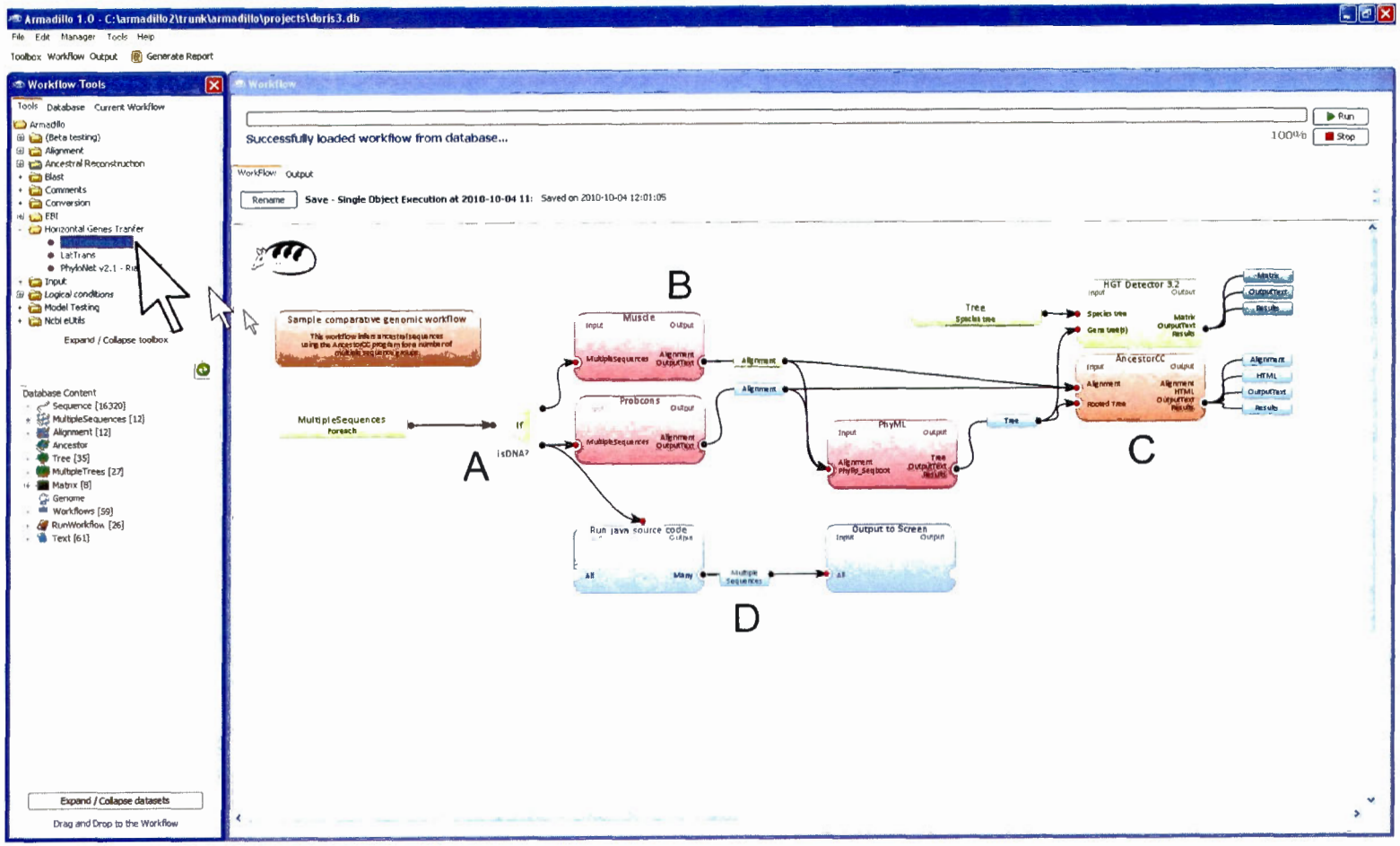

Armadillo 1.0 - C:\armadillo2\trunk\armadillo\projects\doris3.db
File Edit Manager Tools Help
Toolbox Workflow Output Generate Report
Workflow Tools
Tools Database Current Workflow
Armadillo
(Beta testing)
Alignment
Ancestral Reconstruction
Blast
Comments
Conversion
EBI
Horizontal Genes Tranfer
LatTrans
Input
Logical conditions
Model Testing
Ncbi eUtils
Expand / Collapse toolbox
Database Content
Sequence [16320]
MultipleSequences [12]
Alignment [12]
Ancestor
Tree [35]
MultipleTrees [27]
Matrix [8]
Genome
Workflows [59]
RunWorkflow [26]
Text [61]
Expand / Collapse datasets
Drag and Drop to the Workflow
Workflow
Successfully loaded workflow from database...
Run
Stop
Workflow Output
Rename
Save - Single Object Execution at 2010-10-04 11: Saved on 2010-10-04 12:01:05
Sample comparative genomic workflow
MultipleSequences
foreach
If
isDNA?
A
B
Muscle
Probcons
PhyML
Tree
Species tree
HGT Detector 3.2
AncestorCC
C
Run Java source code
Output to Screen
D

Figure 3.4 Un exemple d'une solution bioinformatique réalisée à l'aide de la plate-forme Armadillo. En (A) est présenté le début du flux de travaux et une structure de contrôle (*If*) pouvant être utilisée pour sélectionner des exécutions alternatives du *data-flow*. En (B) est illustré comment différents types d'alignements de séquences peuvent être réalisés. En (C) est suggéré différentes couleurs pouvant être utilisées pour annoter les différentes parties du flux de travaux et ainsi ajouter au processus cognitif. En (D) est présenté l'exemple d'une exécution de code source *Java* si le logiciel ProbCons (Do et *al.*, 2005) est sélectionné lors du *control-flow*.

3.4 Exemple d'utilisation : Inférer des arbres phylogénétiques à l'aide de *Armadillo*

La construction d'un arbre phylogénétique est une étape importante de nombreux projets bioinformatiques tels que la détection des transferts horizontaux de gènes (THG) (Boc *et al.*, 2010; Than *et al.*, 2008; Addario-Berry *et al.*, 2003). Dans cette étude de cas, nous mettrons en évidence les différentes étapes qui sont nécessaires pour construire un arbre phylogénétique de la protéine adiponectine à l'aide de la plate-forme *Armadillo*. L'adiponectine, également désignée comme Acrp30, apM1, GBP28 ou ADIPOQ, en raison de sa découverte par quatre groupes de recherche différents, est une protéine de 244 acides aminés principalement sécrétée par le tissu adipeux blanc. Cette protéine est connue pour ses effets pléiotropiques et est impliquée dans les troubles liés à l'obésité: diabète de type 2, syndrome métabolique et arthérosclérose. Récemment, il a été suggéré que l'adiponectine peut également être un cardioprotecteur et pourrait avoir des propriétés anti-cancéreuses (Brochu-Gaudreau *et al.*, 2010).

3.4.1 Étape I. Créer un jeu de données de l'adiponectine

Inférer un arbre phylogénétique est un processus en trois étapes comprenant: (1) la création d'un ensemble de données de séquences, (2) l'alignement des données de séquences sélectionnées, et (3) l'inférence d'un arbre phylogénétique à partir de l'alignement de séquences suite à l'application d'un modèle d'évolution approprié pour représenter l'histoire évolutive des organismes considérés (Posada, 2008). La première étape consiste ici à rechercher les séquences d'acides aminés. À partir d'un nouveau projet vide (*File→ New Project*), il faut tout d'abord aller dans le menu *File→Preferences*, puis entrer une adresse

courriel, et cliquer sur « *Update* ». Par la suite, on peut utiliser le menu principal de l'application: *Manager→Sequences* pour ouvrir le *Sequence Manager* (Figure 3.5A). Dans le gestionnaire de séquences, choisir l'option: *File→ Import from Internet*. Ceci va ouvrir une nouvelle boîte de dialogue, nommée *Search Internet*, qui permet d'accéder à des données distantes provenant de trois grandes banques de données de séquences: *HUGO* (Human Gene Nomenclature Committee), *GenBank* (soutenu par NCBI) et *ENSEMBL* (voir le Tableau 3.1). Aux fins de présentation, nous allons choisir dans le menu déroulant l'option de base de données de protéines *NCBI* avec le mot-clé : « *adiponectin* ». Une fois la recherche terminée, on peut filtrer les résultats de cette recherche en retenant pour notre analyse seulement les séquences de l'adiponectine et non celles des récepteurs de cette protéine (*c.-à-d.* AdipoR1 et AdipoR2). Pour cette étude de cas, il faut alors entrer dans le champ *Filter your sequences* la taille de « 240 – 244 » et ordonner les séquences par le champ *Description.* Après cette étape, on peut choisir différentes séquences protéiques de l'adiponectine (Figure 3.5A) telles que des séquences provenant de *Homo sapiens*, *Macaca sp.*, *Sus scrofa*, *Bos taurus*, *Felis catus*, *Gallus gallus*, etc. Après cette opération de filtrage, les séquences sélectionnées peuvent être téléchargées à partir de *GenBank* en utilisant l'option *Import selected sequences*, situées dans le bas de la boîte de dialogue. Le gestionnaire de séquences peut ensuite être minimisé.

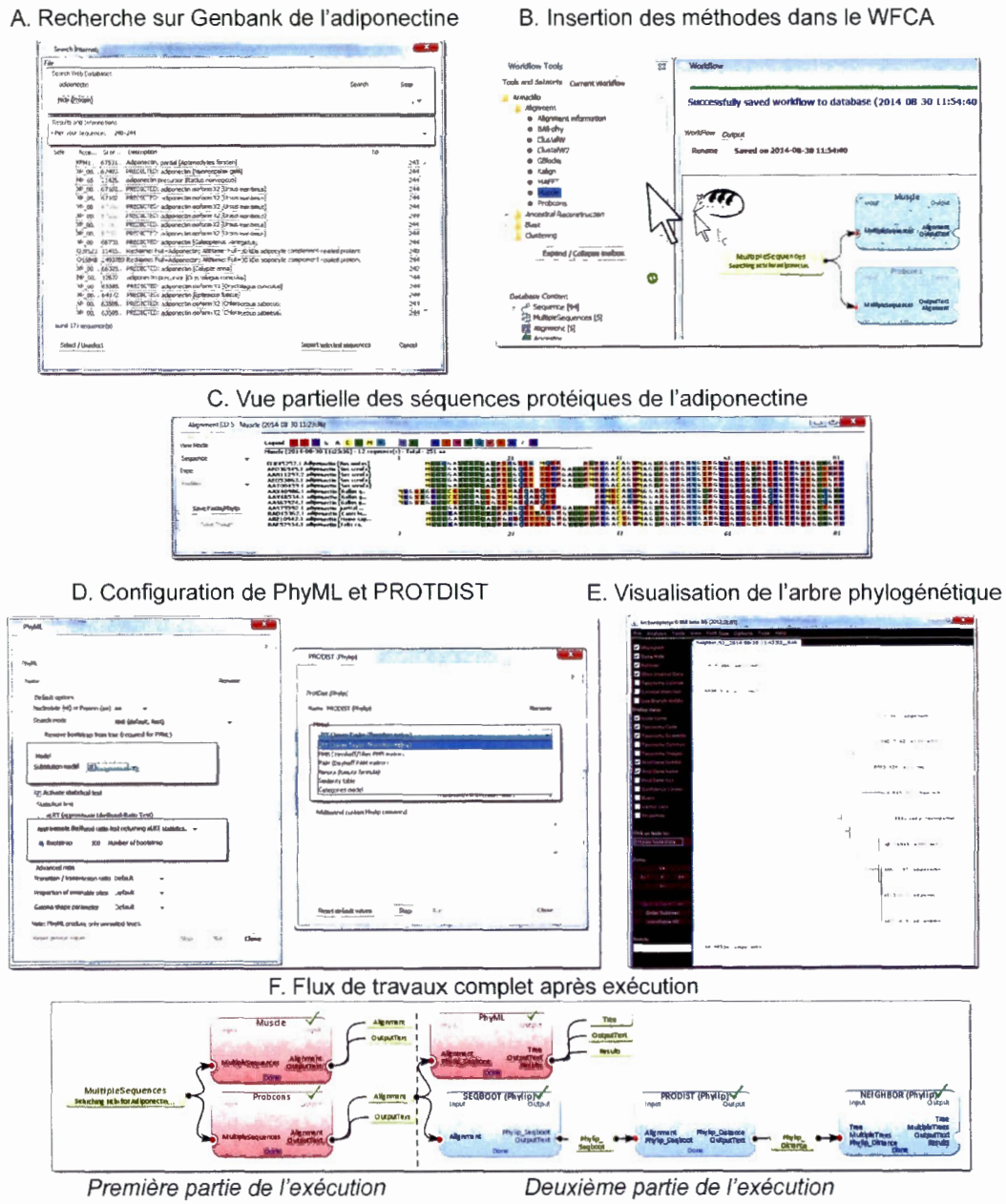


Figure 3.5 Aperçu des différentes étapes nécessaires pour inférer un arbre phylogénétique à l'aide de la plate-forme Armadillo. Étape (A), une boîte de recherche permet un accès rapide à différentes banques de données de séquences biologiques sur Internet. Étape (B), création et interconnexion des composants individuels par des opérations de *drag-and-drop* dans le *WFCA*. Les logiciels d'alignement de séquences Muscle et

ProbCons sont présentés ici. Étape (C), représentation des séquences alignées en utilisant le visualiseur de séquences de la plate-forme. Étape (D), configuration des options des applications PhyML et PROTDIST avant le début de l'inférence phylogénétique. Étape (E), visualisation de l'arbre phylogénétique inféré par PhyML en utilisant le logiciel Archaeopteryx. Étape (F), vue globale du flux de travaux après l'exécution séquentielle de la première partie (*alignement des séquences*) et de la deuxième partie (*l'inférence d'arbres phylogénétiques*).

3.4.2 Étape II. Génération d'alignements de séquences protéiques

La génération d'un alignement valide des séquences choisies est la deuxième étape du protocole d'inférence de l'arbre phylogénétique (Edgar, 2004; Do *et al.*, 2005). Pour aligner les séquences de protéines téléchargées, on peut aller dans la boîte à outils (*Toolbox*) à gauche de la fenêtre d'édition du flux de travaux (Figure 3.5B), sélectionner les *MultipleSequences* nouvellement ajoutées dans le panneau *Tools→ Databases* et faire glisser ces données vers la zone de création du flux de travaux (*WFCA*). Deux logiciels d'alignement de séquences seront utilisés dans cet exemple pour traiter l'ensemble de données de protéines de l'adiponectine. La première est l'application Muscle (Edgar, 2004), qui démontre une plus grande précision que l'algorithme populaire ClustalW (Thompson *et al.,* 1994) lors de l'alignement des séquences de protéines, et la seconde est l'application Probcons (Do *et al.,* 2005), basée sur la modélisation probabiliste de l'alignement des séquences. Pour utiliser ces logiciels, il faut les intégrer dans le flux de travaux en allant dans le panneau *Toolbox → Tools* et en choisissant l'option *Tools → Alignment subtree*. Une fois les programmes Muscle et Probcons trouvés, ils peuvent être déposés sur le *WFCA* (Figure 3.5B) et reliés au composant *MultipleSequences* ajouté précédemment. On peut maintenant exécuter le flux de travaux en utilisant soit le bouton *Execute*, situé dans le coin supérieur droit du *WFCA*, ou l'option *Execution→Run* du menu principal. Une fois que toutes les étapes de l'exécution sont terminées, la barre de progression en haut du *WFCA* avance à la marque de 100 %. On peut alors visualiser les résultats obtenus (Figure 3.5C) soit en sélectionnant la sortie du programme directement dans le *WFCA* et en faisant un *double-clic* sur l'alignement résultant, soit en allant dans le panneau *Toolbox →Current workflow,* puis en étendant le sous-arbre représentant l'exécution des applications Muscle et Probcons. Par la

suite, un clic droit sur l'alignement choisi et la sélection de l'option *Details* ou *View Graphics* dans le menu contextuel permet l'affichage des alignements de séquences.

3.4.3 Étape III. Inférer l'arbre phylogénétique de l'adiponectine

Une fois que l'utilisateur est satisfait de l'un des deux alignements de séquences obtenus, il peut commencer à réaliser l'inférence de l'arbre phylogénétique. Nous présentons ici deux versions alternatives: (1) un premier arbre pourra être obtenu à l'aide d'une méthode par maximum de vraisemblance implémentée dans le logiciel très populaire PhyML (Guindon et Gascuel, 2003), (2) tandis que le second arbre sera généré par la méthode Neighbor-Joining (Saitou et Nei, 1987) disponible dans le *package* PHYLIP (Felsenstein, 1989; Application *Neighbor*). Pour effectuer l'inférence de l'arbre, on peut aller dans *Toolbox* → *Tools,* et sélectionner l'option *Tree* → *PhyML,* pour la faire glisser sur le *WFCA*, puis finalement la connecter à l'alignement nouvellement généré. De même, on peut aller dans la sous-arborescence *Tree* → *Tree(Phylip)*, et faire glisser le logiciel SEQBOOT, PROTDIST et NEIGHBOR sur le *WFCA* et ainsi créer un second pipeline en reliant ces applications (tel qu'indiqué à la Figure 3.5F). Pour personnaliser les paramètres d'exécution des différents logiciels, on peut *double-cliquer* sur la boîte correspondante à l'application dans le *WFCA*. Par exemple, dans le cas du programme PhyML et du jeu de données portant sur l'adiponectine, on peut sélectionner les paramètres suivants : 1) dans la zone des options par défaut: *aa* (protéine); 2) pour l'ensemble de données, le modèle d'évolution JTT; 3) 100 - pour le nombre de répétitions (nombre de *bootstrap*) (voir Figure 3.5D). On pourrait également effectuer au préalable le choix d'un modèle d'évolution approprié en ajoutant le logiciel ProtTest (Abascal *et al.,* 2005) : *Model testing* → *ProtTest* au pipeline d'exécution. Une fois que les exécutions sont terminées, les arbres phylogénétiques obtenus (au format *Newick*) peuvent être visualisés en *double-cliquant* sur le fichier de sortie généré. D'autre part, une représentation graphique des arbres phylogénétiques obtenus peut être générée par un *clic droit* sur l'arbre dans le *WFCA* et en sélectionnant l'option « *View Tree in Archaeopteryx* » ou l'option « *View Tree in PhyloWidget* » dans le menu contextuel (voir Figure 3.5E). Évidemment, le flux de travaux, incluant les résultats intermédiaires, est automatiquement enregistré et peut être facilement modifié pour procéder à des analyses subséquentes.

3.5 Conclusions

Dans ce chapitre, nous avons décrit une nouvelle plate-forme de flux de travaux originale, *Armadillo*, dédiée à la conception et la réalisation d'analyses phylogénétiques et phylogénomiques. Cette nouvelle plate-forme met en œuvre une approche intuitive pour l'automatisation des tâches et la conception des simulations. La version actuelle de *Armadillo* permet la création de flux de travaux répétitifs, tout en assurant la compatibilité de nombreuses applications du domaine de la bioinformatique et permet la comparaison de leurs résultats. Par exemple, la plate-forme *Armadillo* peut être utilisée pour effectuer des analyses de plusieurs gènes (en utilisant par exemple la boucle *For* pour traiter les fichiers de plusieurs gènes). De plus, on peut utiliser *Armadillo* pour concevoir et réaliser des études par simulations comparant les méthodes les plus populaires pour inférer les évènements de transferts horizontaux de gènes *c.-à-d.* l'utilisation des logiciels HGT-détection de Boc *et al.*, (2010), RIATA_HGT (Than *et al.*, 2008) ou LatTrans (Addario-Berry *et al.*, 2003). Ces logiciels ont d'ailleurs été comparés dans un article de 2010 (Boc *et al.*, 2010) en utilisant cette plate-forme. De plus, *Armadillo* a été utilisée avec succès comme un outil d'aide à l'enseignement au cours des années 2010-11 par des professeurs de bioinformatique à l'Université du Québec à Montréal dans le cadre de cours de biologie computationnelle aux études supérieures. Plus récemment, la plate-forme *Armadillo* a été utilisée pour effectuer une étude de la classification des micro-ARNs menée par des bioinformaticiens de l'Université du Québec à Montréal, ainsi qu'une étude de la pression sélective sur le virus de l'immunodéficience humaine (VIH) menée par des chercheurs de l'Hôpital Sainte-Justine de Montréal. Notre logiciel fournit un certain nombre de caractéristiques intéressantes, non-disponibles dans les plates-formes Taverna et Galaxy (voir le Tableau 3.2), comprend des outils pour partager les résultats obtenus, pour interroger des banques de données génomiques et permet l'intégration de nouveaux logiciels. La version actuelle de *Armadillo* a été publiée sous la licence *Open Source GNU General Public License*. Le code source, plusieurs tutoriels et des exemples d'applications sont fournis sur le site Web[20]. Il est à noter que nous avons également mis en place un guide d'utilisation de style *Wiki*. Un tel manuel interactif permet

[20] http://www.bioinfo.uqam.ca/armadillo

aux utilisateurs de suggérer l'ajout de nouvelles options et applications. De plus, la plate-forme *Armadillo* peut être facilement étendue par d'autres développeurs. Cette extension peut concerner soit la plate-forme de flux de travaux en elle-même, l'addition de types de données ou l'ajout d'autres outils bioinformatiques. Dans l'avenir, nous prévoyons inclure dans *Armadillo* différents fichiers multimédias permettant d'augmenter sa capacité d'apprentissage. Nous prévoyons également ajouter à la nouvelle plate-forme une application de gestion des données et fournir aux utilisateurs une installation de stockage de données distantes. Toutes ces améliorations de la plate-forme *Armadillo* aideront les enseignants à créer des outils pédagogiques spécifiques pour les étudiants en bioinformatique. Ils permettront également à *Armadillo* de devenir un outil de choix pour réaliser des simulations auprès des biologistes moléculaires et évolutifs, ainsi que des biostatisticiens.

3.6 Perspectives

Depuis son lancement en 2012, la popularité de la plate-forme va en augmentant (Figure 3.6) avec plus de 1900 téléchargements et 5 citations dans des revues scientifiques. De fait, la plate-forme *Armadillo* est, selon des commentaires reçus d'utilisateurs, a été employée dans le domaine de l'enseignement, principalement en Amérique du Sud. Nous sommes toujours à élargir le nombre de méthodes se retrouvant dans cette plate-forme (Figure 3.7). Bien que la plate-forme puisse être augmentée par les utilisateurs, la philosophie sous-jacente est que toutes les applications incluses devraient, sans requérir à l'installation de nouveaux logiciels, fonctionner sous tous les systèmes d'exploitation.

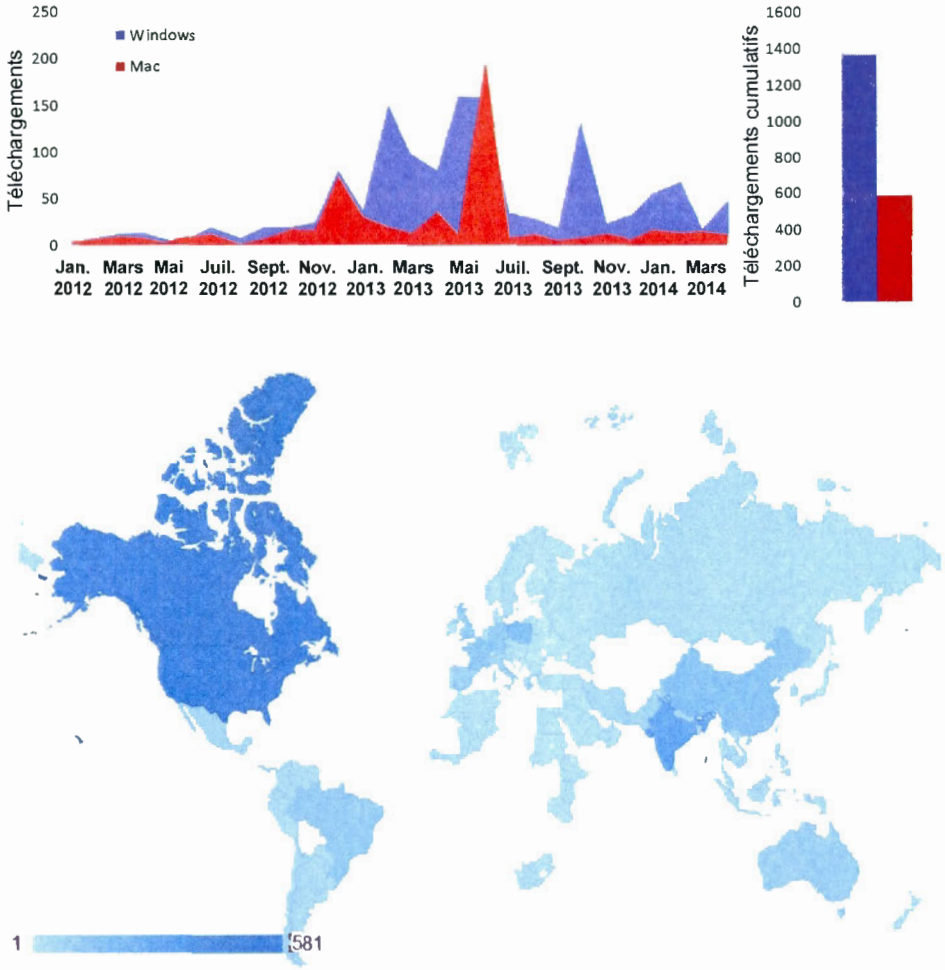


Figure 3.6 Total des téléchargements de la plate-forme *Armadillo* depuis son lancement en 2012 et régions géographiques des téléchargements.

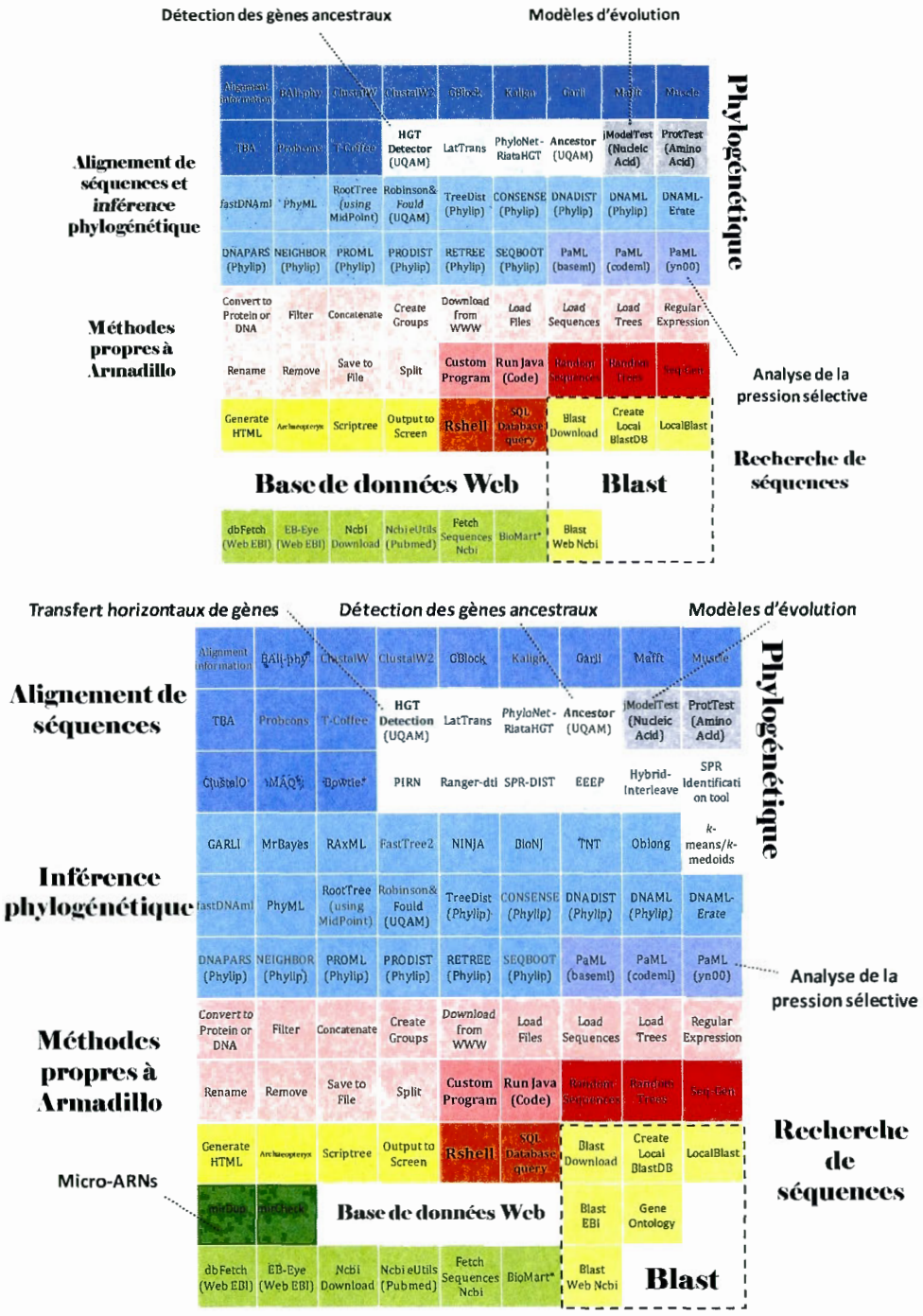


Figure 3.7 Évolution du « *langage applicatif* » de *Armadillo* entre 2012 (*haut*) et 2014 (*bas*).

De plus, dans cette première version, nous avons privilégié l'ajout de nouvelles applications pour avoir un *langage applicatif* le plus complet possible, plutôt que le développement de fonctions permettant le déplacement de tâches vers des ordinateurs distants (Lee *et al.*, 2009; Vairavanathan *et al.*, 2012; Rahman *et al.*, 2013). De fait, des logiciels de phylogénomique tels que RAxML (Stamatakis, 2006) et Oblong (Goloboff, 2014) permettent maintenant de gérer des collections de milliers de gènes (ou de sites) et taxa et d'avoir des performances correctes sur des ordinateurs personnels. De plus, des versions parallélisées de plusieurs logiciels permettant cette distribution de tâches automatique, sont disponibles (voir Annexe C). Ainsi, l'utilisation de la plate-forme de gestion de flux de travaux sert alors à simplifier l'utilisation de ces outils, à gérer les différents formats de fichiers et sources de données, mais encore à conserver une trace de l'analyse de manière à pouvoir la ré-exécuter avec d'autres jeux de données. Une résultante de l'utilisation de la plate-forme actuelle est que plusieurs milliers de flux de travaux peuvent être générés lors d'une seule expérience *in silico*. De plus, plusieurs versions de différents flux de travaux, exécutés partiellement ou en entier, sont sauvegardées dans la plate-forme (Figure 3.8).

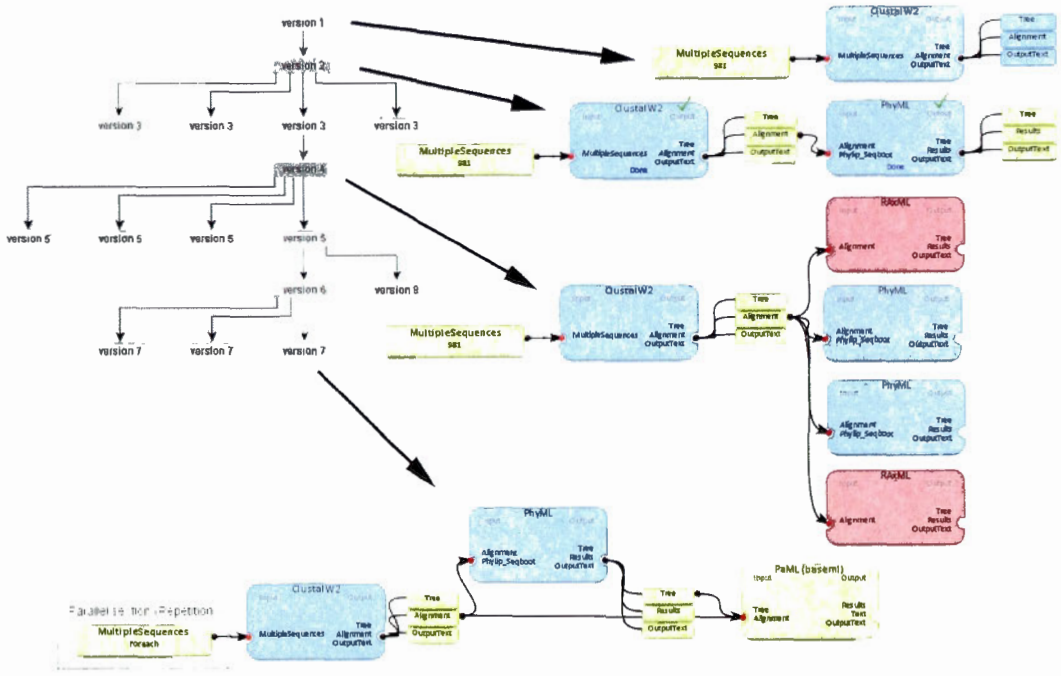


Figure 3.8 Évolution d'une expérience *in silico* avec la plate-forme *Armadillo*.

Cette perspective fait d'ailleurs partie des différentes phases de la vie d'un flux de travaux (Figure 3.9) : 1) *construction et utilisation*, 2) *soumission à une bibliothèque de sauvegarde* (portail; Section 2.3) en vue de sa 3) *réutilisation* (voir Figure 3.8 et Friesen et Rüping, 2010). Dans le prochain chapitre, nous proposerons une solution à cette problématique d'accumulation de flux de travaux, en explorant leur classification et leur regroupement. Pour ce faire, nous utiliserons des techniques de classification issues de la phylogénétique, mais aussi des techniques plus classiques de regroupement telles que *k*-means et *k*-medoids.

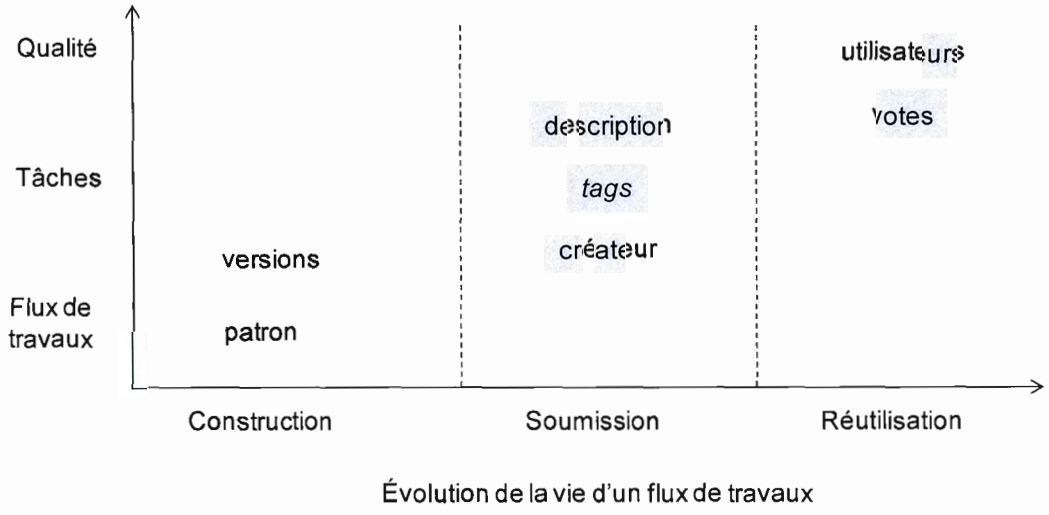
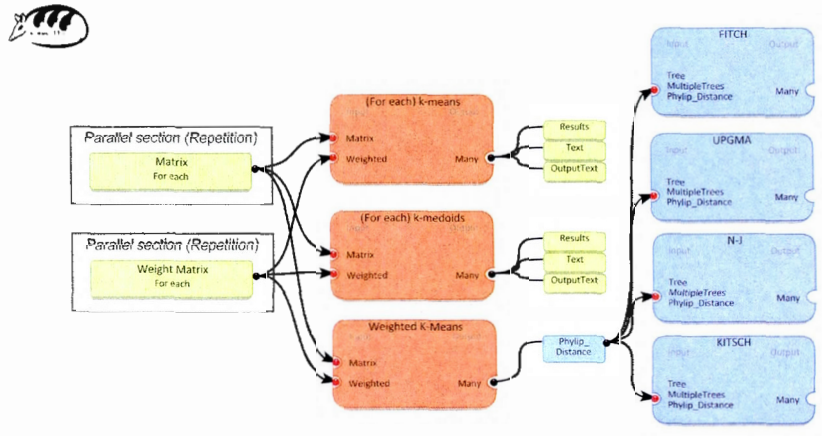


Figure 3.9 Évolution de la vie d'un flux de travaux (Friesen et Rüping, 2010).

CHAPITRE IV

CLASSIFICATION DE FLUX DE TRAVAUX PAR ALGORITHMES DE PARTITIONNEMENT ET REGROUPEMENT HIÉRARCHIQUES



~50 000 Simulations – 96 conditions – 1 flux de travaux – Etienne Lord

4.1 Préface

Dans ce chapitre, nous introduirons de nouveaux types d'encodage des flux de travaux permettant leur regroupement en fonction d'un contexte d'exécution, ou encore permettant leur classification en fonction de mots-clés. Ces stratégies d'encodage peuvent aussi servir à comparer des sections de flux de travaux et permettre ainsi leur réutilisation dans de nouveaux flux de travaux (Hettne *et al.,* 2012). Le texte présenté a été publié dans la revue *BMC Bioinformatics* (Lord *et al.,* 2015a).

4.2 Résumé

Les applications de flux de travaux, comprenant une collection de plusieurs tâches liées, deviennent de plus en plus populaires dans de nombreux domaines scientifiques, y compris la bioinformatique. Par exemple, les simulations, qui sont devenues un *a priori* pour valider statistiquement les résultats de nouvelles méthodes et de nouveaux logiciels, sont souvent réalisées en utilisant des applications des flux de travaux. Les flux de travaux servent alors à organiser la simulation ainsi qu'à minimiser la durée totale des opérations requises. La classification des flux de travaux peut être utilisée pour décider quelles tâches, ou quelles parties des flux de travaux, devraient être fusionnées, exécutées ensemble, exécutées séparément ou parallélisées. Nous proposons dans cette étude quatre nouveaux encodages de flux de travaux: certains d'entre eux permettent de regrouper les flux de travaux ayant des caractéristiques topologiques similaires, alors que d'autres permettent de regrouper les flux de travaux en tenant compte à la fois des caractéristiques topologiques et du temps d'exécution de chacune des tâches. Un nouveau critère de support, permettant d'évaluer le regroupement des flux de travaux individuels, est également proposé. Ce critère de support et de l'encodage des différents flux de travaux a été étudiée en utilisant deux algorithmes de partitionnement pondérés: *k*-means et *k*-medoid, ceci avec trois critères de validité différents: Calinski-Harabasz, Silhouette et LogSS. Deux distances, la distance Euclidienne et la distance cosine, ont été utilisées comme mesures de similarité. Quatre algorithmes hiérarchiques différents: Neighbor-Joining, Unweighted Pair Group Method with Arithmetic Mean (UPGMA), Fitch et Kitsch ont également été considérés.

4.3 Introduction

Les pipelines ou flux de travaux impliquent une série de tâches et de méthodes interconnectées, dont la première est appelée une entrée et la dernière une sortie. Ces enchaînements de tâches peuvent être utilisés pour modéliser une séquence de processus connexes interreliés (Lin *et al.*, 2001). De simples pipelines d'exécution, ils peuvent mener à la création de systèmes complexes qui permettent la planification et l'exécution de flux de données contenant des exécutions conditionnelles et la répartition des tâches sur des grappes d'ordinateurs en réseau (Giardine *et al.*, 2005; Bharathi *et al.*, 2008). La première utilisation

des flux de travaux concerne leur application dans les environnements commerciaux et financiers (Oinn *et al.*, 2007; Woollard *et al.*, 2008). Aujourd'hui, les flux de travaux sont largement utilisés dans les sciences de la vie pour la réalisation d'analyses scientifiques complexes, ainsi que pour la réalisation d'études, par simulations, requises pour la validation de nouveaux logiciels et méthodes statistiques (Costa *et al.*, 2012). Les flux de travaux automatisés, et les plates-formes créées par certains groupes de recherche, sont conçus pour simplifier le processus de découverte de données, le réusinage des données, le traitement des données et la visualisation des résultats (Oinn *et al.*, 2007; Beaulah *et al.*, 2008). Deux plates-formes de flux de travaux scientifiques largement utilisées sont la plate-forme Web Galaxy (Giardine *et al.*, 2005) et la plate-forme de bureau Taverna (Oinn *et al.*, 2007). Ces plates-formes s'appuient sur un langage de programmation interne spécifique et un modèle computationnel d'automatisation permettant l'exécution des flux de travaux. Nous avons récemment développé une plate-forme bioinformatique de flux de travaux locale appelée *Armadillo* (voir chapitre 3 ou Lord *et al.*, 2012). Celle-ci est principalement dédiée au domaine de l'analyse phylogénétique et phylogénomique. La plate-forme *Armadillo* permet aux utilisateurs de déterminer le temps d'exécution exact de chacune des méthodes disponibles. Elle a été utilisée dans cette étude pour générer des données réelles sur des flux de travaux bioinformatiques considérées dans nos simulations (voir la Figure 4.1 pour un exemple de cinq flux de travaux bioinformatiques conçus à l'aide de *Armadillo).*

Ce chapitre est organisé comme suit. Dans la section 4.4, nous passerons brièvement en revue les études existantes sur la classification des flux de travaux. Les méthodes de partitionnement utilisées pour le regroupement de flux de travaux seront discutées dans la section 4.5 qui sera suivi par la description des méthodes de classification hiérarchiques dans la section 4.6. Dans la section 4.7, nous présenterons quatre stratégies d'encodage de flux de travaux qui seront évaluées dans la section 4.8 en utilisant les algorithmes de partitionnement de *k*-means (MacQueen, 1967) et de *k*-medoids (Kaufman et Rousseeuw, 1990) en conjonction avec trois critères de classification bien connus: Calinski-Harabasz (Calinski et Harabasz, 1974), logSS (Hartigan, 1975) et Silhouette (Rousseeuw, 1987) (voir la section 2.11). Dans la section 4.9, les résultats d'une classification hiérarchique seront présentés. Dans la section 4.10, un nouveau critère de validation de la stabilité du

regroupement sera présenté et évalué dans le contexte de la classification des flux de travaux. Elle sera suivie par une section de conclusion (section 4.11).

4.4 Regroupement de flux de travaux - revue de la littérature

L'objectif le plus commun du regroupement des flux de travaux est la découverte de flux de travaux existants pour permettre leur réutilisation ou leur modification en vue de cette réutilisation (Goderis *et al.*, 2008). Un autre objectif consiste à l'optimisation du temps global d'exécution du flux de travaux (Rahman *et al.*, 2013). Par exemple, Vairavanathan et ses collègues (2012) ont récemment mis au point un système de fichiers optimisé pour les flux de travaux dans le domaine de l'infonuagique (*cloud computing*) qui, compte tenu de l'information structurelle des flux de travaux, peut réduire le temps d'exécution de celui-ci par un facteur de sept. De plus, en étiquetant et divisant des flux de travaux en sous-graphes, Singh *et al.* (2008) ont pu minimiser le temps d'exécution total de flux de travaux du domaine de l'astronomie par plus de 97%. Par ailleurs, Tsai *et coll.* (2012) ont aussi développé une nouvelle approche de partitionnement de flux de travaux en groupes de tâches. Cet algorithme de partitionnement a permis de réaliser des gains de temps de l'ordre de 51% lors de l'exécution de plusieurs flux de travaux. Encore, Chen *et al.* (2013) ont adressé le problème du partitionnement des flux de travaux en recherchant la manière de les combiner. Ce faisant, ces derniers auteurs ont mis de l'avant trois méthodes de balancement des tâches dans une queue d'exécution en se basant sur la variance dans la composition des tâches, la variation dans le temps d'exécution et l'impact des différentes tâches sur le temps d'exécution global. Cependant, nous ne traiterons pas de la division de flux de travaux dans cette étude mais plutôt de la partition et de la classification d'un ensemble de flux de travaux.

Un certain nombre d'études récentes ont examiné la question de la classification des flux de travaux (Santos *et al.*, 2008). Les techniques de regroupement des flux de travaux peuvent être divisées en deux classes soit : les approches linguistiques ou encore les approches structurelles (Wombacher et Li, 2010). Dans les approches linguistiques, des mesures de distances de chaîne de caractères, telles que les distances de Hamming ou de Levenshtein, sont appliquées pour évaluer les dissimilarités entre les flux de travaux considérés (Wombacher, 2006). On se base ainsi sur l'extraction de métadonnées et de données

textuelles provenant des flux de travaux et sur la comparaison des mots-clés communs pour mesurer leurs degrés de similarité (Kastner *et al.*, 2009; Costa *et al.*, 2012). Par exemple, en considérant une matrice d'occurrence de termes (en langage naturel) se retrouvant dans des portails de flux de travaux, Costa et ses collaborateurs (2012) ont constaté que dans plus de 90 % des cas, les méthodes de classification de flux de travaux qu'ils proposent sont capables de regrouper de manière cohérente un ensemble de 53 flux de travaux hétérogènes, provenant du même champ d'études. Ces derniers auteurs ont cependant déterminé que les métadonnées étaient rares, inégalement réparties, dans différents formats et souvent mal identifiées dans les différents annuaires de flux de travaux évalués. Le regroupement structurel des flux de travaux repose sur l'analyse des différences dans la structure de graphe des flux de travaux et sur la représentation adoptée (*p.ex.* réseaux de Petri, graphes orientés, etc.). Des mesures basées sur la distance d'édition des graphes (*graph edit distance*), l'isomorphisme des graphes et le nombre minimal ou maximal de sous-graphes communs (*maximum common induce subgraph; MCIS*), ont été utilisées activement dans ce contexte (Jung et Bae, 2006; Wombacher et Li, 2010). Néanmoins, ces méthodes de classification de flux de travaux basées sur la structure ont généralement des complexités algorithmiques très élevées (Conte *et al.*, 2007). Les flux de travaux peuvent également être convertis en représentations sous la forme de vecteurs binaires où chaque tâche du flux de travaux disponible (*c.-à-d.* les applications, élément ou activité) est codée comme présente (1) ou absente (0). Si une représentation vectorielle est considérée, des mesures de similarité comme la distance cosine, Euclidienne ou la distance Euclidienne au carré peuvent être utilisées pour estimer la similarité ou la dissimilarité entre les flux de travaux (Schaeffer, 2007; Santos *et al.*, 2008). Cependant, en utilisant seulement les données sous forme de présence-absence dans la représentation du flux de travaux, on supprime toutes les informations structurelles, ainsi que la caractérisation des données et des méthodes. Afin de contourner ce biais représentatif, on peut appliquer une stratégie de codage vectoriel multiple en utilisant un vecteur de transition, ou encore un vecteur codant pour des processus (Jung et Bae, 2006). Une stratégie portant sur l'utilisation de vecteurs encodant les transitions entre différentes méthodes a été étudiée par Kastner *et al.* (2009) en utilisant plusieurs algorithmes de classification. De plus, Wombacher et Li (2010) ont adopté une représentation *N*-gram de flux de travaux dans laquelle les tâches adjacentes, reliées entre elles, ont été utilisées pour définir un alphabet spécifique. Cet

alphabet a alors été considéré comme une base pour soit encoder une représentation vectorielle des flux de travaux, soit pour calculer directement une distance d'édition des graphes entre les flux de travaux en utilisant comme critère la mesure de *MCIS*.

D'autres informations utiles pour le regroupement peuvent être extraites à partir des flux de travaux en plus du nombre et du type des méthodes : les types d'entrées et de sorties, liens entre les méthodes, des statistiques telles que le temps moyen d'exécution des tâches, la taille des données transmises, les succès ou les échecs de l'exécution des méthodes, ainsi que les paramètres des tâches sélectionnées peuvent également être pris en compte dans le processus de regroupement (Goderis *et al.*, 2008; Kastner *et al.*, 2009; Silva *et al.*, 2011).

Par exemple, Silva et ses collaborateurs (2011) ont récemment mis au point le programme *SimiliFlow* qui accepte en entrée différents formats de flux de travaux et prend en compte la structure des flux de travaux, le type des activités, les ports d'entrée et de sortie, et les relations entre les activités fournies (*p.ex.* distance entre deux activités dans le graphe) lors de la classification des flux de travaux.

4.5 Méthodes de partitionnement pour la classification des flux de travaux

Des méthodes de partitionnement pour le regroupement des flux de travaux ont été considérées par Santos *et al.* (2008) et Kastner *et al.* (2009). Pour tenir compte des informations structurelles contenues dans ces derniers, Santos et ses collègues ont utilisé comme mesure de similarité la distance de *MCIS*, ainsi que la distance cosine entre des représentations vectorielles binaires des flux de travaux. Ils ont ensuite procédé à l'application de la méthode de partitionnement *k*-means sur cette représentation vectorielle, ou encore par l'application de la méthode *k*-medoids sur les représentations fondées sur des graphes. Kastner *et al.* (2009) ont, pour leur part, encodé la transition entre deux tâches disjointes et ont utilisé l'algorithme de *k*-means avec la distance cosine pour obtenir le meilleur regroupement d'une série de flux de travaux simulés.

L'algorithme de *k*-means (MacQueen, 1967; Bock, 2007) est un algorithme de regroupement par partitionnement qui regroupe de manière itérative en K groupes un ensemble de n éléments (*c.-à-d.* des objets, taxa, ou flux de travaux dans cette étude) caractérisés

par *m* variables (*c.-à-d.* des tâches ou des méthodes dans cette étude), tandis que les centres des différents groupes sont choisis pour minimiser la distance intragroupe. Les distances les plus couramment utilisées dans le cadre du partitionnement par *k*-means sont la distance Euclidienne, la distance de Manhattan et la distance de Minkowski. Chaque groupe est centré autour d'un point, appelé le *centroïde*, qui représentent la moyenne des coordonnées des différents éléments composant le groupe. Un des inconvénients de l'algorithme *k*-means est que ce centroïde n'a pas de signification réelle et doit être recalculé après chacune des itérations. Comme le problème général de la répartition des éléments par la méthode *k*-means est NP-difficile, plusieurs heuristiques en temps polynomial ont été proposées. Elles exigent en moyenne $O(K \times n \times m \times i)$ opérations pour trouver une solution au regroupement, où *i* est le nombre d'itérations de l'algorithme, *n* est le nombre de flux de travaux considérés, *m* est le nombre de variables (tâches ou méthodes) et *K* est le nombre de groupes.

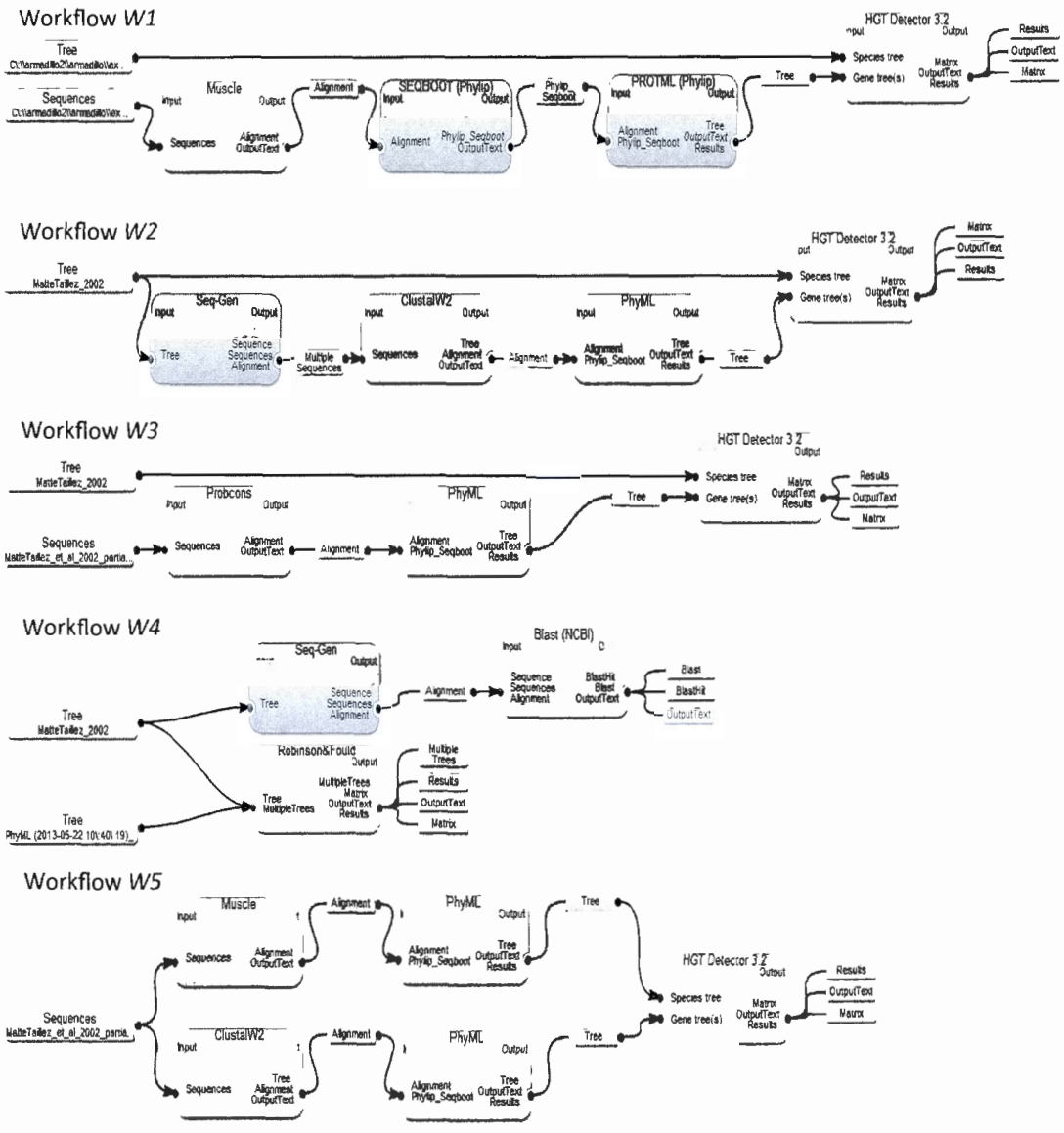


Figure 4.1 Cinq flux de travaux bioinformatiques créés à l'aide de la plate-forme *Armadillo*. Ces flux de travaux ont été utilisés pour illustrer les stratégies d'encodage des flux de travaux discutées dans ce chapitre. Les quatre types d'encodage des flux de travaux examinés et définis dans cette thèse sont présentés dans le Tableau 4.1.

La version la plus populaire de l'algorithme *k*-means peut ainsi être résumée comme suit:

1. Répartir au hasard chacun des éléments du jeu de données à l'un des K groupes;
2. Calculer la valeur du centroïde de chaque groupe;
3. Étant donné les nouvelles coordonnées des centroïdes, réattribuer chaque élément au centroïde le plus rapproché;
4. Répéter les étapes 2 et 3 jusqu'à ce que les positions des centroïdes soient fixes.

L'algorithme *k*-medoids (Kaufman et Rousseeuw, 1990) est une modification de l'algorithme *k*-means, dans lequel les centroïdes, renommés médoïdes, sont des éléments représentatifs du groupe (*centre*). Les médoïdes sont choisis après chaque itération de manière à minimiser la distance l'intragroupe. L'avantage principal de cet algorithme est qu'il est plus robuste que la méthode *k*-means en présence de bruit ou de valeurs aberrantes (*outliers*) (Reynolds *et al.*, 2004). L'algorithme *k*-medoids a cependant une plus grande complexité algorithmique de $O(K \times (n-K)^2 \times m \times i)$.

L'algorithme de *k*-medoids peut être résumé comme suit:

1. Choisir K éléments aléatoirement comme des éléments représentatifs initiaux (médoïdes);
2. Attribuer chacun des éléments restants au médoïde le plus proche;
3. Pour chaque élément représentant i (médoïde), sélectionner au hasard un élément non-représentant j;
4. Calculer le coût total S de la permutation du médoïde i avec élément non-représentatif j;
5. Si S <0, permuter l'élément i avec j;
6. Répéter les étapes 2 à 5 jusqu'à ce que les positions des médoïdes soient fixes.

Reynolds *et al.* (2004) ont proposé des modifications à la méthode originale de *k*-medoids afin de réduire sa complexité algorithmique. La procédure a ainsi été optimisée en

sauvegardant en mémoire une liste ordonnée de proximités pour chacun des médoïdes et en testant l'élément le plus proche en tant que nouveau médoïde potentiel à chaque itération.

En 2001, Makarenkov et Legendre ont décrit une version pondérée de l'algorithme de partitionnement *k*-means. Les optimisations suivantes ont été considérées lors de l'ajout des poids à l'algorithme. Soit **W**, une matrice bidimensionnelle contenant des mesures pour n éléments (*c.-à-d.* des flux de travaux; qui sont représentés par les colonnes de la matrice) et m variables (*c.-à-d.* des tâches individuelles, ou des paires de tâches contenues dans le flux de travaux; représentées par les lignes de la matrice). Soit $\mathbf{y} = \{y_1, ..., y_m\}$, le vecteur des coefficients de pondération attribués aux variables. Dans le contexte du regroupement des flux de travaux, ces poids peuvent refléter par exemple le temps d'exécution des méthodes. En utilisant les équations de Makarenkov et Legendre (2001), les Équations 23 et 24[21] représentent la distance Euclidienne d entre des flux de travaux (Équation 23) et le problème d'optimisation associé (Équation 24):

$$d_{ij} = \sqrt{\sum_{p=1}^{m} y_p (w_{ip} - w_{jp})^2} \,. \tag{23}$$

$$\sum_{k=1}^{K} \left[\sum_{i,j=1}^{n_k} d_{ij}^2 \right] / n_k \rightarrow \min \,. \tag{24}$$

Dans ces équations, K désigne le nombre total de groupes et n_k le nombre d'éléments dans le groupe k. Nous considérons également la distance cosine dans l'application des algorithmes de regroupement. Cette dernière peut être représentée sous la forme pondérée suivante (Équation 25):

---

[21] *N.B.* certaines des équations présentées dans ce chapitre sont aussis décrites aux chapitres 1 et chapitres 2, elles sont répétées ici pour une meilleure compréhension de ce chapitre.

$$d_{ij} = 1 - \cos\theta = 1 - \frac{\sum_{p=1}^{m} y_p (w_{ip} \times w_{jp})}{\sqrt{\sum_{p=1}^{m} y_p w_{ip}^2} \times \sqrt{\sum_{p=1}^{m} y_p w_{jp}^2}}. \quad (25)$$

Dans leur travail pionnier, Santos *et al.* (2008) ont été les premiers à utiliser la distance cosine (non pondérée) dans le cadre du regroupement de flux de travaux. Un des inconvénients de la méthode de partitionnement par *k*-means ou *k*-medoids est la nécessité de choisir le nombre de groupes préalable à ce regroupement. Ce dernier point n'a pas été abordé par Santos et collègues. D'ailleurs, cette question a rarement été considérée dans le cadre de la classification des flux de travaux. Dans notre étude, nous procéderons à l'évaluation du nombre optimal de groupes à l'aide des trois critères suivants : Calinski-Harabasz (Calinski et Harabasz, 1974), logSS (Hartigan, 1975) et Silhouette (Rousseeuw, 1987). Nous déterminerons ainsi lequel est le mieux adapté à la classification des flux de travaux. Le critère de Calinski-Harabasz ainsi que l'indice logSS ont été considérés en fonction de leurs performances supérieures de classification, évaluées par Milligan et Cooper (1985), tandis que l'indice Silhouette a été sélectionné suite à l'évaluation de Arbelaitz *et al.* (2013).

Le critère de Calinski-Harabasz (CH) est un indice de type rapport tenant compte à la fois des variances intergroupes et intragroupe (Équation 26). Ici, le coefficient SS_B représente la variance intergroupes globale, et SS_W est la variance intragroupe globale. On définit aussi K comme le nombre total de groupes et n comme le nombre de flux de travaux considérés:

$$CH(K) = \frac{SS_B}{SS_W} \times \frac{(n-K)}{(K-1)}. \quad (26)$$

Le coefficient SS_B (Équation 27) est évalué par le calcul de la norme L^2 (distance Euclidienne) entre le $mean_k$ (k =1 ... K; $mean_k$ représente le centroïde ou le médoïde du groupe k) et le vecteur *mean*, représentant la moyenne globale de tous les échantillons. Ici, n_k représente le nombre d'éléments dans le groupe k. Le coefficient SS_W (Équation 28) peut alors être calculé de la même manière. De même, w_{ik} est le vecteur représentant le flux de

travaux *i* dans le groupe *k*. Lorsque le critère Calinski-Harabasz est considéré, le nombre de groupes correspondant à sa valeur la plus élevée est sélectionné comme étant le nombre de groupes optimal.

$$SS_B = \sum_{k=1}^{K} n_k \left\| mean_k - mean \right\|^2, \text{ et} \tag{27}$$

$$SS_W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| w_{ik} - mean_k \right\|^2. \tag{28}$$

L'indice logSS (Équation 29) repose également sur l'évaluation du ratio entre la distance intergroupe et la distance intragroupe pour suggérer le nombre de groupes optimal. Lorsque le critère logSS est considéré, le nombre optimal de classes K correspond à la plus petite différence entre les deux scores de logSS subséquents (logSS(K) et logSS(K+1)).

$$\log SS(K) = \log \frac{SS_B}{SS_W}. \tag{29}$$

D'un autre côté, l'indice Silhouette représente une estimation de l'appartenance d'un élément au groupe en cours, et non à celui le plus proche. Pour chaque flux de travaux *i* dans l'ensemble donné des flux de travaux $W = \{w_1, ..., w_n\}$, la variable $a(i)$ représente la distance moyenne entre *i* et tout autre élément du groupe (*c.-à-d.* les autres flux de travaux) dans le groupe c_i auquel *i* appartient. Pour tout groupe *c*, en dehors de c_i, la distance $d(i, c)$ est définie comme étant la distance moyenne entre *i* et tous les autres flux de travaux dans *c*. Ensuite, $b(i)$ représente la plus petite des distances entre tous les groupes différents de c_i. Le groupe *c*, pour laquelle $d(i,c) = b(i)$ peut alors être considéré voisin de *i*. Par conséquent, la moyenne de la largeur Silhouette pour un groupe donné c_k peut être calculée de la manière suivante (Équation 30):

$$s(k) = \left[\sum_{i=1}^{n_k} \frac{b(i) - a(i)}{\max(a(i), b(i))} \right] / n_k. \tag{30}$$

En utilisant les $s(k)$ de l'Équation 30, le nombre optimal de groupes K est identifié comme celui ayant la valeur maximale de la somme des moyennes des largeurs de l'indice Silhouette (Équation 31) :

$$\overline{s}(K) = \sum_{k=1}^{K} [s(k)] / K . \tag{31}$$

4.6 Méthodes de classification hiérarchique de flux de travaux

Dans cette étude, quatre méthodes de classification hiérarchiques différentes ont été considérées dans cette thèse : la méthode *Unweighted Pair Group Method with Arithmetic Mean* (UPGMA) (Sokal et Michener 1958), la méthode de Neighbor-Joining (NJ) de Saitou et Nei (1987), ainsi que les méthodes Fitch et Kitsch implémentées par Felsenstein (Fitch et Margoliash, 1967; Felsenstein, 1989). Ces méthodes hiérarchiques de classification peuvent être appliquées directement sur des matrices de distances calculées à l'aide des quatre schémas d'encodage des flux de travaux que nous présentons dans la section 4.7. Les méthodes UPGMA et Kitsch fournissent une classification ultramétrique (*c.-à-d.* arbre ultramétrique), dans laquelle les longueurs de branches ne peuvent pas être arbitraires; elles sont contraintes de sorte que la longueur totale d'un chemin unique de la racine de l'arbre vers n'importe quelle feuille de cet arbre est toujours la même. Les méthodes NJ et Fitch, par contre, proposent une classification des arbres plus générale correspondant à un arbre additif, ou phylogénétique, c'est-à-dire que dans ce cas les distances entre les différentes feuilles de l'arbre satisfont à la condition des quatre points (Felsenstein, 2004).

Les algorithmes de Fitch et de Kitsch s'appuient sur une fonction objective des moindres carrés visant à minimiser la somme des écarts quadratiques entre les valeurs observées et prédites des éléments (Felsenstein, 2004). L'équation 32 décrit un tel procédé de minimisation, dans laquelle d_{ij} est la distance entre les éléments observés i et j, et δ_{ij} est la distance d'arbre estimée (équivalente à la longueur du chemin reliant i et j dans l'arbre ultramétrique ou additif obtenu). La valeur de l'exposant p est égale à 2 dans le cas des algorithmes de Fitch et Kitsch (Felsenstein, 1984) (Équation 32):

$$\sum_i \sum_j \frac{(\delta_{ij} - d_{ij})^2}{d_{ij}^p} \rightarrow \min . \quad (32)$$

L'algorithme NJ suit le principe de l'évolution minimum visant à minimiser la longueur totale (*c.-à-d.* la somme des longueurs de branches) de l'arbre additif obtenu, tandis que la méthode UPGMA est une approche de regroupement par agglomération ascendante simple et largement utilisée. La complexité temporelle des algorithme de Fitch et Kitsch est de $O(n^4)$, tandis qu'elle est $O(n^3)$ pour NJ et que de $O(n^2)$ pour UPGMA, en considérant en entrée une matrice de dissimilarités de taille (nxn). Nous avons utilisé ces algorithmes de classification hiérarchique pour comparer les quatre stratégies d'encodage des flux de travaux et leurs différentes variantes présentées dans la section suivante.

4.7 Stratégies d'encodage des flux de travaux

Dans cette section, nous discuterons des quatre types généraux d'encodage des flux de travaux. Ces derniers se doivent d'être exprimés sous forme matricielle avant l'application des algorithmes de regroupement considérés. En outre, un vecteur de poids sur les méthodes est prévu pour caractériser les tâches ou méthodes composant le flux de travaux (*p.ex.* les poids reflétaient les temps d'exécution des méthodes dans notre étude). L'application de poids est ainsi souvent considérée pour moduler l'importance de certaines variables ou encore pour réduire la dimension des données (Makarenkov et Legendre, 2001). Par exemple, des poids sont considérés pour représenter les fréquences inverses de certains termes lors du groupement de données textuelles (Schaeffer, 2007). Contrairement à l'approche de Makarenkov et Legendre (2001), dans laquelle on estime que tous les poids ont des valeurs non-négatives et la somme des poids est égale à 1, nous supposons dans cette étude que les poids sont définis par l'utilisateur et ne sont soumis qu'à la contrainte de non-négativité. Un exemple des quatre types d'encodage des flux de travaux qui seront discutés est présenté dans le Tableau 4.1. Il représente l'encodage des cinq flux de travaux bioinformatiques réels présentés à la Figure 4.1.

4.7.1 Encodage de type I

La manière la plus simple d'encoder un flux de travaux est la présentation de données sous la forme d'une matrice binaire reflétant la *présence ou l'absence* de chacune des méthodes disponibles. Dans l'exemple, Figure 4.1, la présence et l'absence de 10 logiciels et méthodes phylogénétiques, utilisées dans le flux de travaux, étaient d'abord encodées (Tableau 4.1). Un tel encodage a été proposé par plusieurs chercheurs, y compris Kastner *et coll.* (2009) et Costa *et al.* (2012). Pour complémenter le travail de Costa *et al.* (2012), nous considérons ici l'addition d'un vecteur de poids représentant le temps d'exécution moyens des méthodes. Le temps d'exécution moyen des 10 méthodes phylogénétiques considérées est indiqué à droite dans le Tableau 4.1. Ce type d'encodage général peut être utilisé soit pour regrouper certains flux de travaux similaires en un tout, qui pourrait ensuite être exécuté ensemble sur un serveur, ou encore pour envoyer vers des services Web différents quelques flux de travaux présentant des temps d'exécution plus important afin de minimiser le temps d'exécution total de l'ensemble des flux de travaux (Vairavanathan *et al.*, 2012).

4.7.2 Encodage de type II

L'encodage des flux de travaux de type II est basé sur les informations *d'occurrence des méthodes.* Ici, nous considérons en plus le poids de chacune des méthodes. Ces poids peuvent être définis par l'utilisateur et appliqués aux méthodes. Dans l'exemple présenté dans le Tableau 4.1 (voir encodage de Type II), le procédé appelé HGT Detector 3.2 a reçu le poids de 1.0, tandis que les neuf autres méthodes ont reçu le poids de 0.1. Les pondérations appliquées peuvent être définis par l'utilisateur à travers l'introduction de mots-clés spécifiques caractérisant certaines méthodes. Le poids de la méthode correspondante peut être donné suite à la présence ou l'absence de la série de mots-clés dans l'annotation des différentes méthodes. Ce type d'encodage pourrait être particulièrement utile pour la recherche et la sélection des flux de travaux appropriés dans une grande banque de données de flux de travaux caractérisés par leurs métadonnées.

4.7.3 Encodage de type III

Afin de déterminer si l'information structurelle des flux de travaux peut fournir une meilleure classification par rapport à la seule présence ou absence des méthodes, nous avons représenté les cinq flux de travaux de la Figure 1 comme des graphes orientés connexes et les avons encodés au format *paire-de-tâches* (*pair-of-tasks*) (voir encodage de type III dans le Tableau 4.1). Ce type d'encodage des flux de travaux, qui est similaire à l'encodage *N*-gram de Wombacher et Li (2010), conserve l'information structurelle essentielle, sans requérir à une méthode dérivée de la théorie des graphes, algorithmiquement beaucoup plus complexe. Un vecteur de temps, caractérisant le temps d'exécution moyen de chaque *paire-de-tâches*, est utilisé comme poids dans l'utilisation de ce type d'encodage. Vairavanathan *et al.* (2012) ont récemment décrit un système de gestion de fichiers adapté aux flux de travaux, qui donnant l'information structurelle des flux de travaux, permet une exécution plus rapide lors du déploiement et l'exécutions de ceux-ci sur des serveurs distants (*cloud computing*). Le regroupement de flux de travaux par leur structure a également été abordé par Kastner *et al.* (2009). Cependant, ces derniers auteurs n'ont pas considéré l'ajout de poids caractérisant les différentes méthodes.

4.7.4 Encodage de type IV

Enfin, nous avons également envisagé l'addition de métadonnées et d'informations sur les ports d'entrée et de sortie des méthodes à la matrice de *paire-de-tâches* et au vecteur de poids. Ce type d'encodage, qui prend en compte le début et la fin du flux de travaux, souligne l'importance des entrées et des sorties dans le pipeline d'analyse. Un tel encodage peut être particulièrement utile dans des situations dans lesquelles l'utilisateur peut revoir et réutiliser les résultats des processus complexes qui ont déjà été exécutés avec des données d'entrées et de sorties identiques à celles spécifiées par l'utilisateur. Ce type de flux de travaux comprend des flux de travaux souvent longs et complexes utilisés en bioinformatique et destinés à l'extraction, l'analyse ou le traitement de gros volumes de données génomiques (Giardine *et al.*, 2005, Oinn *et al.*, 2007). Dans cet encodage, le vecteur de poids est défini comme suit. Premièrement, un poids de 1.0 est attribué aux variables représentant les données d'entrées et de sorties du flux de travaux, ainsi qu'à des variables associées à des méthodes de calcul

choisies par l'utilisateur (par exemple, les méthodes correspondant à des mots-clés spécifiques). Deuxièmement, le poids de 0.1 est attribué aux variables correspondant aux autres méthodes et tâches composant le flux de travaux.

Tableau 4.1 Les quatre propositions d'encodage des flux de travaux et leurs vecteurs de poids associés pour les cinq flux de travaux bioinformatiques présentés à la Figure 4.1. Dans le cas de la double occurrence de la méthode PhyML, utilisée dans le flux de travaux W5, les méthodes sont identifiées comme PhyML(1) et PhyML(2) dans l'encodage de type I.

Encoding of Type I	*W1*	*W2*	*W3*	*W4*	*W5*	Weights for Encoding of Type I
Blast (NCBI)	0	0	0	1	0	0.35
ClustalW2	0	1	0	0	1	0.49
HGT Detector 3.2	1	1	1	0	1	0.88
Muscle	1	0	0	0	1	0.41
PROTML (Phylip)	1	0	0	0	0	0.68
PhyML (1)	0	1	1	0	1	1.13
PhyML (2)	0	0	0	0	1	1.13
Probcons	0	0	1	0	0	0.55
Robinson&Foulds distance	0	0	0	1	0	0.25
SEQBOOT	1	0	0	0	0	0.14
Seq-Gen	0	1	0	1	0	0.43
Encoding of Type II	***W1***	***W2***	***W3***	***W4***	***W5***	**Weights for Encoding of Type II**
Blast (NCBI)	0	0	0	1	0	0.10
ClustalW2	0	1	0	0	1	0.10
HGT Detector 3.2	1	1	1	0	1	1.00
Muscle	1	0	0	0	1	0.10
PROTML (Phylip)	1	0	0	0	0	0.10
PhyML	0	1	1	0	2	0.10
Probcons	0	0	1	0	0	0.10
Robinson&Foulds distance	0	0	0	1	0	0.10
SEQBOOT	1	0	0	0	0	0.10
Seq-Gen	0	1	0	1	0	0.10
Encoding of Type III	***W1***	***W2***	***W3***	***W4***	***W5***	**Weights for Encoding of Type III**
Blast (NCBI)	0	0	0	1	0	0.35
HGT Detector 3.2	1	1	1	0	1	0.88
Robinson&Foulds distance	0	0	0	1	0	0.25
ClustalW2→PhyML	0	1	0	0	1	1.62
Muscle→PhyML	0	0	0	0	1	1.54
Muscle→SEQBOOT (Phylip)	1	0	0	0	0	0.55
PROTML (Phylip)→HGT Detector 3.2	1	0	0	0	0	1.56
PhyML→HGT Detector 3.2	0	1	1	0	2	2.01
Probcons→PhyML	0	0	1	0	0	1.68
SEQBOOT (Phylip)→PROTML (Phylip)	1	0	0	0	0	0.82
Seq-Gen→Blast (NCBI)	0	0	0	1	0	0.78
Seq-Gen→ClustalW2	0	1	0	0	0	0.92
Encoding of Type IV	***W1***	***W2***	***W3***	***W4***	***W5***	**Weights for Encoding of Type IV**
Blast (NCBI)	0	0	0	1	0	0.10
HGT Detector 3.2	1	1	1	0	1	1.00
Robinson&Foulds distance	0	0	0	1	0	0.10
ClustalW2→PhyML	0	1	0	0	1	0.10
Muscle→PhyML	0	0	0	0	1	0.10
Muscle→SEQBOOT (Phylip)	1	0	0	0	0	0.10
PROTML (Phylip)→HGT Detector 3.2	1	0	0	0	0	1.00
PhyML→HGT Detector 3.2	0	1	1	0	2	1.00
Probcons→PhyML	0	0	1	0	0	0.10
SEQBOOT (Phylip)→PROTML (Phylip)	1	0	0	0	0	0.10
Seq-Gen→Blast (NCBI)	0	0	0	1	0	0.10
Seq-Gen→ClustalW2	0	1	0	0	0	0.10
INPUT_Sequences	1	0	1	0	1	1.00
INPUT_Tree	1	1	1	2	0	1.00
OUTPUT_Blast (NCBI)	0	0	0	1	0	1.00
OUTPUT_Matrix	1	1	1	1	1	1.00
OUTPUT_MultipleTrees	0	0	0	1	0	1.00
OUTPUT_OutputText	1	1	1	2	1	1.00
OUTPUT_Results	1	1	1	1	1	1.00

En fonction du type d'encodage, les cinq flux de travaux illustrés à la Figure 1 ont été regroupés dans les sous-ensembles optimaux suivants en utilisant la version pondérée de l'algorithme de partitionnement *k*-means, et le critère d'optimisation de Calinski-Harabasz. Ici, *K* désigne le nombre de groupes pour encodage de type I: *K*=4 - {W1}, {W2}, {W3, W4} et {W5}, encodage de type II: *K*=3 - {W1,W3,W5}, {W2} et {W4}, encodage de type III: *K*=4 - {W1}, {W2,W4}, {W3} et {W5}, et l'encodage de type IV: *K*=4 - {W1}, {W2, W3}, {W4} et {W5}.

4.8 Classification des encodages à l'aide des méthodes de partitionnement

Pour évaluer les quatre stratégies d'encodage des flux de travaux définis dans les sections précédentes, nous avons considéré une sélection de 120 flux de travaux bioinformatiques créés et exécutés à l'aide de la plate-forme *Armadillo* (Lord *et al.*, 2012), ainsi que 100 flux de travaux extraits du portail *myExperiment*[22] (Goderis *et al.*, 2008) (Tableau 4.2). Le jeu de données de *Armadillo* contenait quatre classes de flux de travaux ($K = 4$) et 17 tâches différentes (*c.-à-d.* des méthodes) lors des encodages de type I et II, 30 tâches différentes pour l'encodage de type III et 47 tâches différentes pour l'encodage de type IV (Voir le Tableau supplémentaire B.1 en annexe). Chaque flux de travaux de l'ensemble de données de *Armadillo* est composé d'un nombre maximum de huit tâches, choisies parmi un groupe de 17 méthodes couramment utilisées en bioinformatique et divisé en quatre catégories: (1) les méthodes d'alignement de séquences: Alignment information, ClustalW2, Baliphy, Muscle, Probcons et Kalign; (2) les méthodes d'inférence d'arbres phylogénétiques: Garly, Neighbor, PhyML, ProtML, Seqboot et ProtPars; (3) les méthodes pour la détection de transferts horizontaux de gènes et les méthodes de comparaison d'arbres: HGT Detector, Riata, BLAST, Robinson and Foulds distance, et Random tree; et, enfin, (4) un échantillon mixte formé des méthodes sélectionnées dans les trois catégories mentionnées ci-haut. Le mot-clé utilisé pour l'encodage de type II et de type IV était « HGT » (signifiant transferts horizontaux de gènes). Ainsi, les méthodes annotées avec le mot-clé « HGT » ont reçu un poids de 1.0, alors que toutes les autres méthodes ont reçu le poids de 0.1. Les 100 flux de travaux formant l'ensemble de données de *myExperiment* ont été récupérés à partir du portail

Web en utilisant les mots-clés « phylogenetics » et « bioinformatics ». Parmi les flux de travaux extraits, nous avons sélectionné ceux générés par la plate-forme *Taverna* (version 1 et 2; Oinn *et al.*, 2007). Puisque l'exécution expérimentale n'était pas possible pour tous les flux de travaux dans cet ensemble de données, le temps de fonctionnement approximatif de chacune des 318 méthodes disponibles a été établi sur la base de notre connaissance de ces méthodes. La classification de ces flux de travaux dans 15 classes ($K = 15$) est fondée sur l'analyse des métadonnées disponibles sur le site de *myExperiment* (voir le Tableau B.2 en annexe). Pour cet ensemble de données, le mot-clé utilisé pour l'encodage de type II et IV était « BLAST ».

Tableau 4.2 Principales caractéristiques des flux de travaux réels provenant des jeux de données *Armadillo* et *myExperiment* considérés dans cette étude.

Jeux de données	Nombre de flux de travaux (N)	Méthodes de Types I et II	Méthodes de Type III	Méthodes de Type IV	Nombre de classes (K)	Mots-clés utilisés pour l'encodage de Types II et IV
Armadillo	120	17	30	47	4	HGT
myExperiment	100	318	345	497	15	BLAST

Dans la première partie de notre étude, nous n'avons considéré que l'ensemble de données *Armadillo,* ainsi que l'algorithme de partitionnement de *k*-means et la distance Euclidienne. Pour chacun des quatre encodages de données discutés dans la section précédente, l'algorithme de *k*-means pondéré a été exécuté avec un nombre de départs aléatoires fixé à 1000 et un nombre maximal de classes égal à 40. L'évaluation de la qualité des stratégies d'encodage a été effectuée par le calcul de l'indice Rand (*RI*, Rand, 1971). L'indice Rand a été évalué en comparant la partition de 120 flux de travaux obtenue avec la partition de référence de quatre classes du jeu de données de *Armadillo* (voir le Tableau B.2 en annexe). L'indice Rand a été calculé séparément pour les flux de travaux ayant des nombres différents de méthodes (ce nombre varie de 1 à 8 dans les flux de travaux de

[22] www.myexperiment.org

l'ensemble de données de *Armadillo*). Les résultats obtenus ont été présentés en fonction du nombre de méthodes incluses dans des flux de travaux (Figure 4.2). Les critères de Calinski-Harabasz (CH), Silhouette (SI) et logSS ont été utilisés pour déterminer le nombre optimal de groupes lors de ces simulations.

Nous avons d'abord évalué les performances de la stratégie d'encodage de base (type I, voir la Figure 4.2a), constituée d'une matrice de présence et d'absence binaire accompagnée des poids proportionnels aux temps de fonctionnement des méthodes. Nous avons constaté que le nombre optimal de classes proposées par l'indice CH était de 3 (le RI moyen de CH sur la Figure 4.2a est de 0.738), par l'indice Silhouette était de 8 (le RI moyen de SI est 0.851), et par l'indice logSS était de 5 (le RI moyen de logSS est 0.808). Ces résultats suggèrent que, en fonction de l'indice Rand, SI était supérieur aux indices CH et logSS pour l'encodage de type I. L'autre tendance que l'on peut observer est que l'augmentation du nombre de méthodes dans les flux de travaux conduit à une augmentation de la qualité du regroupement pour ce jeu de données, indépendamment du critère d'optimisation sélectionné (CH, SI ou logSS).

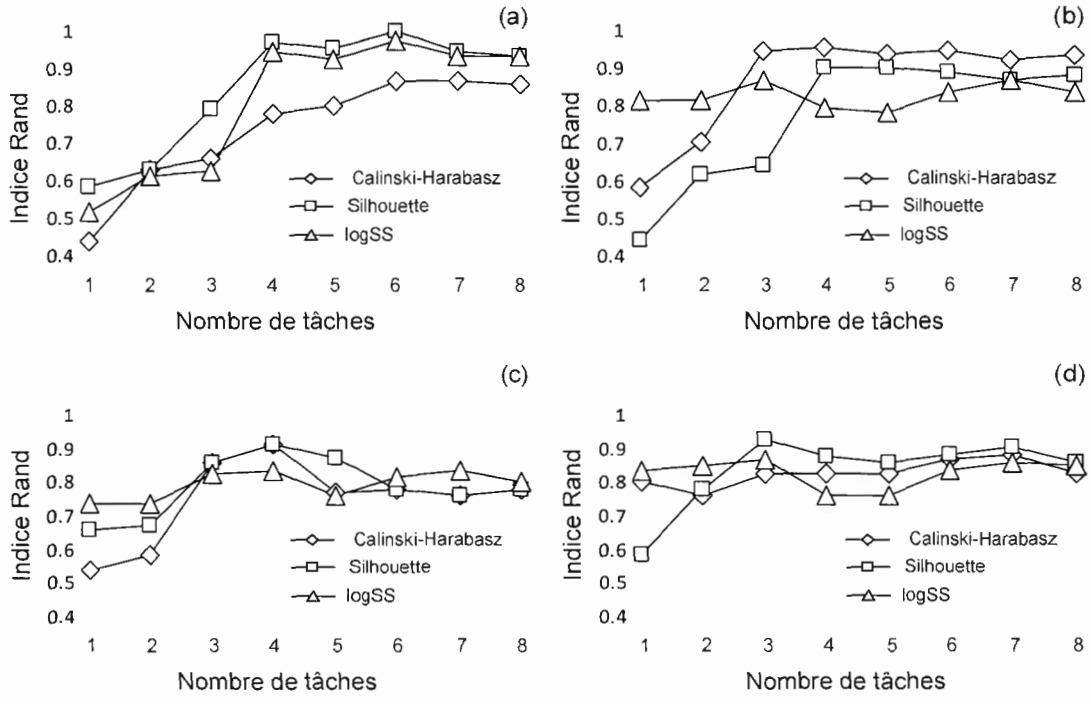


Figure 4.2 Les résultats des simulations obtenus pour les quatre stratégies d'encodage des flux de travaux décrites dans ce chapitre. L'axe des abscisses indique le nombre de tâches (par exemple des méthodes computationnelles) dans le flux de travaux; l'axe des ordonnées indique la valeur de l'indice Rand (RI). Les résultats obtenus en utilisant l'encodage de type I, II, III et IV sont présentés dans les panneaux (a), (b), (c) et (d), respectivement.

Nous avons ensuite évalué la stratégie d'encodage de type II (Figure 4.2b). La matrice d'occurrence des méthodes et le vecteur de poids correspondant au mot-clé choisi « HGT » ont été ici considérés. Nous avons constaté que le nombre optimal de groupes proposés par le critère CH était de 12 (le RI moyen de CH sur la Figure 4.2b est 0.841), par SI était de 4 (le RI moyen des SI est de 0.754), et par logSS était de 39 (le RI moyen de logSS est 0.826). Ces résultats suggèrent que le critère de CH était supérieur à SI et logSS pour l'encodage de type II, en utilisant comme référence l'indice Rand. L'autre tendance que l'on observe est que l'augmentation du nombre de tâches contenues dans les flux de travaux augmente la valeur de RI dans le cas des trois critères d'optimisation considérés. De plus,

l'indice logSS fournit ici un grand nombre de classes (tandis que le nombre optimal de classes est de 4). Ce critère est ainsi non recommandé lors de l'utilisation de l'encodage de type II.

Le troisième type d'encodage (Figure 4.2c) consiste en la représentation de la structure du flux de travaux sous la forme de *paire-de-tâches*. Ce type d'encodage permet ainsi de conserver les éléments structuraux du flux de travaux, à la différence des matrices de tâches de types I et II. Comme dans l'encodage de type I, les poids représentent ici la durée moyenne d'exécution des méthodes bioinformatiques sélectionnées. Les résultats suivants ont été obtenus: le nombre optimal de groupes proposés par CH était de 8 (la moyenne de RI pour le critère CH à la Figure 4.2c est de 0.735), par SI était de 11 (la moyenne de RI pour SI est de 0.761), et par logSS était à nouveau 39 (le RI moyen de logSS est de 0.793). Encore une fois, l'indice logSS était loin de fournir le nombre optimal de classes, en dépit d'une bonne performance en terme des valeurs de RI.

L'encodage de type IV (Figure 4.2d) met l'accent sur les types de données d'entrée et de sortie. Ce type de regroupement a été recommandé par Goderis *et al.* (2008), puis par Wombacher et Li (2010). Contrairement à ces études, nous n'avons considéré dans notre encodage que les entrées et les sorties primaires des flux de travaux, ignorant ceux des tâches intermédiaires. Ce type d'encodage est en accord avec les spécifications utilisées dans la populaire plate-forme de flux de travaux scientifiques *Taverna* (Oinn *et al.*, 2007). Ici, nous avons utilisé un poids de 1.0 pour les données en entrée et en sortie, ainsi que pour les *paires-de-tâches* contenant la méthode HGT Detector, et le poids de 0.1 pour toutes les autres *paires-de-tâches* disponibles. Suite à l'analyse, on retrouve que le nombre optimal de classes pour le critère CH était de 22 (le RI moyen pour CH sur la Figure 4.2d est de 0.826), pour SI était de 13 (RI moyen pour SI est de 0.819), et pour logSS à nouveau de 39 (RI moyen de logSS est de 0.789). Cette fois encore, l'indice logSS était loin de fournir le nombre optimal de groupes pour ce type d'encodage des flux de travaux.

La tendance générale qui peut être observée dans cette simulation, pour les quatre types d'encodage, est que l'augmentation dans le nombre de tâches dans des flux de travaux, mène à une augmentation de la valeur de RI dans le cas des indices Calinski-Harabasz et Silhouette et logSS.

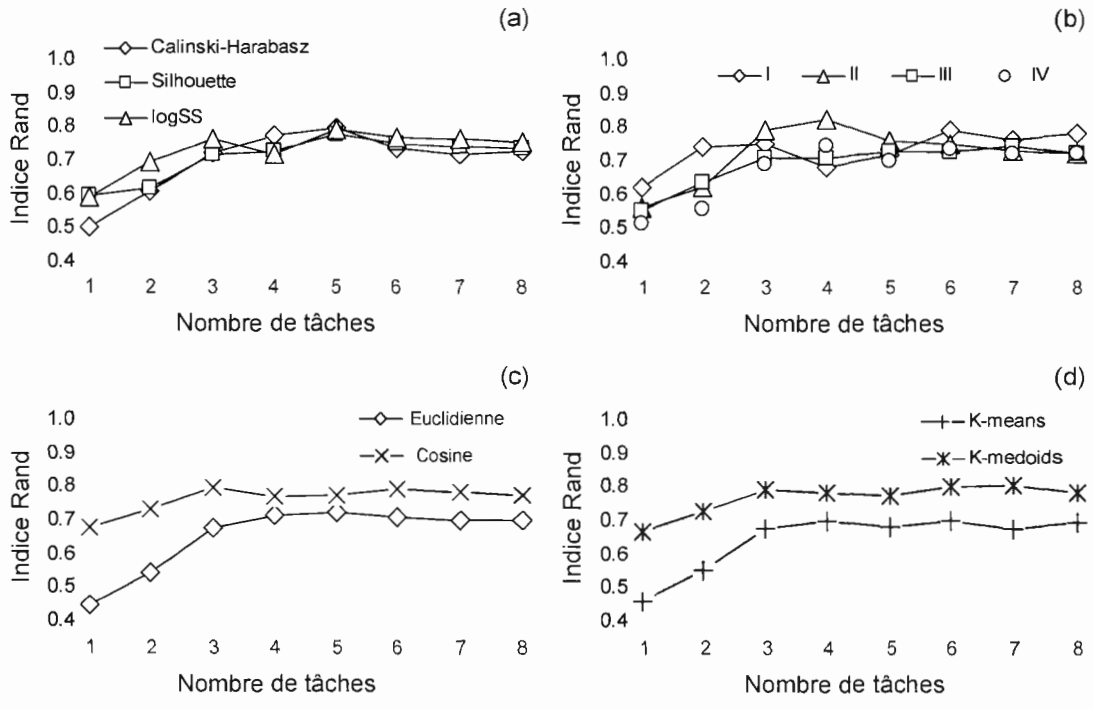


Figure 4.3 Résultats de la simulation étudiant l'évolution de l'indice Rand moyen pour l'ensemble de données de *Armadillo*. L'axe des abscisses indique le nombre de tâches (*c.-à-d.* de méthodes) dans le flux de travaux. Le panneau (a) illustre l'effet du critère d'optimisation; le panneau (b) - l'effet du type d'encodage; le panneau (c) - l'effet de la mesure de distances; et le panneau (d) - l'effet de l'algorithme de partitionnement appliqué.

La deuxième partie de nos simulations a été réalisée en utilisant à la fois les ensembles de données de *Armadillo* et de *myExperiment*, les algorithmes de partitionnement pondérés *k*-means et *k*-medoids et les distances Euclidienne et cosine. Dans ces simulations, les options de 100 départs aléatoires et un nombre maximal de groupes égal à 20 ont été sélectionnées. Chaque point présenté dans les Figures 4.3 et 4.4 représente la moyenne calculée sur l'ensemble des combinaisons de paramètres, mis à part, les paramètres d'intérêt fixés (*p.ex.* à la Figure 4.3a, la moyenne est calculée sur les résultats obtenus en utilisant les méthodes *k*-means et *k*-medoids, les distances cosine et Euclidienne et les quatre types d'encodage discutés). Le test de Student-Newman-Keuls a été mis en place pour identifier les

moyennes des échantillons qui étaient significativement différentes l'une de l'autre et le test de Kolmogorov-Smirnov a servi à vérifier la normalité des données. Tous les tests statistiques ont été effectués en utilisant le programme de nStat v3.0. Toujours en considérant l'indice Rand comme une mesure de l'efficacité du regroupement, nous avons confirmé que pour l'ensemble des flux de travaux de *Armadillo*, un plus grand nombre de tâches dans le flux de travaux mène en général à des meilleurs résultats de classification quel que soit le critère d'optimisation (CH, Silhouette ou logSS) utilisé pour sélectionner le nombre optimal de classes ($p<0.01$; Figure 4.3a). Cependant, dans le cas des simulations effectuées avec l'ensemble de flux de travaux de *myExperiment* (Figure 4.4a), après un certain point (c'est à dire l'intervalle de 40-50 tâches pour ces données), avoir plus de tâches dans chacun des flux de travaux n'aboutit pas à une meilleure classification. Un tel résultat peut être lié à l'accumulation du bruit avec l'augmentation du nombre de variables considérées lors de la classification (Makarenkov et Legendre, 2001). Globalement, l'application d'un critère d'optimisation particulier n'a pas eu un impact significatif sur la performance du regroupement (voir les Figures 4.3a, 4.4a et 4.5a) en terme de l'indice Rand ($p> 0,05$).

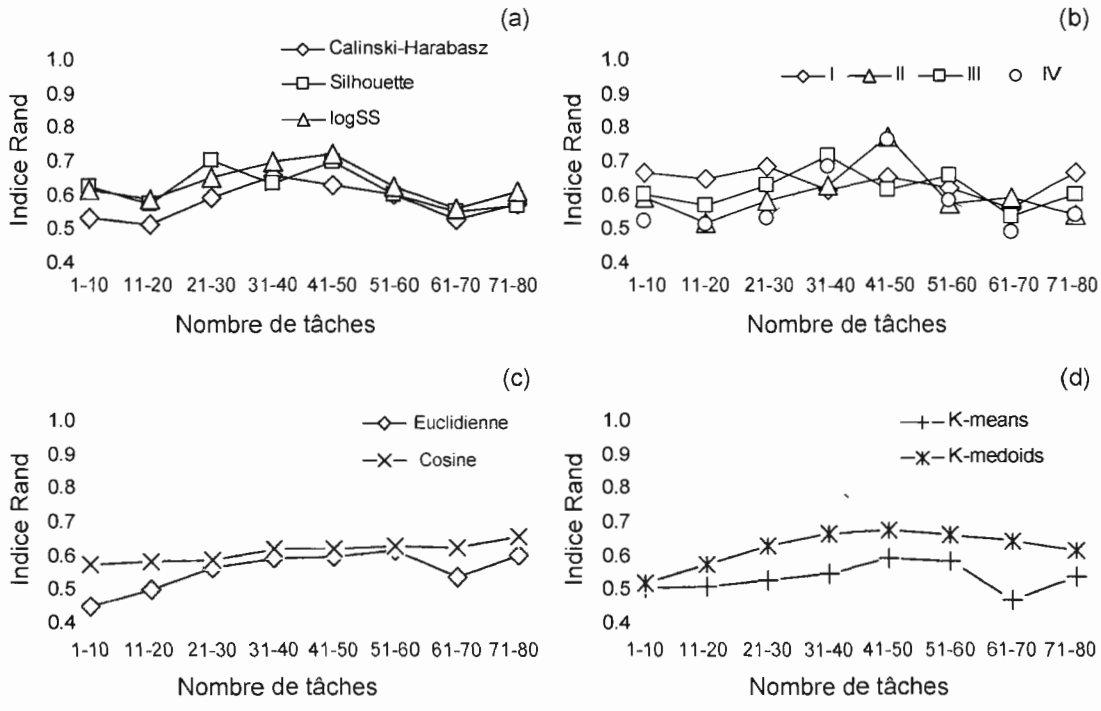


Figure 4.4 Résultats de la simulation représentant l'évolution de l'indice Rand moyen en fonction du nombre de tâches contenues dans les flux de travaux de l'ensemble de données de *myExperiment*. L'axe des abscisses représente un intervalle correspondant au nombre de tâches (*c.-à-d.* les méthodes computationnelles) présentes dans le flux de travaux; 8 intervalles ont été pris en compte dans nos simulations. Le panneau (a) illustre l'effet des critères d'optimisation; le panneau (b) - l'effet du type d'encodage; le panneau (c) - l'effet de la mesure de distances; et le panneau (d) - l'effet de l'algorithme de partitionnement appliqué.

De même, aucune corrélation significative n'a été observée entre le type d'encodage du flux de travaux et le nombre de tâches composant le flux de travaux (Figures 4.3b et 4.4b). Toutefois, lorsque nous avons combiné les résultats obtenus pour les deux ensembles de données (Figure 4.5a) et considéré les encodages non pondérés, nous avons trouvé des différences significatives dans la moyenne de l'indice Rand pour l'encodage de type I ($p<0.01$) et de type II ($p<0.05$), comparativement aux résultats globaux non pondérés pour ces deux types d'encodage (ils sont désignés comme Unw I, II à la Figure 4.5a). En revanche, aucune différence significative ($p>0.05$) n'a été observée entre les résultats correspondant aux

tâches résultant de l'encodage des données sous la forme de matrices de *paires-de-tâches* pondérées et non pondérées (voir les barres désignés par Unw III, IV, III et Type III et Type IV sur la Figure 4.5a). De plus, aucune différence significative n'a été observée en comparant les résultats obtenus avec les trois critères considérés lors du regroupement (CH, SI et logSS). Le critère SI a cependant fourni les meilleurs résultats globaux en fonction de l'indice Rand pour les encodages de types II et III, tandis que logSS a mieux performé que les deux autres critères de classification pour les encodages de types I et IV (Figure 4.5a).

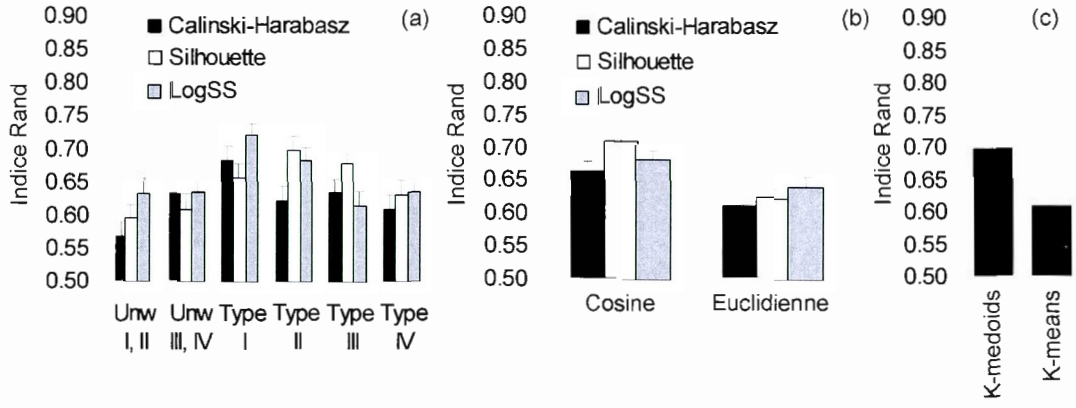


Figure 4.5 Résultats de classification combinés pour l'ensemble de flux de travaux de *Armadillo* et de *myExperiment* obtenus en utilisant les quatre types d'encodage avec et sans poids (indice moyen Rand ± SEM). Les stratégies d'encodage non pondérées ont été respectivement désignées comme Unw I, II (résultats combinés pour les encodages non pondérés des types I et II) et Unw III, IV (résultats combinés pour les codages non pondérés de types III et IV). Le panneau (a) illustre l'effet des critères d'optimisation pour les encodages non pondérés (les deux premiers jeux de barres) et pondérés (quatre derniers jeux de barres); le panneau (b) - l'effet de la mesure de distances; et le panneau (c) - l'effet de l'algorithme de partitionnement appliqué.

L'évaluation de la répartition des flux de travaux en fonction de la mesure de distances a montré que la distance cosine est nettement supérieure à la distance Euclidienne (RI moyen

de 0.68 vs 0.61 et $p<0.001$, voir les Figures 4.3c, 4.4c et 4.5b). Les meilleurs résultats pour la distance cosine ont été obtenus indépendamment du nombre de tâches contenues dans les flux de travaux (Figures 4.3c et 4.4c). Cette conclusion est en accord avec les travaux de Santos *et al.* (2008), qui ont suggéré l'utilisation de la distance cosine (non pondérée) pour le regroupement des flux de travaux. Bien que l'indice Silhouette a fourni de meilleurs résultats de classification (en moyenne) que CH et logSS lorsque la distance cosine a été considérée, la différence obtenue n'était pas significative ($p>0.05$). La comparaison des résultats moyens des classifications en utilisant les algorithmes de partitionnement *k*-medoids et *k*-means permet de souligner d'une manière significative les meilleures performances de l'algorithme *k*-medoids (moyenne de RI 7.0 vs 6.1, $p<0,001$; voir la Figure 4.5c). Finalement, lorsque le *k*-partitionnement a été réalisé, les indices SI et logSS ont surclassé l'indice CH avec un RI moyen respectif de 0.71, 0.72 pour les deux premiers et de 0.65 pour CH ($p<0.01$).

4.9 Classification des encodages à l'aide des méthodes de classification hiérarchique

Dans cette section, nous présentons les résultats obtenus en utilisant les méthodes de classification hiérarchiques dans le cadre du regroupement de flux de travaux. Dans nos simulations, nous avons testé les quatre stratégies d'encodage de flux de travaux définies dans la section 4.5. Leurs formes pondérées et non pondérées ont été considérées. Comme dans nos simulations précédentes, la distance Euclidienne ou la distance cosine ont été utilisées pour calculer la similarité entre les flux de travaux provenant soit du jeu de données de *Armadillo* ou du jeu de données de *myExperiment*. Les algorithmes de reconstruction d'arbres Fitch, Kitsch, Neighbor-Joining (NJ) et UPGMA ont été appliqués pour inférer les classifications hiérarchiques (*c.-à-d.* des arbres additifs) en utilisant les logiciels Fitch, Kitsch et Neighbor du *package* PHYLIP (Felsenstein, 1989). Les regroupements résultants ont été évalués au moyen de la distance topologique de Robinson et Foulds (*RF*; Robinson et Foulds, 1981) à l'aide de logiciels disponibles sur le site Web[23] de T-Rex (Boc *et al.*, 2012).

---

[23] http://trex.uqam.ca

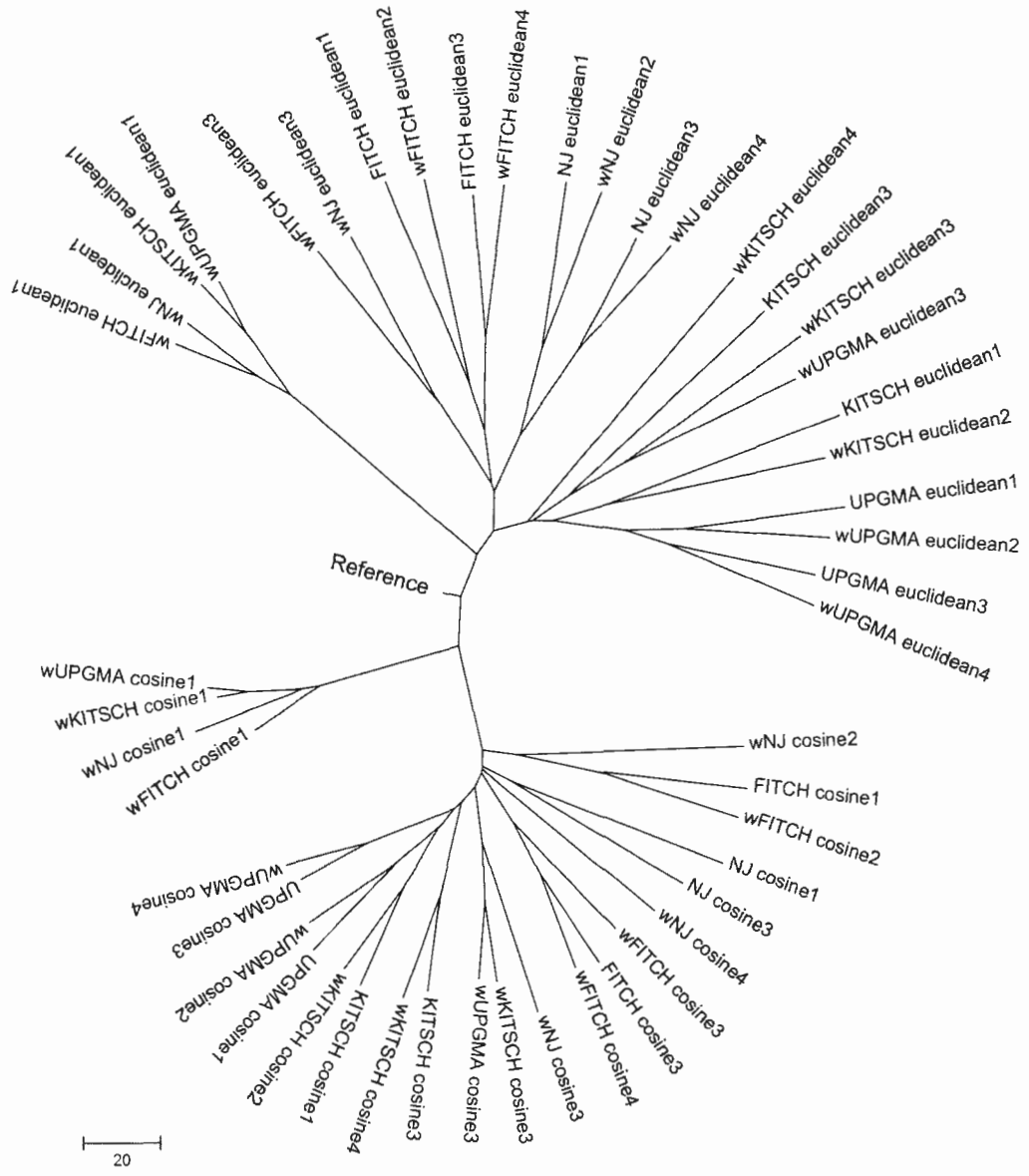


Figure 4.6 Classification hiérarchique des stratégies de regroupement hiérarchiques des flux de travaux obtenue pour les quatre types d'encodage pondérés et non pondérés (I, II, III et IV) décrit dans cette thèse, la distance cosine et la distance Euclidienne et quatre algorithmes de regroupement hiérarchique différents (Fitch, Kirsch, NJ et UPGMA) pour l'ensemble de données de *Armadillo* (n=120). L'utilisation du type d'encodage pondéré est indiquée par la lettre « *w* » précédant le nom de la méthode, alors que le type d'encodage est

indiqué par le chiffre correspondant. Le taxon de référence représente le regroupement optimal.

Les arbres obtenus ont été comparés à un arbre de référence construit à partir des classes définies précédemment (voir les Tableaux supplémentaires D.1 et D.2). Dans ces arbres de référence non-binaires, les flux de travaux appartenant à la même classe ont été reliés par une multifurcation (un nœud de degré supérieur à 3).

Comme il était impossible de présenter chaque arbre additif obtenu pour chaque combinaison de paramètres dans nos simulations, nous avons décidé de comparer ces arbres à l'aide de la distance RF. De plus, nous avons appliqué l'algorithme NJ (NJ a été appliqué à la matrice de distances RF) afin de fournir une classification hiérarchique unique des arbres obtenus pour les deux ensembles de données expérimentales considérées (voir les Figures 4.6 et 4.7). Dans les arbres de classification illustrés, chaque taxon représente un arbre d'additif obtenu en utilisant la combinaison des paramètres de simulations indiqués. La visualisation de la classification résultante des Figures 4.6 et 4.7 a été effectuée à l'aide du logiciel MEGA (Tamura *et al.*, 2011).

En utilisant l'algorithme NJ et la distance RF comme une mesure de proximité arbre, nous avons constaté que pour le jeu de données de *Armadillo*, la distance cosine pondérée et l'encodage de type I fournissent la meilleure classification hiérarchique par rapport à la classification de flux de travaux de référence. Le groupe de quatre arbres obtenus en utilisant la distance cosine pondérée et l'encodage de type I est la plus proche de l'arbre de référence en terme de la distance d'additive (*c.-à-d.* la somme des longueurs des branches constituant l'unique chemin reliant les taxa dans l'arbre, voir la Figure 4.6). Pour l'ensemble de données de *myExperiment*, nous avons constaté que la distance cosine pondérée et la distance Euclidienne pondérée, de concert avec les encodages de types I et III, ont fourni les meilleurs résultats lors de la classification hiérarchique (Figure 4.7).

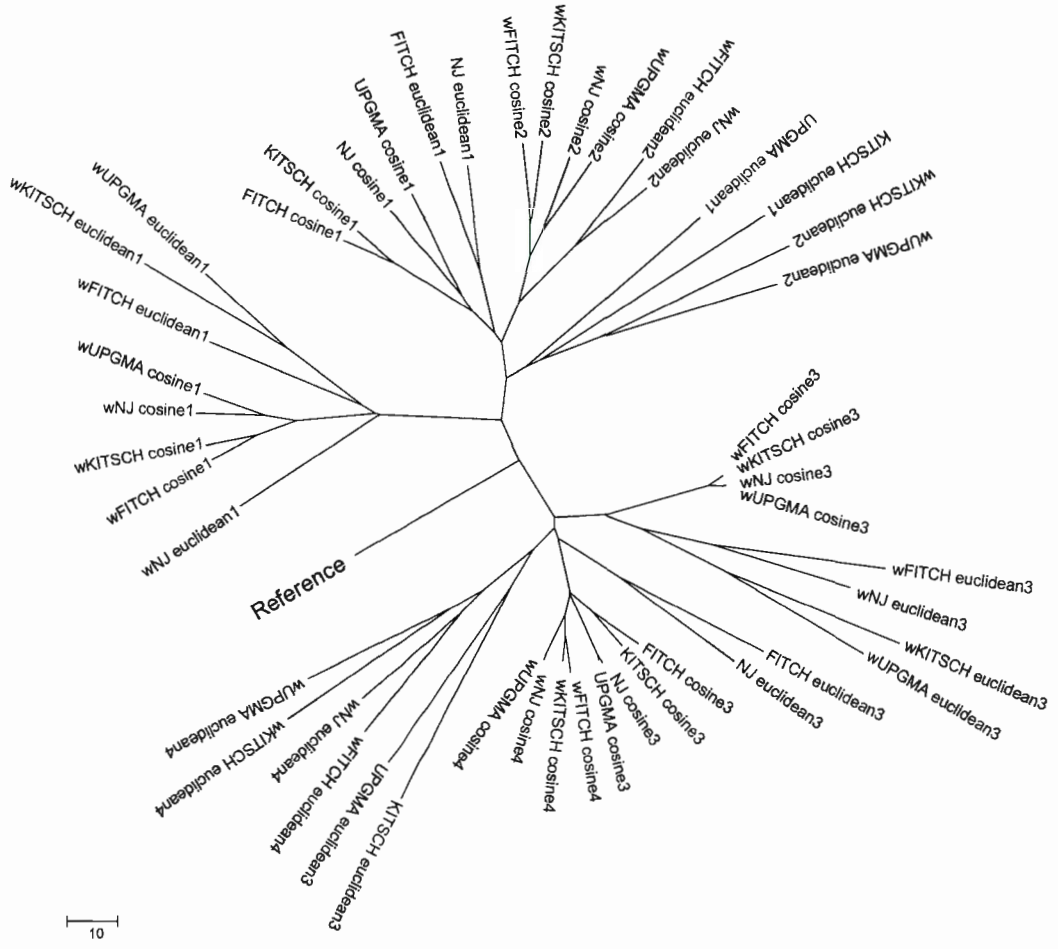


Figure 4.7 Classification hiérarchique des stratégies de regroupement hiérarchique de flux de travaux obtenue pour quatre distances pondérées ou non pondérées et quatre types d'encodage des flux de travaux (I, II, III et IV) décrit dans cette thèse, les distance cosine et Euclidienne et quatre algorithmes de regroupement hiérarchiques différents (Fitch, Kirsch, NJ et UPGMA) pour le jeu de données de *myExperiment* (n=100). L'utilisation du type d'encodage pondéré est indiquée par la lettre « *w* » précédant le nom de la méthode, alors que le type d'encodage est indiqué par le chiffre correspondant. Le taxon de référence représente le regroupement optimal.

Les résultats agrégés de nos simulations pour les deux ensembles de données expérimentales en terme de la distance RF moyenne entre les arbres de référence et les arbres de classification obtenus (Figure 4.8a) suggèrent une différence significative entre les résultats

correspondant à l'encodage non pondéré et ceux correspondant à l'encodage pondéré, pour l'encodage de type I (distance RF moyenne de 108.3 vs 102.4, $p<0.05$). Notez que les plus petites valeurs de la distance RF correspondent à des meilleurs résultats de regroupement. Aucune différence significative n'a été trouvée pour d'autres types d'encodage des flux de travaux en utilisant cette méthodologie. Lorsque les performances des quatre algorithmes de regroupement hiérarchique ont été considérées, aucune différence significative entre les distances RF moyennes correspondantes n'a été trouvée (Figure 4.8b). Néanmoins, l'algorithme de Fitch a fourni en général les plus petites valeurs de RF. Enfin, les résultats obtenus par les méthodes utilisant la distance Euclidienne et la distance cosine ont également été comparés (Figure 4.8c). Nous avons trouvé que l'utilisation de la distance cosine a conduit à de biens meilleurs résultats de regroupement globaux dans le cadre de la classification hiérarchique (la distance RF moyenne pour la distance cosine était de 101.3 vs 109.0 pour la distance Euclidienne; $p<0.001$). En résumé, les résultats obtenus pour les ensembles de données de *Armadillo* et de *myExperiment* montrent une meilleure classification hiérarchique en utilisant l'algorithme Fitch avec la distance cosine pondérée et l'encodage de type I.

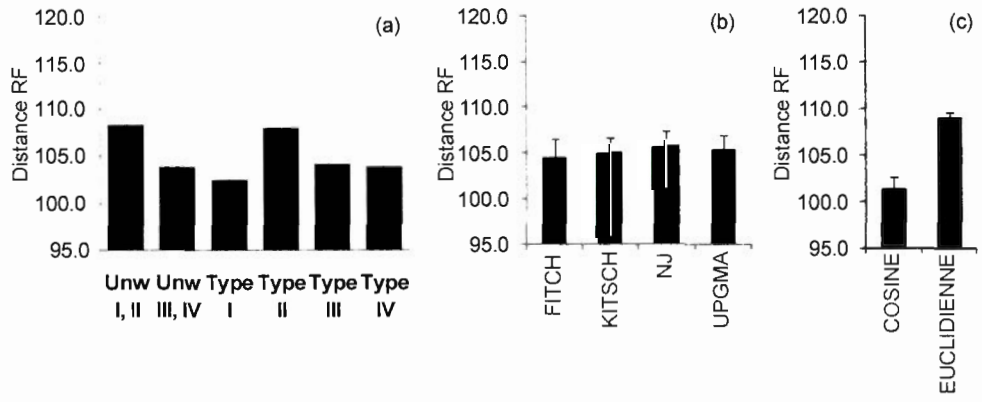


Figure 4.8 Résultats combinés de la classification hiérarchique obtenus pour les ensembles de données de *Armadillo* et de *myExperiment* (n=220). La moyenne de la distance topologique de Robinson et Foulds (RF± SEM) a été utilisée pour mesurer les performances des stratégies de regroupement. Les stratégies d'encodage non pondérées ont été respectivement désignées comme Unw I, II (résultats combinés pour les encodages non

pondérés de types I et II), ainsi que Unw III, IV (résultats combinés pour les encodages non pondérés de types III et IV). Le panneau (a) illustre l'effet du type d'encodage pour les stratégies non pondérées (deux barres) et pondérées (les quatre dernières barres); le panneau (b) - l'effet de l'algorithme de classification hiérarchique appliqué; le panneau (c) - l'effet de la mesure de distances.

4.10 Nouvelle mesure de support du regroupement par paires

Lors de l'exécution de nos simulations, nous avons observé une certaine variabilité dans l'attribution des flux de travaux individuels à leurs groupes, selon la partition aléatoire utilisée comme un point de départ pour les algorithmes de partitionnement *k*-means et *k*-medoids (voir la Figure 9c et 9d comme exemple). Il est connu qu'une seule valeur aberrante (*outliers*) peut grandement influencer les performances du regroupement de ces algorithmes de partitionnement (Hennig, 2008). Dans de nombreux cas, les heuristiques *k*-means et *k*-medoids atteignent seulement un minimum local qui est ensuite retourné comme la solution du regroupement (Makarenkov et Legendre, 2001). Nous avons observé que certaines paires de flux de travaux étaient plus susceptibles d'être affectées à la même classe ou à des classes différentes, indépendamment du nombre de classes proposées par le critère de classification de référence (*p.ex.* CH).

L'évaluation de la stabilité des solutions de partitionnement par l'algorithme *k*-means a été faite par différents chercheurs (Lange *et al.* 2004; Steinley, 2003, 2006, 2008, Cheng et Milligan, 1996, Ben-Hur *et al.* 2002). Steinley (2008) a évalué la cooccurrence de paires d'éléments et la cooccurrence des éléments entre chacune des classes pour estimer la qualité d'un partitionnement suite à l'exécution répété de *k*-means avec différentes valeurs d'initialisation. Par ailleurs, Ben-Hur *et al.* (2002) ont proposé un algorithme basé sur le coefficient de Jaccard permettant de détecter le manque de structure dans un regroupement hiérarchique. Dans ce cas, le calcul de la cooccurrence de paires d'éléments suite à des échantillonnages multiples a été utilisé. De manière similaire, Wang (2010) a proposé une estimation du nombre de groupes présents dans un partitionnement en divisant un jeu de données en deux parties et en évaluant l'instabilité de ces deux parties. Finalement, Hennig a développé différentes stratégies pour évaluer le support des différents groupes formés par un

partitionnement. La première stratégie (Hennig, 2007) implique l'utilisation du coefficient de Jaccard jumelé à des techniques de ré-échantillonnage (*p.ex. bootstrap*, *jittering* ou *subsetting*). Fang et Wang (2012) ont aussi évalué des techniques basées sur le *bootstrap* pour évaluer la stabilité globale du partitionnement en examinant le caractère aléatoire du partitionnement des éléments. Une deuxième stratégie (Hennig, 2008) implique la notion du point de dissolution d'un groupe et de la robustesse de l'isolation d'une solution de partitionnement. Dans ce cas, des éléments sont ajoutés au jeu de données de manière aléatoire. Milligan et Cheng (1996) ont aussi évalué l'ajout ou le retrait d'éléments d'un jeu de données pour confirmer la stabilité d'un partitionnement. Ainsi, une mesure de la stabilité des paires de flux de travaux lors du regroupement peut être introduite dans la situation où différents points de départ aléatoires sont considérés (*c.-à-d.* différentes partitions de départ). Une telle mesure va tenir compte de la probabilité de chaque paire d'éléments (à savoir les flux de travaux dans notre étude) d'être affectée à la même classe ou à des classes différentes.

Soit Q, le nombre de départs aléatoires (*c.-à-d.* les itérations, ou nombre d'exécution) de l'algorithme de partitionnement choisi (k-means ou k-medoids dans notre étude). Chaque départ aléatoire q génère une partition résultante, P_q, de classes non-chevauchantes dans lesquelles chacun des n flux de travaux considérés (ou chaque élément ou objet dans un cas plus général) est affecté à une classe. Un score de partitionnement, S_q, qui représente la qualité du partitionnement obtenu peut être associé à chaque partitionnement P_q. Par exemple, les critères CH, SI et logSS de regroupement considérés dans notre étude peuvent être utilisés comme des scores de séparation. Ensuite, une valeur de support par paires ou *pairwise support score (PS)* entre les flux de travaux w_i et w_j peut être définie comme suit (Équation 33):

$$PS(w_i, w_j) = \frac{\sum_{q=1}^{Q} S_{q,ij}}{\sum_{q=1}^{Q} S_q}, \tag{33}$$

où S_q est la valeur de l'indice de regroupement sélectionné, CH ou SI, est associé au partitionnement des flux de travaux n obtenus à l'itération q (S_q est égal à 1 si l'indice logSS est utilisé pour éviter les valeurs négatives et les valeurs de zéro); $S_{q,ij}$ est égale à S_q si les flux de travaux w_i et w_j sont affectés à la même classe du partitionnement des flux de travaux obtenu à l'itération q, sinon il est égal à 0. Un exemple de calcul du score *PS*, utilisant les flux de travaux W2 et W3 (Figure 4.1), est donné à la Figure 4.9a. Les éléments non-diagonaux de la matrice présentée dans la Figure 4.9b sont les scores *PS* obtenus pour les cinq flux de travaux bioinformatiques de la Figure 4.1. Les options suivantes ont été utilisées: 100 exécutions du logiciel de regroupement le critère de regroupement CH, et la distance cosine avec l'encodage de type I.

Alors, la valeur de *support individuel* de chacun des flux de travaux w_i, représentant la probabilité de w_i d'être un élément singleton dans sa catégorie, peut être définie comme suit (Équation 34; par exemple, elle définit les éléments diagonaux de la matrice de support de la Figure 4.9b):

$$PS(w_i) = \frac{\sum_{q=1}^{Q} S_{qi}}{\sum_{q=1}^{Q} S_q}, \tag{34}$$

où S_{qi} est égale à S_q si le flux de travaux w_i a été affecté à une classe singleton dans le partitionnement obtenu à l'itération q, sinon il est égal à 0. Ainsi, un flux de travaux qui a toujours été classé comme un élément unique d'une classe singleton aura le score individuel de soutien de 1 et tous les scores de support par paires (*PS*) de 0 (*p.ex.* voir le flux de travaux W4 à la Figure 4.9b). Si une paire de flux de travaux est toujours affectée à la même classe, le soutien par paire correspondant sera de 1.0 (*p.ex.* voir les flux de travaux W2 et W5 à la Figure 4.9b).

Finalement, une mesure de soutien globale du regroupement, *PSG*, pour un ensemble de flux de travaux $W = \{w_1, ..., w_n\}$, peut se définir comme suit (Équation 35):

$$PSG(W) = \frac{2(\sum_{i=1}^{n}\sum_{j=1}^{i-1}\max(PS(w_i,w_j),1-PS(w_i,w_j)) + \sum_{i=1}^{n}\max(PS(w_i),1-PS(w_i)))}{n^2}. \quad (35)$$

Enfin, un critère de support individuel pour chaque flux de travaux w_i($i = 1, \ldots, n$) peut être calculé comme suit (Équation 36):

$$PSG(w_i) = \frac{(\sum_{j=1(j\neq i)}^{n}\max(PS(w_i,w_j),1-PS(w_i,w_j))) + \max(PS(w_i),1-PS(w_i))}{n}. \quad (36)$$

Les deux premiers termes présents au numérateur des Équations 35 et 36 permettent de prendre en compte la proportion des flux de travaux qui sont jumelés en paires dans la même classe, ou se trouvent dans des classes différentes, suite à plusieurs répétitions (*random starts*) de l'algorithme de partitionnement. Par exemple, deux flux de travaux qui se retrouvent toujours dans la même classe, ou ne se retrouvent jamais dans la même classe, contribuent la même valeur maximale de 1 à la somme de l'Équation 36 ou à la double somme de l'Équation 35. Le second terme du numérateur de ces équations permet de tenir compte de la stabilité des éléments singletons. Finalement, chacune des équations est normalisée par le nombre total de termes présents dans son numérateur. Les indices *PSG* global (Équation 35) et *PSG* individuel (Équation 36) sont alors compris dans une tranche de valeurs variant de 0.5 à 1. Plus la valeur de *PSG* se rapproche de 1, plus la solution de partitionnement associée est robuste pour l'élément donné ou l'ensemble des éléments données. Steinley (2008) a aussi considéré l'application d'une mesure de support par paires. Cependant, Steinley n'a pas appliqué cette mesure pour obtenir une valeur de support global à un partitionnement, ni pour connaitre le support d'éléments individuels dans une classification.

Nous allons maintenant étudier comment les mesures de soutien définies dans les Équations (33-36) varient en fonction de l'algorithme de partitionnement choisi, du critère de regroupement et du nombre d'exécutions de la méthode de partitionnement. Tout d'abord, nous avons estimé que pour l'ensemble des cinq flux de travaux bioinformatiques présentés à

la Figure 4.1, la valeur de support global (*PSG*) (Équation 35) est égale à 0.93, tandis que les supports des flux de travaux individuels (Équation 36) sont les suivants : *PSG* (W1) = 0.93, *PSG* (W2) = 0.94, *PSG* (W3) = 0.86, *PSG* (W4) = 1.0 et *PSG* (W5) = 0.4. La meilleure valeur de support a été trouvée ici pour le flux de travaux W4 qui a toujours été classé seul, c'est-à-dire dans une classe singleton. Dans cet exemple, l'algorithme de partitionnement *k*-means, le critère de regroupement CH, la distance cosine et l'encodage de type I ont été les paramètres sélectionnés dans nos calculs.

Deuxièmement, nous avons considéré le jeu de données de *Armadillo* pour évaluer le comportement des indices *PS* et *PSG* lorsque les algorithmes de partitionnement *k*-means et *k*-medoids sont exécutés avec la distance cosine et l'encodage de type I (les deux dernières options ont fourni les meilleures performances de regroupement dans nos simulations décrites dans les sections précédentes). Pour les deux algorithmes, 1000 exécutions ont été réalisées. Les distributions des valeurs optimales des critères CH et SI retrouvées pour les 1000 essais indépendants des algorithmes *k*-means et *k*-medoids sont illustrés respectivement dans les Figures 4.9c et 4.9d. Le Tableau 4.3 présente également les valeurs de l'indice général de support, *PSG*, pour les deux algorithmes de partitionnement et les critères de regroupement CH, SI et logSS.

Tableau 4.3 Valeurs du support global du regroupement des flux de travaux, *PSG* (Équation 35), obtenues pour l'ensemble de données de *Armadillo* en utilisant comme paramètres la distance cosine et l'encodage de type I. Les résultats de partionnement pour les algorithmes *k*-means et *k*-medoids, les critères de classifications CH, SI et logSS sont présentés. Ces valeurs de support ont été calculées à partir de 1000 exécutions différentes pour chaque combinaison de paramètres.

Critères de regroupement	***k*-means**	***k*-medoids**
Calinski-Harabasz	0.932	0.714
Silhouette	0.664	0.840
logSS	0.653	0.823

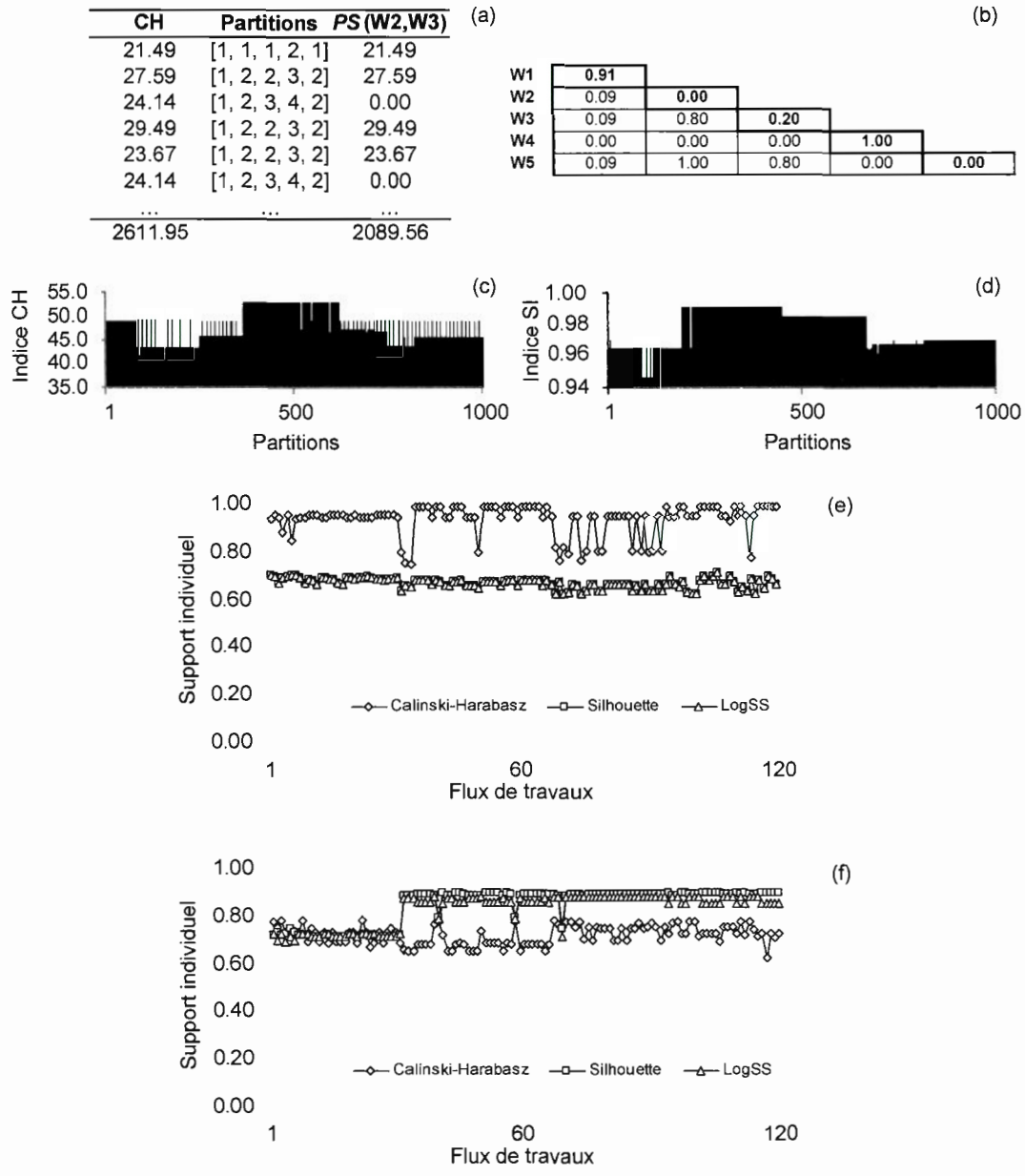


Figure 4.9 Les résultats de simulation évaluant le comportement des indices de support par paires définies dans cette thèse (voir les Équations 33 à 36). Le panneau (a) décrit

un exemple de calcul du score *PS* pour les flux de travaux W2 et W3, sélectionnés de l'ensemble des 5 flux de travaux bioinformatiques présentés à la Figure 4.1. Le panneau (b) présente la matrice des scores de *PS* calculée pour le même ensemble de 5 flux de travaux bioinformatiques. Une valeur de support de 1.0 dans la diagonale indique que l'élément concerné est toujours dans une classe singleton, alors qu'une valeur de support de 1.0 à une position non-diagonale indique que les deux éléments correspondants étaient toujours regroupés ensemble. Les panneaux (c) et (d) représentent respectivement la répartition du critère CH et SI obtenue pour le jeu de données de *Armadillo* (n=120) pour 1000 exécutions des algorithmes *k*-means (c) et *k*-medoids (d) en utilisant la distance cosine et l'encodage de type I. Le panneau (e) et le panneau (f) illustrent la distribution de l'indice de support *PSG* individuel pour les flux de travaux du jeu de données de *Armadillo* pour les algorithmes *k*-means et *k*-medoids, respectivement.

Nous avons constaté que dans le cas du regroupement par *k*-means le critère CH a produit les indices de support individuels et globaux les plus élevés pour les flux de travaux évalués par rapport aux critères Silhouette et logSS (*c.-à-d.* valeurs de support global *PSG* de 0.93 pour CH vs 0.66 pour SI et 0.65 pour logSS, p<0.0001; voir le Tableau 4.3 et la Figure 4.9e). Dans le cas de l'algorithme *k*-medoids, nous pouvons observer que l'utilisation de CH fournit des valeurs de support beaucoup plus faibles pour les flux de travaux individuels, ainsi que pour la valeur de l'indice de support global *PSG,* en comparaison aux critères SI et logSS (*c.-à-d.* la valeur de support *PSG* pour les flux de travaux était de 0.71 pour CH vs 0.84 pour SI et 0.82 pour logSS; p<0.0001; voir le Tableau 4.3 et la Figure 4.9f). Ces résultats concordent avec les résultats de nos simulations décrites à la section 4.8, où nous avons déterminé que dans les conditions expérimentales de cette étude, le critère CH donne de meilleurs résultats lorsque la classification par *k*-means est effectuée, alors que SI et logSS fournissent de meilleurs résultats dans le cadre du partitionnement par la méthode *k*-medoids.

4.11 Conclusions

Dans ce chapitre, nous avons défini quatre nouveaux types généraux d'encodage des flux de travaux et montré qu'ils peuvent conduire à différentes solutions de regroupement dans différentes solutions situations pratiques. Nos conclusions, fondées sur l'analyse de 220 flux

de travaux réels en bioinformatique suggèrent que l'ajout de l'information structurale à l'encodage ne conduit pas à de meilleures solutions de partitionnement. L'utilisation des ensembles de données générés par la plate-forme de *Armadillo* et provenant du site de *myExperiment* nous a permis de constater que la distance cosine, en association avec l'algorithme de partitionnement *k*-medoids et l'encodage de type présence-absence résultent aux valeurs les plus élevées de l'indice Rand, parmi toutes les stratégies de regroupement évaluées dans notre étude. Dans nos simulations, les critères d'optimisation Silhouette (SI) et logSS ont généralement surclassé le critère de Calinski-Harabasz (CH) dans le cadre du regroupement par *k*-medoids, tandis que l'indice de CH a généré de meilleurs résultats de classification dans le cas du regroupement utilisant par *k*-means. L'indice SI a donné des résultats de classification très stables lorsqu'il a été utilisé en conjonction avec la distance cosine. Nos analyses ont également démontré que l'application de poids peut avoir un impact majeur sur le nombre optimal de groupes retrouvés, ainsi que sur la composition des groupes obtenus en utilisant les algorithmes par partitionnement ou par classification hiérarchique. Dans l'ensemble, l'ajout d'un vecteur de poids représentant la moyenne des temps d'exécution des tâches nous a permis d'améliorer les résultats du regroupement. Comme nous l'avons également illustré, les encodages de types I et II, sur la base des informations de présence-absence, ont généralement surclassé les encodages plus complexes de types III et IV, tenant compte de l'information structurelle et des formats de données des ports d'entrée et de sortie des flux de travaux. Cette conclusion est en accord avec la conclusion de Wormbacher et Li (2010) qui ont fait valoir que l'encodage *N*-gram pendant le regroupement de flux de travaux est plus efficace que l'encodage de flux de travaux sous forme de graphes. Comme chacune des tâches composant les flux de travaux (*c.-à-d.* les méthodes) sont elles-mêmes caractérisées par leurs propres paramètres, qui peuvent varier d'une exécution des flux de travaux à une autre, l'ensemble du flux de travaux peut également être encodé comme données de dissimilarité à trois niveaux, et ensuite analysé en utilisant des techniques de classification structurelles appropriées (Vicari et Vichi, 2009).

La classification des flux de travaux effectuée en utilisant des méthodes hiérarchiques était en faveur de l'encodage de type I en association avec la distance cosine. Dans l'avenir, il serait

intéressant de comparer les classifications de flux de travaux hiérarchiques obtenues au moyen des méthodes de distances avec celles construites à l'aide d'approches de maximum de parcimonie (*MP*) et de maximum de vraisemblance (*ML*). Le principal avantage des méthodes MP et ML est qu'elles peuvent être appliquées directement sur la matrice bidirectionnelle objets-variables, sans avoir à utiliser des distances approximatives. On peut aussi en calculer directement des distances pondérées entre les objets. En outre, des valeurs de soutien par *bootstrap* des arbres inférés par ces méthodes peuvent également être calculées.

Finalement, dans ce chapitre, nous avons également introduit et validé, une nouvelle mesure de stabilité des solutions de regroupement par paires, *PS*, qui peut être appliquée dans des situations où une série de logiciels indépendants est utilisée pour réaliser la classification d'éléments (par exemple lorsque différentes partitions aléatoires sont considérées comme entrée d'un algorithme de partitionnement). Une telle mesure, évaluée sur plusieurs exécutions indépendantes, permet de refléter la probabilité de chaque paire d'éléments à être affectée à la même classe. En outre, nous avons introduit un indice de support global, *PSG*, permettant d'estimer le soutien global de la solution de regroupement proposée, ainsi que le soutien global des éléments individuels (flux de travaux dans notre cas). Dans notre étude, nous nous sommes limités aux flux de travaux du domaine de la bioinformatique. Il serait important d'étudier le comportement des indices présentés (*PS* et *PSG)* en utilisant des flux de travaux liés à d'autres domaines d'applications, tels que l'économie, les affaires ou la médecine, car ils peuvent avoir des propriétés structurelles et informatiques différentes. Cette nouvelle mesure de la stabilité par paires pourrait également être comparée avec d'autres mesures de stabilité de regroupement connues, comme par exemple celles décrites par Hennig (2007, 2008).

4.12 Perspectives

Nous avons choisi de travailler avec des algorithmes de classification non-supervisés tels que *k*-means et *k*-medoids pour les deux raisons suivantes. La première est liée à la capacité de ces algorithmes de prendre en compte des milliers de tâches (Bharathi *et al.,* 2008; Ramakrishnan et Gannon, 2008), alors que les méthodes de graphes et celles basées sur la

distance d'édition ont des complexités algorithmiques exponentielles (Bunke *et al.,* 2002; Conte *et al.,* 2003). Deuxièmement, contrairement aux méthodes de graphes, plusieurs flux de travaux ont des éléments communs, mais pas nécessairement connectés entre eux, rendant ces approches difficilement applicables (Santos *et al.,* 2008). Vu les bons résultats associés à la distance cosine, une approche hybride, combinant les méthodes hiérarchiques, ainsi que les méthodes de partitionnement, comme défini dans l'algorithme CLUTO (Zhao *et al.*, 2005), pourrait être appliquée pour classifier des flux de travaux. Il faudrait, dans ce cas, ajouter à cette méthode un vecteur de poids durant le calcul des distances pour prendre en compte soit le temps des méthodes soit d'autres paramètres importants des flux de travaux considérés.

Finalement, concernant le critère de support des éléments, deux autres jeux de données ont aussi servi à valider ce nouveau critère de support. Il s'agit du jeu de données de *Iris* (Fisher, 1936) composé de 4 classes et du jeu de données de *Zoo* (Forsyth, 1990) composé de 7 classes. Les résultats de cette analyse additionnelle sont présentés dans l'annexe D. Des simulations Monte-Carlo comprenant d'autres cas d'usages pourraient aussi être réalisées pour déterminer les limites d'applications de cette méthodologie.

CHAPITRE V

UTILISATION PRATIQUE DE LA PLATE-FORME ARMADILLO

5.1 Introduction

Dans ce chapitre, nous introduirons des exemples de flux de travaux créés et exécutés dans la plate-forme *Armadillo* qui ont mené à trois publications. La première partie du chapitre traite des micro-ARNs dans la régulation génétique chez le blé. La plate-forme *Armadillo* a été exploitée pour automatiser plusieurs étapes critiques pour permettre l'identification de micro-ARNs et de gènes cibles associées, ainsi que des étapes préliminaires de filtrage et de conversion de données. La deuxième partie du chapitre traite de l'évolution du virus de l'immunodéficience de type I (VIH type I) chez des femmes enceintes. La plate-forme *Armadillo* a alors servi à rapidement compléter les expériences *in silico* menant à la réalisation de cette étude.

5.2 Prédiction de micro-ARNs associés aux stress abiotiques chez le blé

Le blé (Triticum *aestivum L.*) est responsable de 19 % de l'apport alimentaire sur Terre (Ray *et al.,* 2013). Cette céréale a un génome hexaploide très volumineux (~17 Gbp) avec 80% de séquences répétitives (Hernandez *et al.,* 2012). De plus, le blé est très sensible aux facteurs abiotiques tels que le froid, la sècheresse, le sel et l'aluminium (Jones-Rhoades *et al.,* 2006). Différents réseaux de gènes sont impliqués dans la résistance du blé à ces facteurs, mais leur régulation et leur importance sont encore mal connues (Jones-Rhoades *et al.,* 2006).

Les micro-ARNS (*miRNAs*) sont une classe de petits ARNs présents chez les eucaryotes qui permettent la répression d'ARNs endogènes (Lee *et al.,* 1993). Chez les plantes, ces ARNs de 20-24 nucléotides contrôlent l'expression de différents facteurs de transcription, de même que l'expression de protéines de réponses au stress. Ces ARNs sont aussi impliqués dans le

développement et la croissance chez les végétaux (Rogers et Chen, 2013). Plusieurs familles de micro-ARNs (revue par Rogers et Xuemei, 2013) ont été démontrées comme régulateurs de ces réseaux chez les plantes (Sunkar et Zhu, 2004). Malheureusement, chez le blé, seulement 42 micro-ARNs ont été identifiés jusqu'à maintenant (mirBase révision 19, Kozomara et Griffiths-Jones, 2014).

5.2.1 Description de l'étude

Le but de ce projet était l'identification et la découverte de nouveaux micro-ARNs chez le blé. La méthodologie préconisée pour identifier ces micro-ARNs potentiels chez le blé a été le séquençage de nouvelle génération (*NGS*) de différents types de blé soumis à des stress abiotiques (Figure 5.1 et Tableau 5.1). Par la suite, différentes méthodes de prédiction des micro-ARNs candidats *in silico* ont été utilisées (Tableau 5.2). Finalement un portail présentant les résultats de l'étude de Agharbaoui *et al.* (2015) et incluant des flux de travaux permettant une reproduction des analyses a été réalisé (Lord *et al.*, 2015b; section 5.3).

Tableau 5.1 Librairies et total des micro-ARNs candidats identifiés chez le blé dans l'étude de Agharbaoui *et al.* (2015).

Conditions et stress abiotiques	Micro-ARNs candidats
Tous les tissus et conditions expérimentales	**1369**
*Librairie 1.*Tissus aériens en phase végétative en conditions normales	**1109**
Librairie 2. Tissus aériens en phase végétative vernalisés à 4°C pendant 2h, 24 h, ou entre 1 et 7 semaines	**1227**
Librairie 3. Tissus aériens de plants en phase reproductive (plants vernalisés dé-acclimatés pendant 3-5 semaines)	**902**
Librairie 4. Tissus aériens de plants de 4 semaines exposés à 200 mM de NaCl	**1163**
Librairie 5. Tissus racinaires de plants de 4 semaines exposés à 200 mM de NaCl	**1107**
Librairie 6. Tissus aériens en phase végétative sous conditions normales	**1104**
Librairie 7. Tissus aériens de plants adaptés au froid pendant 4 semaines	**1100**
Librairie 8. Tissus racinaires exposés à 5 µM d'aluminium	**994**
Librairie 9. Tissus racinaires sous conditions normales	**939**
Librairie 10. Tissus racinaires exposés à 50 µM d'aluminium	**988**

Le volume des données brutes provenant du séquençage des 10 librairies (Tableau 5.1) totalisait ~19 gigaoctets (Leclercq, 2012). Ainsi, trois étapes (Figure 5.1) ont été réalisées en utilisant des flux de travaux (Agharbaoui *et al.*, 2015; Leclercq, 2012). La première étape (Figure 5.1a) a consisté à la recherche de micro-ARNs candidats chez le blé. Les deux dernières étapes ont, quant à elles, consisté à la recherche d'informations sur les cibles potentielles (gènes ou transcrits) de la régulation par ces micro-ARNs candidats (Figure 5.1b et Figure 5.1c) et sur de nouvelles séquences semblables aux micro-ARNs candidats identifiés.

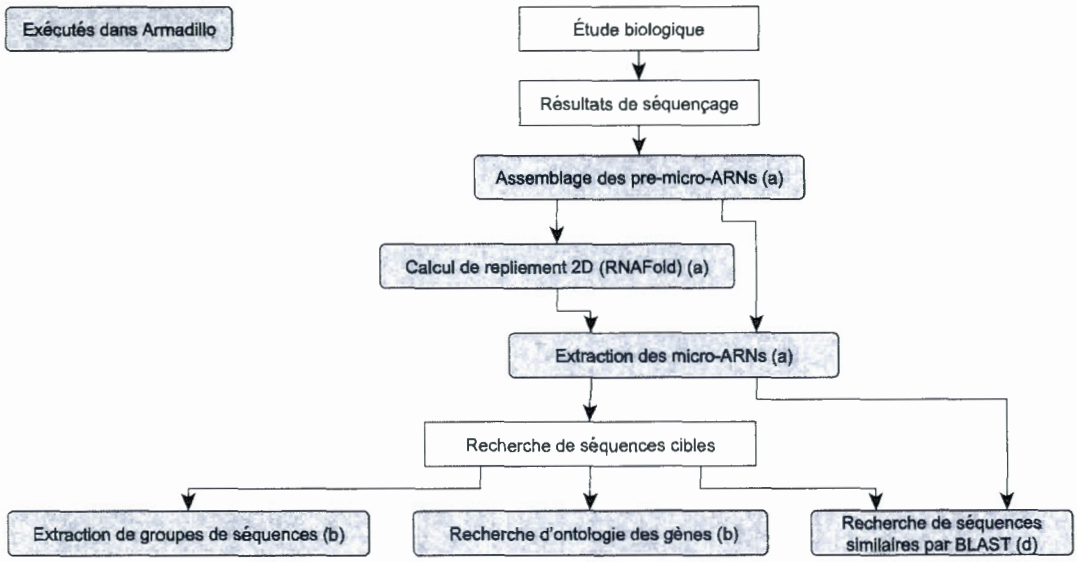


Figure 5.1 Vue d'ensemble de l'étude phylogénomique sur les micro-ARNs du blé. En gris, sections de l'étude ayant requis l'utilisation de flux de travaux exécutés à l'aide de la plate-forme *Armadillo*. Le flux de travaux utilisé en : (a) pour l'assemblage des séquences, le repliement 2D de celles-ci et l'extraction de micro-ARNs candidats est présenté à la Figure 5.2; en (b) pour la recherche d'ontologie des gènes est présenté à la Figure 5.3; et en (c) pour la recherche de séquences similaires est présenté à la Figure 5.4.

5.2.2 Flux de travaux utilisés

Trois flux de travaux principaux ont été conçus lors de l'étude initiale des micro-ARNs. Ils sont présentés aux Figures 5.2 à 5.4. Un quatrième flux de travaux, relié à la présentation des

résultats et à la recherche de nouveaux micro-ARNs candidats, est présenté dans la section 5.3.

5.2.2.1 Recherche de micro-ARNs candidats par un flux de travaux conceptuel

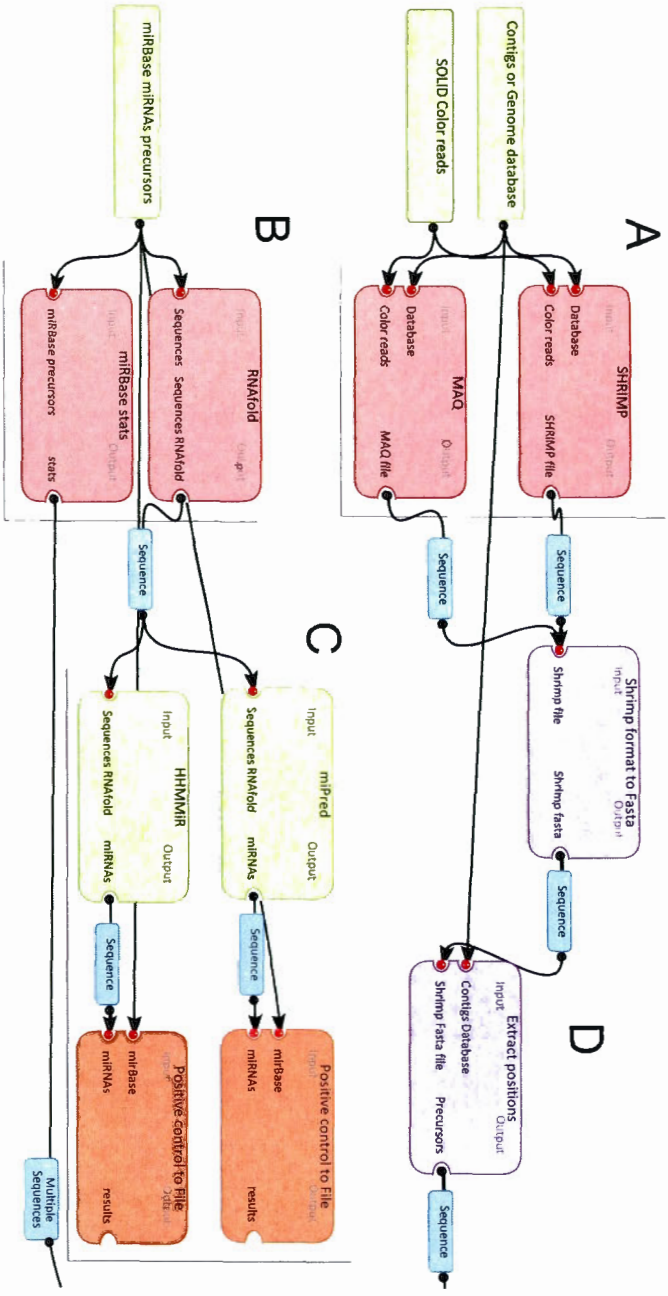
Une première conceptualisation du travail à effectuer a été réalisée dans la plate-forme *Armadillo* (Figure 5.2). Pour ce faire, tous les composants étaient de type « *Custom Program* » permettant d'ajouter des logiciels externes à la plate-forme *Armadillo* (voir Lord *et al.,* 2012). Ainsi, après cette conceptualisation, différents logiciels ont été inclus dans le flux de travaux. Leurs paramètres d'exécution et leur temps d'exécution ont ainsi été validés sur un échantillon des séquences présentes dans les dix librairies de séquences. Après exécution locale de ce flux de travaux, les paramètres d'exécution optimaux ont permis de compléter de façon parallèle, l'analyse des résultats sur 48 ordinateurs en grappes (12 x 2 cpu, 4 giga-octets de mémoire vive et 16 x 8 cpu – 8 giga-octets de mémoire vive, 20 x 4 cpu, 8 giga-octets de mémoire vive) (Leclercq, 2012).

Les détails de ce flux de travaux conceptuel (Figure 5.2) sont les suivants. Tout d'abord, un jeu de données contenant les séquences du blé de GrainGenes (Carollo *et al.,* 2005), Komugi (http://www.shigen.nig.ac.jp/wheat/komugi/), WheatDB (http://wheatdb.ucdavis.edu/), TAGI (http://compbio.dfci.harvard.edu/tgi), TIGR (Childs *et al., 2*006) ainsi que de la base de données NCBI a été collecté et groupé en collection de séquences uniques (ou *contigs*) et exploité comme référence du blé (Figure 5.2a). Par la suite, les séquences (*shorts reads*) issues du séquençage des génomes du blé ont été alignées à cette collection et les séquences résultantes ont pu être extraites en passant par différents filtres permettant de limiter la taille des séquences, de valider la qualité ou d'extraire les données dans le format de données *Fasta* (Figure 5.2d). De manière concurrente, une deuxième partie de ce flux de travaux a permis de tester les paramètres des logiciels de prédiction de micro-ARNs miPred (Batuwita et Palade, 2009) et HHMMiR (Kadri *et al.,* 2009) (Figure 5.2c) prenant en entrée un repliement 2D de séquences en épingle à cheveux (*hairpins*). Puisque la plupart de ces logiciels produisent en sortie des types de données de types séquences (*Sequences*) ou groupes de séquences (*MultipleSequences*), leur utilisation a été possible dans le flux de travaux sans ajout de nouvelles fonctionnalités (ou nouveaux formats) à la plate-forme.

Par la suite, les mêmes étapes de repliement de la séquence en structure bidimensionnelle (Figure 5.2e) et inférence de micro-ARNs candidats (Figures 5.2f et 5.2g) ont été réalisées sur nos propres jeux de données. Une vérification manuelle des résultats a été effectuée pour comparer les résultats des différents prédicteurs. Seulement deux prédicteurs sont présentés dans le flux de travaux de la Figure 5.2. Cependant, d'autres logiciels de prédiction de micro-ARNs ont été utilisés dans l'étude de Agharbaoui *et al.* (2015), soit mirDup et MirCheck (Tableau 5.2).

Tableau 5.2 Logiciels d'inférence des micro-ARNs candidats utilisés dans l'étude de Agharbaoui *et al.* (2015).

Logiciels	**Références**
HHMMiR	Kadri *et al.* (2009)
Mipred	Batuwita et Palade (2009)
mirDup	Leclercq *et al.* (2013)
MirCheck	Jones-Rhoades et Bartel (2004), Meyers *et al.* (2008)


A
Contigs or Genome database
SOLID Color reads
SHRIMP
Database
Color reads
SHRIMP file
MAQ
Database
Color reads
MAQ file
Sequence
Sequence
Shrimp format to Fasta
Shrimp file
Shrimp fasta
Sequence
D
Extract positions
Contigs Database
Shrimp Fasta file
Precursors
Sequence
B
miRBase miRNAs precursors
RNAfold
Sequences
Sequences RNAfold
miRBase stats
miRBase precursors
stats
Sequence
C
miPred
Sequences RNAfold
miRNAs
Output
HHMMiR
Input
Sequences RNAfold
miRNAs
Output
Sequence
Sequence
Positive control to File
mirBase
miRNAs
results
Positive control to File
mirBase
miRNAs
results
Multiple Sequences

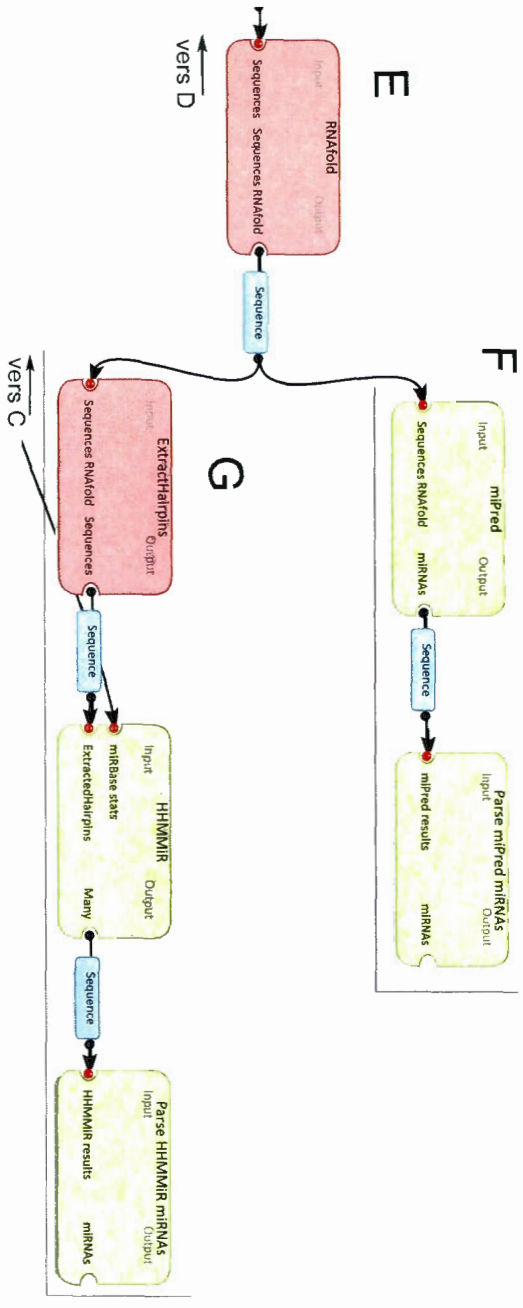


Figure 5.2 Flux de travaux conceptuel permettant la recherche de micro-ARNs candidats à partir de données de séquençage. Dans le panneau (A), les logiciels SHRiMP (Rumble *et al.*, 2009) et MAQ (Li et Durbin, 2014) permettent d'effectuer l'alignement (*mapping*) des *séquences* par rapport à une référence. Au panneau (B), des micro-ARNs

vérifiés, provenant de la banque de données miRBase (Kozomara et Griffiths-Jones, 2014), sont repliés par le logiciel RNAFold (Hofacker *et al.,* 2009) en structures en épingles à cheveux bidimensionnelles. Au panneau (C), inférence de micro-ARNs candidats à partir de miPred et HHMMiR (voir Tableau 5.2). Ces séquences serviront de contrôles positifs lors de l'analyse des résultats. Au panneau (D), conversion et filtre des différentes séquences assemblées lors de l'étape A. Au panneau (E), repliement 2D des séquences trouvées dans les différentes librairies avant inférence des micro-ARNs candidats par les logiciels miPred (panneau F) ou HHMMiR (panneau G). Finalement, pour ces dernières étapes (panneaux F et G), on conserve une trace des micro-ARNs trouvés à même le flux de travaux (étapes identifiées par le mot-clé *parse*). Notez que pour ce flux de travaux, les différents logiciels ont été ajoutés en utilisant l'option d'inclusion de logiciels externes à la plate-forme *Armadillo* (*c.-à-d.* le composant *Your Program* →*Custom Program* de la boîte à outils permettant différentes options et configurations).

5.2.2.2 Recherche de l'ontologie des gènes cibles des micro-ARNs candidats

Un deuxième flux de travaux (Figure 5.3) a été utilisé suite à l'obtention des micro-ARNs candidats (Tableau 1.1). Ce flux de travaux a servi à rechercher l'ontologie des gènes cibles liés aux micro-ARNs candidats *c.-à-d.* la recherche de la fonction et de la localisation de ceux-ci (*biological process, molecular function, cellular compoment*). Dans ce flux de travaux, on effectue de manière concurrente l'exécution du logiciel de recherche de séquences par similarité BLAST (Altschul *et al.* 1990) sur deux banques de données : *NCBI* (voir Johnson *et al.,* 2008; Valentin *et al.*, 2010) et *SwissProt* (Boeckmann *et al.*, 2003) (Figure 5.3a). Ensuite, la recherche de l'ontologie des gènes sur le site *EBI* est effectuée en prenant en entrée les résultats de cette recherche (Figure 5.3b). Toutefois, si l'on obtient aucun résultat provenant de la recherche dans la banque de données *SwissProt*, une recherche subséquente sera réalisée dans la banque de données *TrEMBL* (Boeckmann *et al.*, 2003) avant une nouvelle recherche ontologique (Figure 5.3c). Il est important ici de mentionner que l'aspect conditionnel du flux de travaux (*If*) permet au chercheur de ne pas se préoccuper de l'exécution sur plusieurs banques de données, tout en sauvant du temps d'exécution.

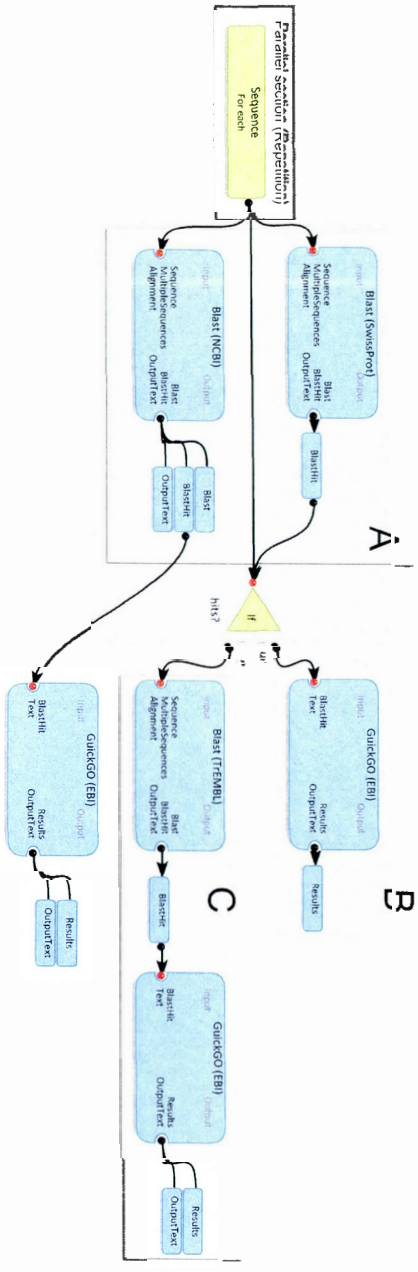


Figure 5.3 Flux de travaux conditionnel de recherche de l'ontologie des gènes cibles chez le blé. Dans le panneau (A), exécution concurrente de l'algorithme de recherche de séquences BLAST sur les bases de données de séquences NCBI et SwissProt. Par la suite, une recherche de l'ontologie de ces gènes (ou protéines) sur ces résultats est effectuée si la

recherche sur SwissProt retourne des résultats (panneau B), ou si rien n'est retourné, une nouvelle exécution de la recherche par BLAST mais sur la base de données TReMBL (voir Boeckmann *et al.*, 2003) et recherche ontologique sur ces résultats (panneau C).

5.2.2.3 Recherche locale de la similarité entre les gènes cibles et les micro-ARNs

Le troisième flux de travaux utilisé dans cette étude (Figure 5.4) a permis de valider les paramètres et de lancer, directement depuis un ordinateur de bureau, l'analyse de milliers de séquences à l'aide du service Web de la méthode BLAST provenant du serveur *NBCI* (Johnson *et al.*, 2008). Dans cette exécution, l'aspect concurrence était important puisque les jeux de données contenaient un total de 1 369 séquences de micro-ARNs candidats et 6 841 séquences cibles. Ce qui nécessite une série de 8 200 requêtes BLAST. Cependant, la recherche devait porter de façon concurrente sur deux bases de données distinctes. La première base de données est celle des ESTs collectionnés et la seconde est celle du serveur NCBI. Ainsi, un mécanisme de gestion concurrente et de répétition d'analyses a été exploité dans *Armadillo*. Ce qui a permis de diviser ce flux de travaux en plusieurs sous-tâches exécutées indépendamment.

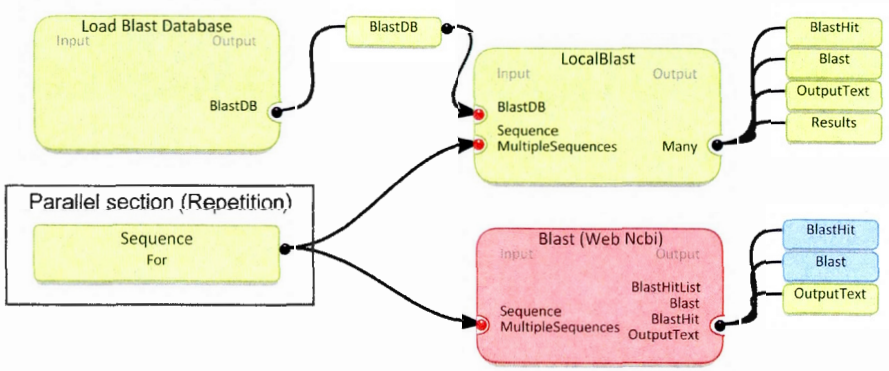


Figure 5.4 Flux de travaux pour l'identification des séquences cibles à l'aide de la méthode de recherche de séquences BLAST (Altschul *et al.*, 1990) et d'une base de données locale de séquences (*en haut*). Dans ce flux de travaux, une recherche de similarité de séquences est exécutée de manière concurrente contre la base de données de NCBI et contre

une base de données locale construite à partir de séquences du blé d'une étude antérieure (Houde *et al.*, 2006).

5.3 Portail Web WMP: *Wheat Micro-RNAs portal*

5.3.1 Préface

Nous présentons dans cette section un nouveau portail Web permettant la recherche sur les micro-ARNs chez le blé. Dans un premier temps, ce portail présente les résultats de l'étude de Agharbaoui *et al.* (2015). De plus, deux des outils utilisés lors de l'étude sont présentés sous la forme d'un serveur Web utilisant des flux de travaux de la plate-forme *Armadillo*. Le texte de cette section a été soumis pour publication dans la revue *Bioinformatics* (Lord *et al.*, 2015b).

Les facteurs abiotiques tels que le froid, la sécheresse, le sel et l'aluminium limitent la croissance et le développement des plantes. De plus, ces facteurs réduisent le rendement de plusieurs cultures importantes, dont la croissance et le développement du blé *Triticum aestivum L.* Plusieurs études ont identifié un grand nombre des micro-ARNs responsables de la régulation des gènes associés au stress. Cependant, il y a un besoin urgent de portails Web permettant d'avoir une vue concise et globale pour l'analyse et l'exploration des caractéristiques des micro-ARNs candidats et de leurs gènes cibles. Dans cette étude, nous proposons un regroupement de ressources permettant de visualiser et d'analyser l'expression des micro-ARNs dans le blé pour les conditions de stress abiotiques. Le portail Web permet des requêtes sur 10 bibliothèques de petits ARNs, de nouvelles séquences de micro-ARNs du blé prédites *in silico*, ainsi que la comparaison de profils d'expression de petits ARNs. Ce portail offre également aux chercheurs un accès direct aux logiciels de prédiction de micro-ARNs spécialement adaptés à la détection de micro-ARNs candidats chez le blé, chez les monocotylédons ainsi que chez d'autres espèces de plantes. Le site Web est disponible à l'adresse url suivante : < http://wheat.bioinfo.uqam.ca >

5.3.2 Introduction

L'expression des micro-ARNs varie au cours du développement végétal en fonction des différents tissus et génotypes. Le séquençage de nouvelle génération (*NGS*) offre une avenue

intéressante pour générer et rechercher de nouveaux micro-ARNs qui sont exprimés sous différentes conditions expérimentales (Kurtoglu *et al.*, 2014). Cette approche permet la création de plusieurs bibliothèques (ou librairies) de petits ARNs exprimés comprenant des milliers de séquences. Pour le blé (*Triticum aestivum*), qui possède un génome hexaploïde, différentes librairies sont ainsi disponibles pour la recherche, incluant des données provenant de plants de blé soumis au froid, à l'infection au *fusarium*, ou exposés à des chaleurs excessives simulant la sécheresse (voir par exemple, Kurtoglu *et al.*, 2014 et Agharbaoui *et al.*, 2015).

Cependant, l'accessibilité des données ainsi obtenues est en général difficile vu l'importance et le volume de données disponibles. *MiRBase*, la base de données de référence en matière de micro-ARNs, est la principale ressource pour accéder à des micro-ARNs cités dans la littérature chez les plantes. Cependant, cette ressource ne contient que 42 micro-ARNs de blé validés expérimentalement dans sa dernière révision (révision 20; Kozomara et Griffiths-Jones, 2014). En outre, l'interface de miRBase rend difficile la recherche et l'évaluation de ces micro-ARNs en fonction de leurs origines biologiques et des conditions expérimentales dans lesquelles elles se sont retrouvées exprimées.

Nous présentons, dans ce travail, un portail Web permettant la présentation et l'analyse de l'expression différentielle de micro-RNAs retrouvés chez le blé. Dix bibliothèques de petits ARNs générées dans l'étude de Agharbaoui *et al.* (2015) sont ainsi disponibles pour la recherche. Ce portail permet également aux chercheurs d'accéder directement aux pré-micro-ARNs, aux structures en épingle à cheveux (repliement 2D). Il permet aussi la recherche de nouveaux micro-ARNs candidats *in silico* à l'aide de deux outils de prédiction différents dont miRDup (Leclercq *et al.*, 2013) et MirCheck (Jones-Rhoades *et al.*, 2006).

5.3.3 Contenu et statistiques de la banque de données

La base de données actuelle présente des données de dix petites bibliothèques d'ARNs, produites à partir de plantes cultivées sous différents stress abiotiques et stades de développement (Agharbaoui *et al.*, 2015). Il permet ainsi d'avoir accès aux caractéristiques de 1369 micro-ARNs candidats, 6 481 gènes cibles associés à ces derniers, à 1.4 million de ESTs et à 127 039 groupes de séquences Uniref collectées à partir de sept principales bases

de données portant sur le génome du blé. Au total, 168 834 petits ARNs exprimés dans les différentes bibliothèques sont présentés, ainsi que 466 micro-ARNs conservés chez d'autres végétaux selon le portail miRBase. Pour chacun des micro-ARNs prédits, les gènes cibles candidats ont été identifiés en utilisant le logiciel *Tapir* (Bonnet *et al.,* 2010). Seules des séquences avec un score *Tapir* de moins de 3 (indiquant un appariement de nucléotides presque parfait) ont été considérées dans cette version de la base de données. L'ontologie (Gene Ontology, GO) des gènes cibles a aussi été réalisée menant à un enrichissement de l'annotation de ces séquences. Ainsi, 19 418 processus biologiques (*biological process*), 10 980 composants cellulaires (*cellular component*), et 10 478 fonctions moléculaires (*molecular function*) ont été associés à ceux-ci. Pour améliorer la visualisation et l'identification des micro-ARNs candidats, la structure en épingle à cheveux de chacun des pre-miRNAs a été générée en utilisant la suite *Varna* (Darty *et al.,* 2009). La Figure 5.6 met en évidence les principales caractéristiques présentées pour chaque micro-ARN candidat.

5.3.4 Interface utilisateur

Six options principales sont disponibles pour l'utilisateur du portail: 1) *Recherche de base* (*Search*) : l'utilisateur peut rechercher à l'aide d'un ou plusieurs mots-clés provenant des catégories suivantes: micro-ARN validé, pré-miARN ou structures en épingle à cheveux, la structure des motifs (*dotbracket notation*), les gènes cibles associés aux micro-ARNs, les identifiants Uniref ou le nom officiel d'un micro-ARN, les séquences d'ESTs ou leurs identifications, des mots-clés contenus dans la description de *Gene Ontology* (GO) ou leurs identificateurs (par exemple GO:0008152). 2) *Recherche avancée* (*Advanced search*) : un utilisateur peut rechercher des micro-ARNs exprimés de manière différentielle entre les conditions expérimentales. Ici, l'utilisateur peut sélectionner une condition expérimentale et sélectionner la liste des motifs d'expression de micro-ARNs à explorer soit : régulés à la hausse (*upregulated*), régulés à la baisse (*downregulated*), exprimés de manière différentielle ou non, ou encore retrouvés dans chacune des conditions. Il peut également ajuster les paramètres de signification statistique (*p-value*) et la valeur minimale de transcription des différents micro-ARN retrouvés dans les bibliothèques cibles. 3) Les bibliothèques et les conditions d'entrée permettent d'explorer l'ensemble des micro-ARNs exprimés dans une bibliothèque ou encore sous une contrainte (*c.-à-d.* une valeur d'expression minimale ou

maximale). 4) Dans le menu des données (*Data*) : l'utilisateur peut accéder directement, soit aux micro-ARNs candidats retrouvés dans les dix différentes bibliothèques du blé ou encore, seulement afficher les micro-ARNs conservés, ou l'ensemble des gènes et ESTs cibles associés aux micro-ARNs candidats. Ce menu donne également accès à divers regroupements (*clusters*) de micro-ARNs candidats. Ces regroupements de micro-ARNs sont établis en fonction de la similarité de séquences des micro-ARNs, de leur origine et de leurs gènes cibles. Ce premier jeu de données contient un ensemble 345 micro-ARNs identifiés dans l'étude de Agharbaoui *et al.* (2015). 5) Le menu *d'expression* (*Tools→Expression database*) : permet à l'utilisateur de rechercher à l'aide d'une seule requête, plusieurs séquences dans notre base de données contenant les micro-ARNs candidats et les petit ARNs exprimés. 6) Le menu *Outils* (*Tools*): donne accès à deux outils de prédiction des micro-ARNs, soit *Mircheck* v1.0 (Jones-Rhoades et Bartel, 2004) et *Mirdup* v1.2 (Leclercq *et al.*, 2013) (voir les Figures 5.3 et 5.5). Pour ces deux prédicteurs, l'utilisateur peut évaluer ses propres séquences de micro-ARNs, précurseurs de micro-ARNs et structures secondaires en épingles à cheveux. L'outil Mircheck permet deux paramétrages différents (paramètres par défaut utilisés par Jones-Rhoades et Bartel (2004), ou encore les paramètres optimaux utilisés par Meyers *et al.* (2008) pour l'annotation de micro-ARNs chez les végétaux). Le prédicteur Mirdup (basé sur *AdaBoost*) permet de sélectionner parmi quatre jeux de données d'apprentissage. Ces jeux de données, validés expérimentalement, sont tirés de séquences de micro-ARNs et proviennent de la banque de données *miRBase* (Kozomara et Griffiths-Jones, 2014). Les quatre jeux de données disponibles sont : l'ensemble de miRBase, *Viridiplantae* (plante), monocotylédones ou uniquement des séquences retrouvées chez le blé (*Triticum aestivum L.*). Ces deux outils ont été réalisés à l'aide de la plate-forme *Armadillo*. Nous utilisons une forme réduite de la plate-forme pour exécuter le flux de travaux décrit à la Figure 5.5 sous la forme d'un formulaire Web. À terme, nous voulons permettre à l'utilisateur de créer lui-même son flux de travaux d'analyse permettant ainsi de reproduire les conditions de l'étude, sans y introduire de biais reliés aux méthodes employées.

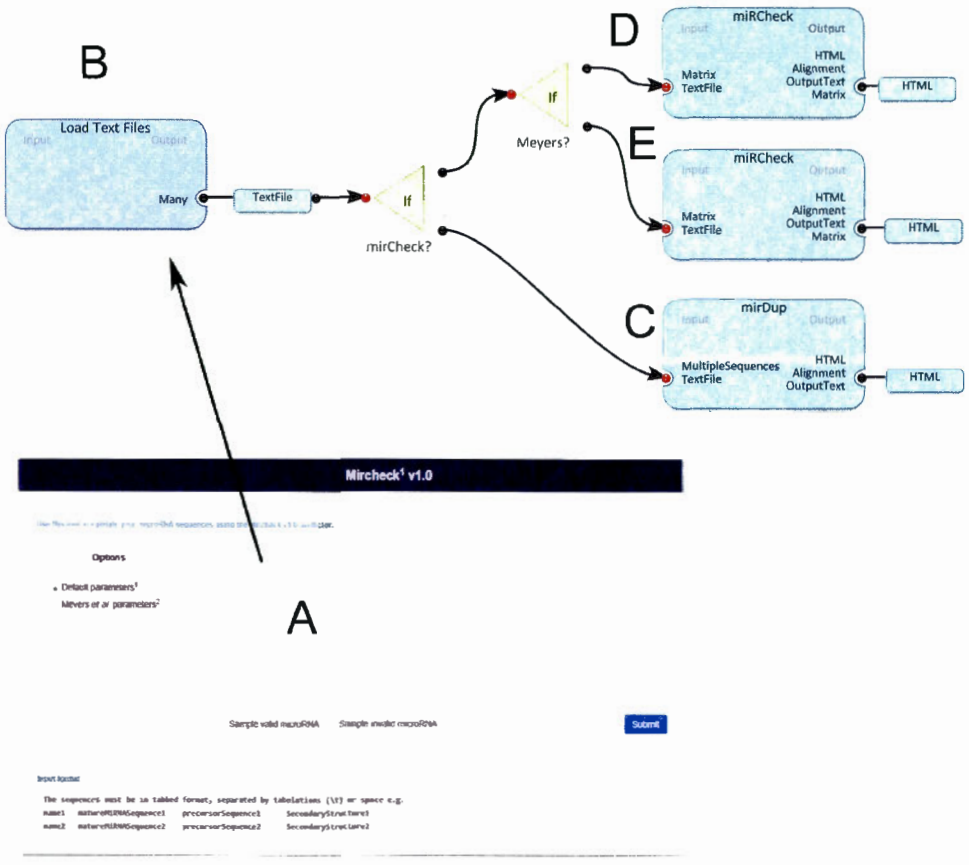


Figure 5.5 Flux de travaux du portail *Wheat micro-RNAs portal*. Ce flux de travaux utilise en entrée les données d'un formulaire Web[24] (panneau A) qui est chargé dans le flux de travaux via le composant *Load Text Files* (panneau B). Selon le contenu de celui-ci, reflétant les choix de l'utilisateur, les données vont être dirigées, par le flux de travaux de type *control-flow* (*If*), vers le logiciel mirDup (C; Leclercq *et al.*, 2013) ou encore vers le logiciel miRCheck avec soit ses paramètres par défaut (D; Jones-Rhoades et Bartel, 2004), soit les paramètres de Meyers *et al.* (2008) (panneau E). Dans le cas de la base de données de référence utilisée par mirDup, le logiciel gère directement ce paramètre à partir du fichier d'entrée.

[24] *p.ex.* http://wheat.bioinfo.uqam.ca/index.php?action=mircheck

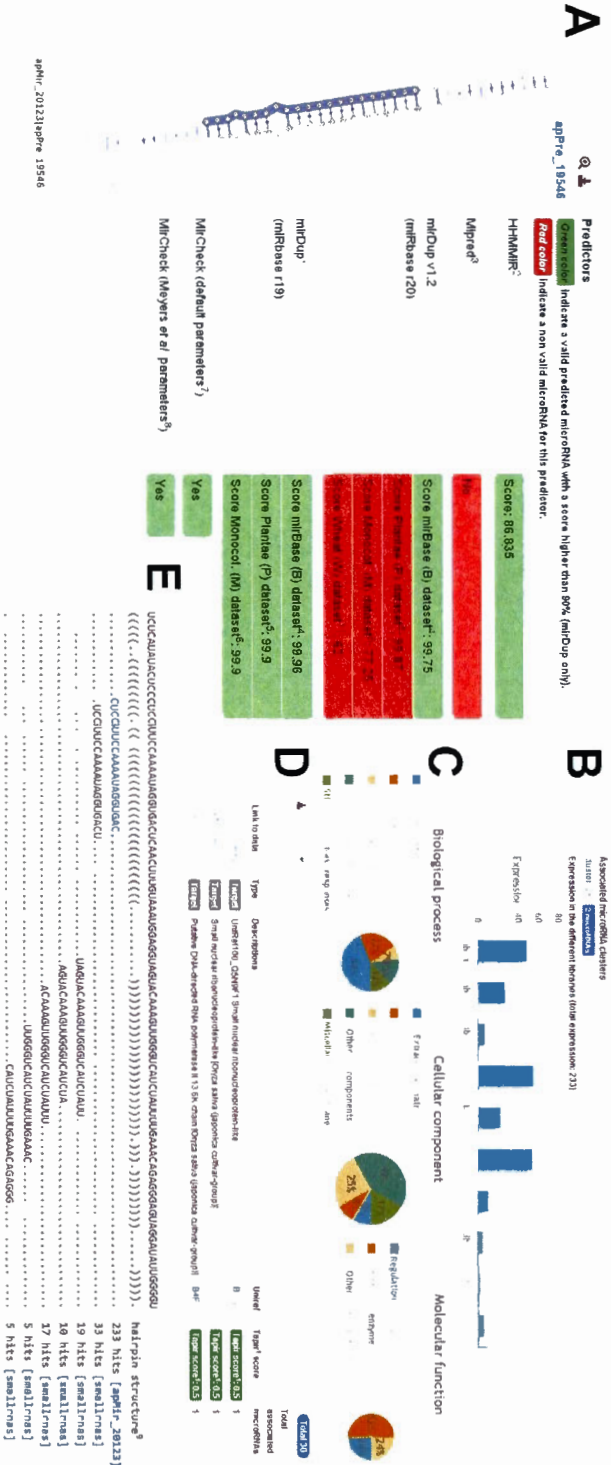


Figure 5.6 Vue d'ensemble des caractéristiques principales du portail Web. Celles ci incluent : A) structure bidimensionnelle (2D) en épingle à cheveux, ainsi que les scores de prédiction pour le *apMir 20123*, B) vue de l'expression de toutes les librairies et groupes de micro-ARNs associés (*clusters*), C) association d'ontologie de gènes (GO) pour un jeu de micro-ARNs ou une condition expérimentale, D) description des gènes cibles associés à chaque micro-ARN candidat, E) vue de l'expression et de la position calculée des micro-ARNs sur leurs pre-miRNAs.

5.3.5 Étude de cas

L'étude de cas suivante illustre la variété d'information accessible pour étudier les stress associés aux petits ARNs à partir de ce portail unique. Cette étude concerne aussi l'identification d'un lien entre l'exposition du blé à l'aluminium et l'expression du gène *wheat aluminum-induced* (*wali*) présent dans notre base de données. L'aluminium (Al) représente 7.5 % de la composition des sols et constitue une limite majeure à la production agricole (Snowden et Gardner, 1993). Dans des conditions acides, l'aluminium provoque l'inhibition rapide de la croissance des racines et réduit la disponibilité de l'eau, tout en limitant l'absorption des nutriments chez les plantes (Zeng *et al.,* 2012). La famille du gène *wali* comprend 7 gènes (*wali* 1, 2, 3, 4, 5) (Snowden et Gardner, 1993) ainsi que (*wali* 6, 7) (Richards *et al.,* 1994). Ceux-ci sont exprimés dans la racine du blé suite aux traitements à l'aluminium (Snowden et Gardner, 1993; Houde et Diallo, 2008), mais aussi après d'autres stress abiotiques (*wali1*, *wali5*) tels que l'exposition au cadmium, à la chaleur, au froid, à la déshydratation, à la salinité et le stress oxydatif produit par l'ajout de peroxyde (H_2O_2) (Snowden et Gardner, 1993; Garg et al 2012). Les gènes *wali1* et *wali5* ont également été proposés comme nouveaux candidats pour les études de réponse au stress chez les végétaux (Garg *et al.,* 2012).

Pour commencer l'étude, nous avons tout d'abord cherché le mot-clé « *wali* » à l'aide de l'option de base de recherche par mots-clés (onglet *Search*). Cette recherche a permis de trouver 6 séquences de gènes cibles associées, 1 séquence *EST* (voir Houde *et al.*, 2006 pour explications) et 33 résultats dans la section des associations par *Gene Ontology*. En utilisant le meilleur *score Tapir* comme critère, nous constatons que la cible similaire à *wali3* (Uniref Q43663) (*score Tapir* de 1) a été repérée dans nos bibliothèques d'expressions chez le blé et dans une moindre mesure les protéines associées *wali5*, *wali6* et *wali7* (*score Tapir* de 3). Bien que l'on dispose de plus d'informations pour le gène *wali1* (Snowden et Gardner, 1993), aucun lien vers cette protéine n'a été repéré dans notre base de données. En regardant l'ontologie des gènes cibles liés à notre recherche, nous constatons que *wali6* et *wali3* sont associés à 11 micro-ARNs liés à la description *composant cellulaire*. En cliquant sur le lien « *link to data* » pour chacune des séquences cibles associées, l'utilisateur peut avoir un aperçu

de leur expression dans les différentes bibliothèques. Il est intéressant de constater que les micro-ARNs associés à *wali3* ne montrent aucune expression en présence de l'aluminium (bibliothèques d'expression 8 et 10; voir Tableau 5.1), tout en affichant différents niveaux d'expression dans les autres conditions expérimentales. De plus, les mêmes micro-ARNs associés sont liés à *wali3* et *wali6* : apMir_19532, apMir_20346, apMir_21417, apMir_21655, apMir_39212, apMir_40495, apMir_46370. En cliquant sur un micro-ARN, l'utilisateur peut consulter la preuve expérimentale disponible pour ce micro-ARN ou micro-ARN candidat. Par exemple, en cliquant ou en recherchant le micro-ARN apMir_19532, on s'aperçoit que celui-ci est associé au pré-miRNAs présentant 17 épingles à cheveux. De plus, ce micro-ARN particulier est également associé à 3 groupes présentant des profils d'expression et de régulation similaires (groupes 34, 120, et 342).

Nous pouvons contrevalider ces résultats en consultant la vue présentant l'expression de micro-ARNs particuliers sous différentes conditions expérimentales. Pour ce faire, nous utilisons l'option de recherche avancée des micro-ARNs (*Advanced search*). Dans cette option, sélectionnons « *Not found microRNAs* » pour la condition expérimentale : « *Aluminium response in spring wheat* » et « *Aluminium response in winter wheat* ». L'exécution de cette requête va rendre accessible la liste des micro-ARNs n'étant pas exprimés dans ces deux conditions expérimentales, en plus de produire un résumé de l'ontologie de leurs gènes cibles. En regardant la cible de ces micro-ARNs non exprimés dans les deux conditions, nous pouvons trouver les protéines *wali3* et *wali6* dans les gènes cibles candidats.

Ainsi, selon notre ensemble de données et l'ensemble des preuves *in silico* disponibles dans le portail WMP, *wali3* et *wali6* pourraient être de bons candidats pour la régulation par des micro-ARNs sous la contrainte de l'aluminium chez le blé et pourraient être une cible intéressante pour une expérimentation *in vivo*.

5.3.6 Conclusions

Notre portail présente une nouvelle ressource unique pour l'analyse des micro-ARNs en offrant un accès mixte aux micro-ARNs candidats, mais aussi à l'expression des petits ARNs dans le contexte de conditions expérimentales chez le blé. Il fournit également un accès direct

à des outils bioinformatiques permettant la prédiction *de novo* micro-ARNs candidats chez le blé, mais aussi pour d'autres espèces de plantes. Dans l'avenir, nous souhaitons ajouter plus de bibliothèques préparées par notre équipe, portant sur les plants de blé cultivés sous d'autres conditions de stress abiotiques. De plus, nous mettons régulièrement à jour notre portail avec de nouveaux petits ARNs publiés chez le blé et d'autres céréales. En outre, nous prévoyons également enrichir la base de données en utilisant les jeux de données de petits ARNs actuellement disponibles dans la banque de données du *Gene Omnibus* (*GEO*) (Barrett *et al.*, 2007) du site de la *National Center for Biotechnology Information* (*NCBI*).

5.3.7 Perspectives de l'étude sur le blé

L'étude sur le blé nous a permis de voir l'ampleur des données générées par ce type d'étude. Après l'application de l'ensemble des logiciels et des méthodes, ~205 giga-octets de données ont été générés (Leclercq, 2012). Puisque ce genre d'étude requiert l'utilisation de plusieurs logiciels et méthodes interconnectées, l'utilisation de flux de travaux à travers la plate-forme *Armadillo* a permis de réaliser des essais sur des échantillons de données et à générer des résultats portant sur les gènes cibles. Cependant, la version 1.1 de la plate-forme, utilisée dans cette étude, n'incorporait pas de mécanismes de dispersion des tâches et des données sur des grappes d'ordinateurs distants. D'autres plates-formes bioinformatiques, telles que Galaxy (Giardine *et al.*, 2005) et Kepler (Altintas *et al.*, 2004) (voir section 2.6), implémentent ces mécanismes.

Toutefois, pour pouvoir exposer cet énorme ensemble de données, un site Web a été réalisé et présenté à la section 5.3. Dans la conception de ce site Web, un aspect modulaire a été implémenté sous la forme d'un flux de travaux réalisant l'analyse de micro-ARNs candidats par les logiciels mirDup (Leclercq *et al.*, 2013) et MirCheck (Jones-Rhoades et Bartel, 2004). L'utilisation de ce flux de travaux (Figure 5.5) permettra à court terme de conserver une trace des différentes exécutions. À plus long terme, l'utilisation de ce flux de travaux permettra d'ajouter rapidement au site Web de nouvelles fonctionnalités d'analyse, sans pour autant avoir à modifier des *scripts* qui peuvent contenir des centaines de lignes de code et qui, à la moindre erreur, ne sont pas exécutés (Turner et Lambert, 2014).

5.4 Étude de la pression sélective du VIH chez les femmes enceintes

Les femmes en âge de procréer représentent 50 % des individus infectés par le virus du VIH (virus d'immunodéficience de type 1) (World Health Organization, 2011). Une grande proportion de celles-ci auront au moins une grossesse au cours de leur vie. Conséquemment, la réponse immunitaire chez ces patientes va subir de profonds changements incluant une diminution de l'immunité à médiation cellulaire (Th1) et une augmentation de la réponse humorale (Th2) (Jamieson *et al.,* 2006). Ces changements entraînent alors une augmentation de la gravité des infections virales (Jamieson *et al.,* 2006).

Le but de cette étude était de vérifier s'il y avait une différence dans la pression sélective (*p.ex.* pression évolutive permettant une diversification des sites) s'exerçant sur l'enveloppe du virus (gp120) du VIH durant les différentes phases de la grossesse. Cette partie du virus est une cible intéressante pour la recherche puisque des anticorps pouvant lier les épitotes $CD4^+$ de cette protéine chez 90 % des souches du VIH de type 1, ont été identifiés et pourraient ainsi servir de point de départ pour des vaccins (Wu *et al.,* 2010). Du point de vue bioinformatique, cette pression sélective peut être évaluée via l'observation d'une plus grande présence de substitutions non-synonymes (*N*) par rapport aux substitutions synonymes (*S*; silencieuses) à l'intérieur d'un gène (Leal *et al.,* 2007). Pour ce faire, l'approche classique est l'analyse du ratio $\omega = dN/dS$ dans lequel un ratio positif (soit plus élevé que 1.0) indique la présence d'une pression évolutive diversifiante. Dans le cas où ce ratio est inférieur à 1.0, l'analyse suggère plutôt une pression négative (Nei et Gojobori, 1986). Un ratio ω proche de 1 suggère plutôt une évolution neutre (Kimura, 1984). Pour estimer le nombre réel de mutations (*c.-à-d.* substitutions) se produisant à un même site de la séquence, on utilise le modèle de Jukes-Cantor ou JTT dans le calcul de ce ratio (Zhang *et al.*, 2006). Cependant, cette approche ne permet par d'avoir directement un niveau de confiance associé aux différents sites (Zhang *et al.*, 2006). Une approche plus récente exploitant la phylogénie des espèces peut alors être utilisée. On utilise dans cette approche une méthode basée sur le maximum de vraisemblance pour évaluer la confiance de différents modèles de pressions sélectives par rapport aux séquences observées (Yang *et al.,* 2000). Cette approche est implémentée dans le logiciel *Phylogenetic Analysis by Maximum Likelihood* (PAML; Yang,

2007) qui fut utilisé pour vérifier différentes hypothèses de pressions sélectives sur le virus du VIH (Figure 5.7).

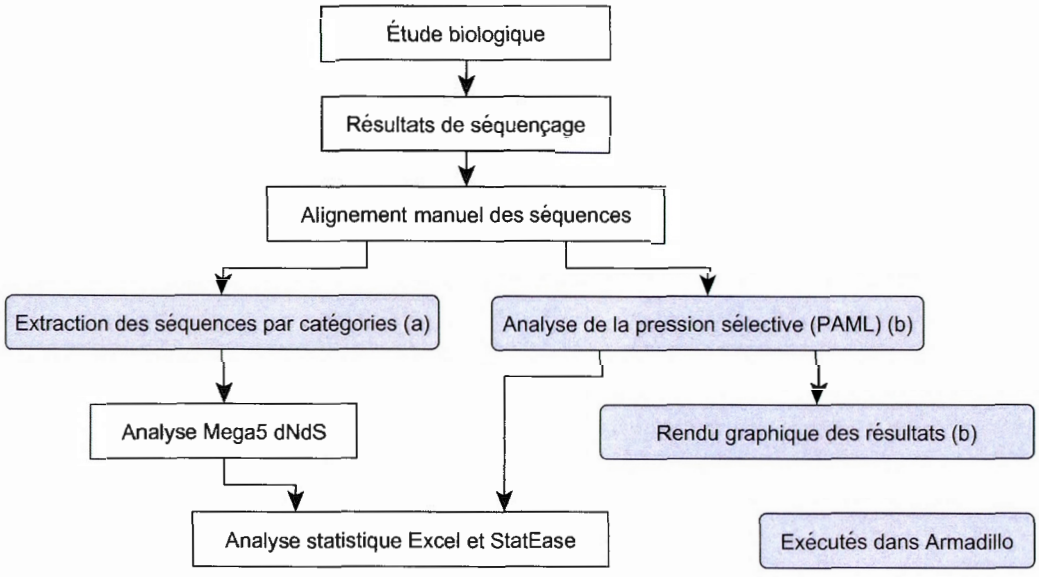


Figure 5.7 Vue d'ensemble de l'étude sur le VIH de type I chez les femmes enceintes (Ransy *et al.*, 2015). En gris, sections de l'étude ayant requis l'utilisation de flux de travaux exécutés à l'aide de la plate-forme *Armadillo*. Le flux de travaux d'extraction des séquences par catégories (a) est présenté à la Figure 5.9. Le flux de travaux permettant de réaliser l'analyse de la pression sélective par le logiciel PAML et le graphique des résultats (b) sont présentés à la Figure 5.10.

5.4.1 Description de l'étude

L'étude (Ransy *et al.*, 2004) comprenait 19 patientes enceintes (278 séquences au total) et 29 femmes contrôles (678 séquences au total). Les séquences composant l'échantillon de cette étude ont été soit téléchargées de la banque de données du *Los Alamos HIV Sequence Database*[25], ou encore prélevées durant trois trimestres de grossesse et séquencées à l'hôpital Ste-Justine (séquences avec les numéros d'accession KF038436-KF038566; KF038595-

[25] http://www.hiv.lanl.gov

KF039148; KF039166-KF039521; KF039537-KF039675 sur *GenBank*[26]). Les trois trimestres ont été désignés comme I (premier trimestre), J (deuxième trimestre) et K (troisième trimestre) lors de l'annotation de ces séquences. Un exemple de la relation évolutive entre les différentes séquences durant la grossesse est présenté à la Figure 5.8.

Lors de l'utilisation du logiciel PAML, six modèles (M0, M1, M2, M3, M7, M8) supposant des fréquences de substitutions et des hypothèses évolutives différentes ont été utilisés dans nos analyses (voir une définition des différents modèles dans Yang *et al.,* 2000). Les modèles d'hypothèse nulle M0, M1, et M7 ne permettaient pas de démontrer l'existence de sites sous pression sélective positive puisque le ratio ω dans ces modèles est compris entre 0 et 1. À l'inverse, les modèles M2, M3, et M8 (Yang *et al.,* 2005) permettaient d'estimer un ratio $\omega > 1$. Pour valider la présence d'une pression sélective positive, nous avons alors comparé l'hypothèse d'une sélection (modèles M2, M3 et M8) versus l'hypothèse nulle présentant une évolution neutre ou encore une pression négative (modèles M0, M1 et M7). Un second calcul du ratio ω aux différents sites des séquences du VIH en utilisant le logiciel MEGA version 5.0 (Tamura *et al.,* 2011) a alors servi à confirmer ces résultats à travers un test hypergéométrique.

Du point de vue bioinformatique (Figure 5.7), une grande portion de l'étude a été la conversion des données (Figure 5.7a et Figure 5.9), le lancement de logiciels d'alignement de séquences (Figure 5.10b), l'inférence phylogénétique (Figure 5.10c) et finalement le calcul de cette pression sélective à l'aide du logiciel PAML (Figure 5.10d) suivi de la présentation des résultats sous une forme graphique (voir par exemple les Figures 5.11 et 5.12). Ainsi plusieurs étapes de l'analyse ont nécessité la conception et l'utilisation de flux de travaux permettant la répétition des analyses sur cet ensemble de données. Nous présenterons dans la prochaine section deux flux de travaux conçus à cette fin.

---

[26] http://www.ncbi.nlm.nih.gov/nuccore/

5.4.2 Flux de travaux utilisés

5.4.2.1 Sélection de groupes de séquences à partir d'un flux de travaux

Nous avons d'abord utilisé un premier flux de travaux pour séparer les différents groupes d'échantillons et permettre leur analyse subséquente (Figure 5.9). Différents filtres (identifiés comme *GREP*, *Rename Sequence* et *Create Groups* dans la plate-forme *Armadillo*) permettant la sélection de séquences et de taxa présents dans des arbres phylogénétiques, sont inclus dans la plate-forme *Armadillo*. Le composant utilisé dans le cadre de cette étude (Menu *Conversion→Create Groups* de la boîte à outils dans *Armadillo*) permet de former des sous-groupes de séquences en cherchant des attributs dans le nom de ces séquences ou directement dans les séquences en utilisant des expressions régulières.

Dans cette étude, 956 séquences du VIH ont été collectées. Celles-ci ont été annotées à la main durant la phase de collecte et se retrouvaient dispersées dans plusieurs fichiers. Pour faire l'analyse selon les trimestres de gestation (voir la Figure 5.11) ou encore pour séparer les séquences des patientes enceintes, ou non (groupe contrôle) (voir la Figure 5.12) nous avons divisé les séquences. Dans un premier temps, nous avons chargé les séquences (définies comme le type *MultipleSequences*) dans la plate-forme *Armadillo*. Par la suite, nous avons appliqué le flux de travaux de la Figure 5.9. Une fois cette étape accomplie, nous avons poursuivi l'analyse par l'application du flux de travaux de la Figure 5.10 directement sur ces groupes de séquences. Notons que la version 1.1 de la plate-forme, utilisée dans cette étude, ne permet pas l'exécution de flux de travaux emboîtés.

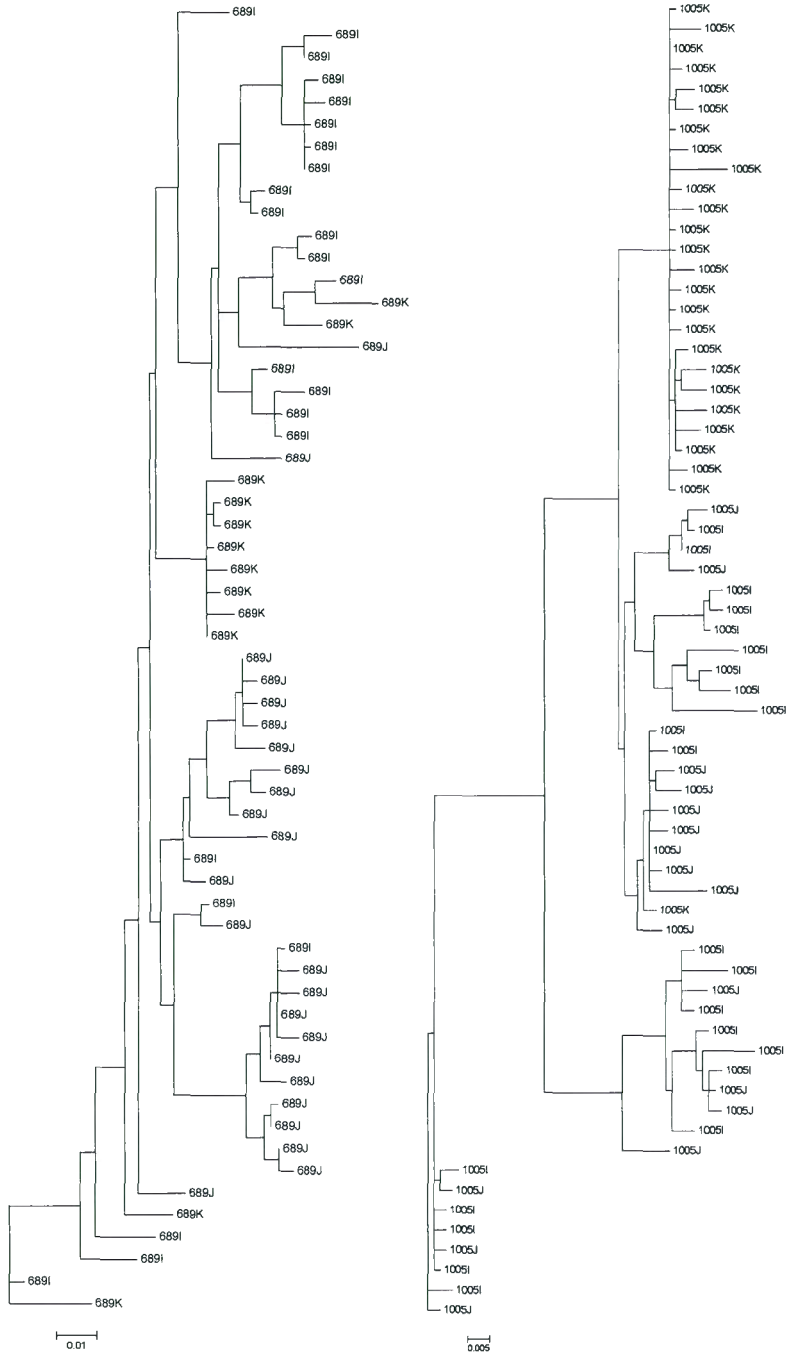


Figure 5.8 Exemples d'arbres phylogénétiques du gène *env* du VIH de deux patientes montrant l'évolution à différents stades de leur grossesse : I (premier trimestre), J (deuxième trimestre) et K (troisième trimestre). L'inférence phylogénétique par maximum de

vraisemblance a été réalisée à l'aide du logiciel PhyML 3.0 (Guindon et Gascuel, 2003) en utilisant le modèle d'évolution HKY et un nombre d'arbres dans le *bootstrap* égal à 100.

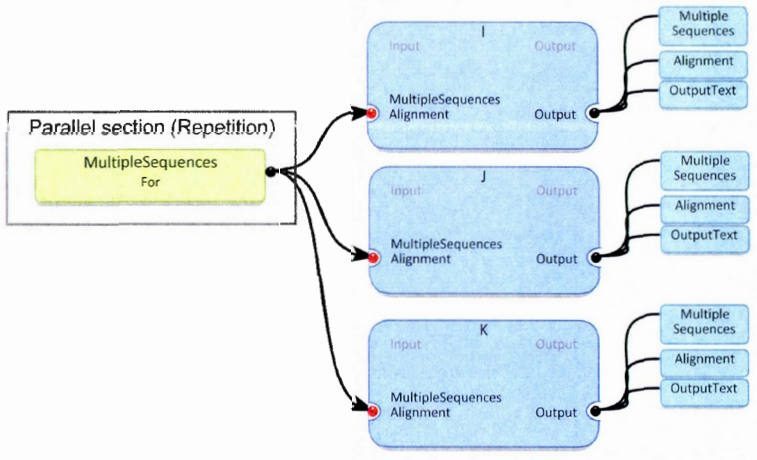


Figure 5.9 Flux de travaux ayant servi au filtrage et à la séparation des jeux de données avant alignement des séquences. Les composants du flux de travaux identifiés comme *I*, *J* et *K* dans la figure sont des filtres (composant *Create Groups* de la plate-forme *Armadillo*) permettant la sélection et la création de nouveaux groupes de séquences.

5.4.2.2 Analyse de la pression sélective positive à l'aide d'un flux de travaux

Dans une deuxième étape, l'analyse de jeux de séquences a été réalisée en trois étapes (Figure 5.10) : 1) alignement des séquences à un génome de référence du VIH (séquence *HXB2* provenant de *GenBank*[27]) et traduction en séquence protéique pour un meilleur alignement, 2) inférence de l'arbre phylogénétique des séquences et 3) analyse de la pression sélective par différents modèles. L'alignement des séquences au génome de référence (Figure 5.10b) a été réalisé avec le logiciel Muscle (Edgar, 2004). Pour réduire le bruit associé au nombre élevé de mutations dans ces séquences (Louwagie *et al.*, 1993), nous avons utilisé dans cette étape les séquences traduites en séquences protéiques. Par la suite, nous avons généré les arbres

phylogénétiques (Figure 5.10c) correspondants aux séquences protéiques retransformées en séquences nucléotidiques. Le logiciel PhyML v3.0 (Guindon *et al.,* 2010) a été utilisé avec le modèle d'évolution HKY et 100 réplicas dans le *bootstrap*. Finalement, l'évaluation de la pression sélective par le logiciel PAML (Yang, 2007) version 4.4c (Figure 5.10d) a été réalisée. Dans cette étude, nous avons utilisé l'application CODEML de PAML. Cependant, les applications yn00 et BASEML sont aussi incluses dans la plate-forme *Armadillo*. Ces étapes qui sont coûteuses en temps de calcul, avec nos jeux de données comprenant 77 jeux de séquences, ont été parallélisées pour réduire le temps d'exécution en exécutant en parallèle le flux de travaux (Figure 5.10a). Cette parallélisation du flux de travaux en utilisant différents *threads* est incluse par défaut dans la version 2.0 de la plate-forme *Armadillo*, mais n'était pas incluse dans la première version utilisée dans cette étude. Ainsi, les flux de travaux ont simplement été exécutés sur différents ordinateurs. L'architecture de *Armadillo* facilite cette distribution en sauvegardant toutes les données dans un seul fichier projet (voir chapitre 3), qui peut alors être partagé sur différents ordinateurs en installant *Armadillo* dans un répertoire partagé (par exemple en utilisant Dropbox[28]). Notons aussi que, suite à l'exécution de ces flux de travaux, une vérification manuelle des alignements de séquences et des arbres phylogénétiques inférés a été effectuée pour les valider, ou ajuster les paramètres d'exécution, si requis. De plus, nous avons séparé la génération de la vue graphique des résultats dans la version actuelle (*non montré*) en utilisant les fonctions graphiques de *Armadillo* (version *2.0*; Figures 5.11 et 5.12 présentées dans l'étude de Ransy *et al.,* 2015).

---

[27] http://www.ncbi.nlm.nih.gov/nuccore/1906382
[28] http://www.dropbox.com

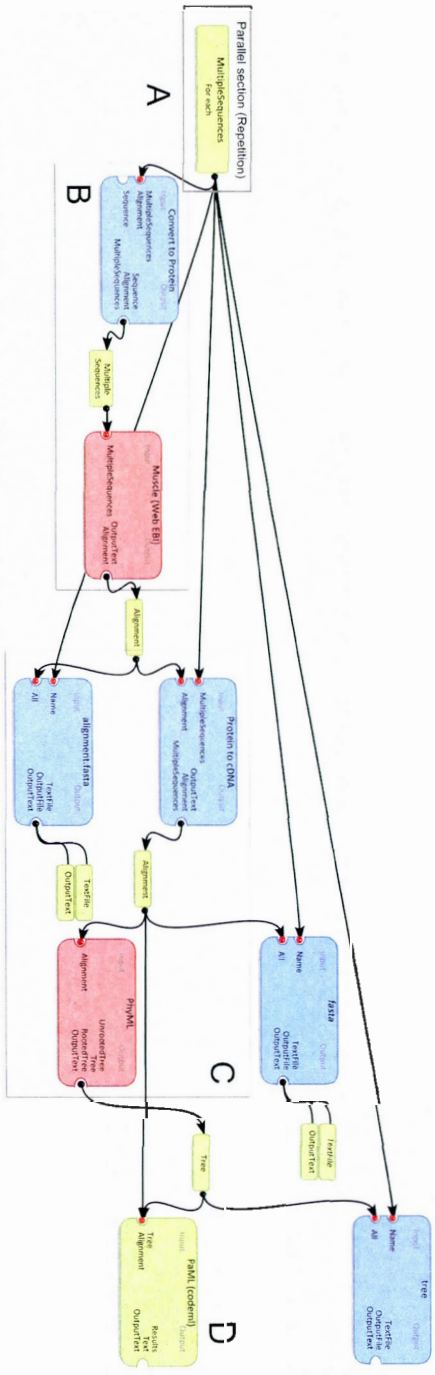


Figure 5.10 Flux de travaux permettant l'évaluation de la pression sélective. Le panneau (A) présente une répétition de jeu de données analysée en parallèle. Le panneau (B)

présente l'alignement des séquences, à l'aide du logiciel Muscle (Edgar, 2004), du gène *env* préalablement traduit en séquences protéiques. Le panneau (C) présente l'inférence de l'arbre phylogénétique à partir d'une méthode de maximum de vraisemblance (dans cet exemple on utilise PhyML) qui sera utilisée en entrée du logiciel PAML, présenté au panneau (D).

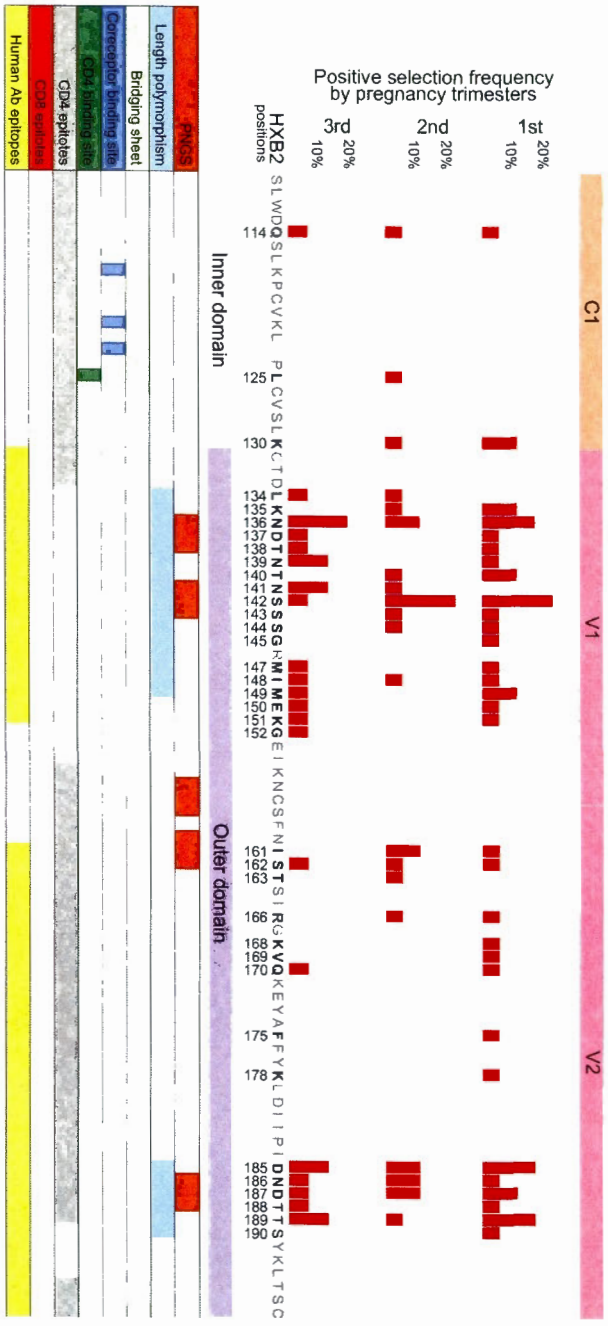


Figure 5.11 Exemples de sites positivement sélectionnés entre les patientes enceintes, suivant l'évolution des trimestres pour la protéine gp120 (gène *env*). La figure complète est présentée dans Ransy *et al.* (2015). Ces résultats ont été obtenus après exécution du flux de travaux présenté à la Figure 5.10 et filtrage des alignements de séquences composant les

différents trimestres dans le flux de travaux présenté à la Figure 5.9. La partie représentée englobe les segments encodant les sous-régions C1, V1, V2, C2, V3 et C3. Les AA représentés (sites 110-348) font référence au génome de référence du VIH de type 1 HXB2[29]. Les caractéristiques de cette partie incluent : les régions variables (*en rose*), les régions constantes (*en orange clair*), les domaines internes (*mauve clair*) et externes (*mauve foncé*) de la protéine, les sites de N-glycosylation potentiels (PNGS; *orange*), les régions polymorphiques V1 et V2 (*bleu pâle*), le feuillet beta d'ancrage (*vert pâle*), les sites de liaison de co-récepteurs (bleu foncé), les sites de liaison des épitopes humains (*jaune*) et des cellules T, $CD4^+$ (*vert foncé*) et $CD8^+$ (*rouge*). La région V3 est hyper-variable.

[29] http://www.ncbi.nlm.nih.gov/nuccore/1906382

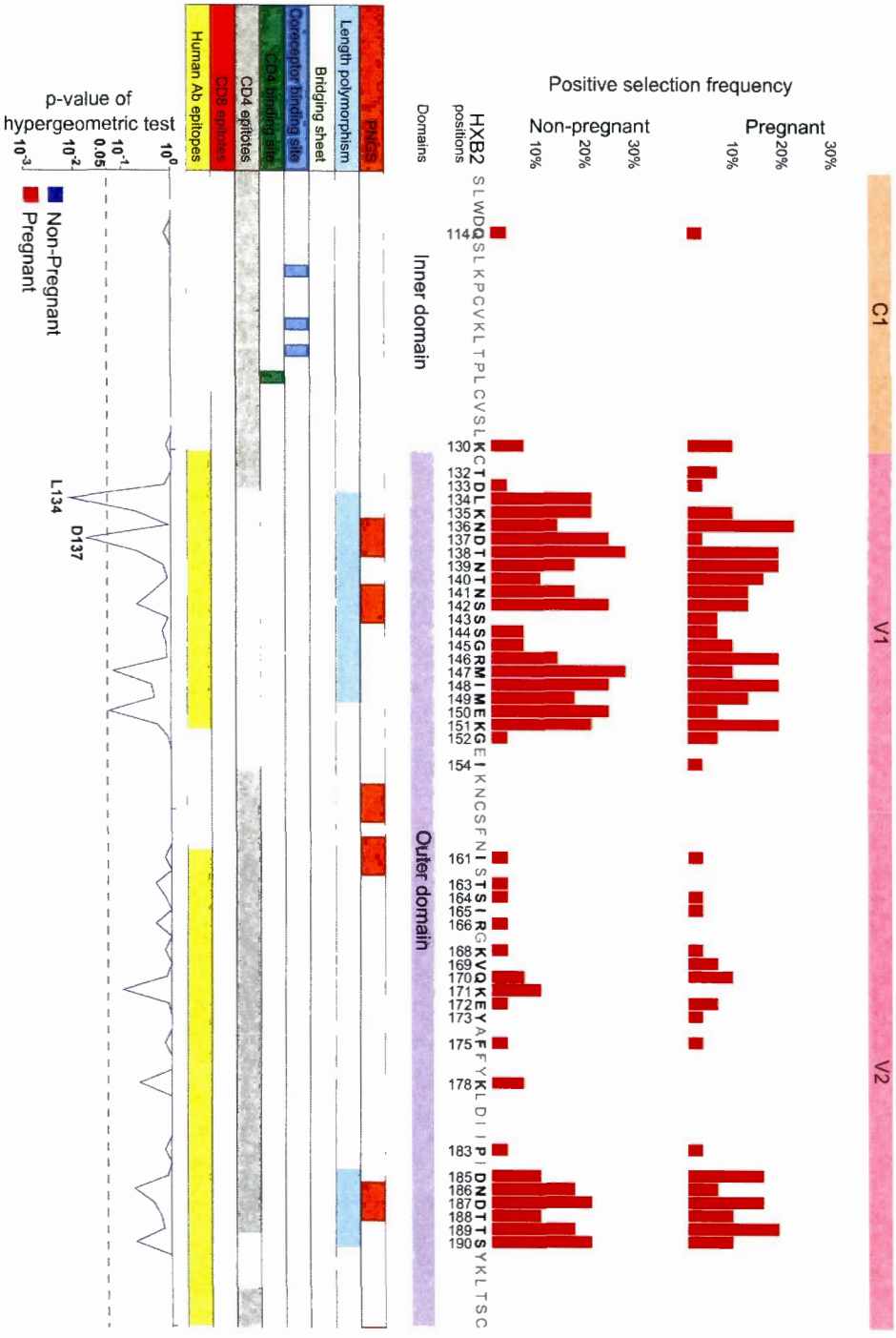


Figure 5.12 Exemples de sites positivement sélectionnés entre les patientes enceintes ou non pour la protéine gp120 (gène *env*). La figure complète est présentée dans Ransy *et al.* (2015). Ces résultats ont été obtenus après exécution du flux de travaux présenté à la Figure

5.10 et filtrage des alignements de séquences pour sélectionner les patientes enceintes ou non. La partie représentée englobe les segments encodant les sous-régions C1, V1, V2, C2, V3 et C3. Les AA représentés (sites 110-348) font référence au génome de référence du VIH de type 1 HXB2. Les caractéristiques de cette partie incluent : les régions variables (*en rose*), les régions constantes (*en orange clair*), les domaines internes (*mauve clair*) et externes (*mauve foncé*) de la protéine, les sites de *N*-glycosylation potentiel (PNGS; *orange*), les régions polymorphiques V1 et V2 (*bleu pâle*), le feuillet beta d'ancrage (*vert pâle*), les sites de liaison de co-récepteurs (bleu foncé), les sites de liaison des épitopes humains (*jaune*) et des cellules T $CD4^+$ (*vert foncé*) et $CD8^+$ (*rouge*). La région V3 est hyper-variable.

5.4.3 Résultats et conclusions de l'étude

L'analyse de la pression sélective chez les femmes enceintes a montré une diminution du nombre de sites où s'exerçaient celle-ci tout au long de la grossesse (diminution de 1.61 site par trimestres) (Ransy *et al.*, 2015). Ainsi, 39 sites de la protéine gp120 ont été identifiés statistiquement comme étant sujets à une pression sélective. La grande majorité de ceux-ci se trouvaient dans des sites identifiés comme responsables de l'attachement d'anticorps associés à la neutralisation non-spécifique (*cross reacting-antibodies*). Cependant, seulement quatre sites entre les patientes enceintes ou non ont été identifiés statistiquement par un test hypergéométrique ($p<0.05$; Trame de la Figure 5.12) comme étant réellement sous pression sélective : L134, D137 (sites présentés à la Figures 5.12), R308, et S347. En définitive, il y a effectivement des sites propres à une certaine pression sélective sur la protéine gp120 chez les femmes enceintes durant la grossesse. Cependant, il est prématuré de les associer à des causes spécifiques (Ransy *et al.*, 2015). Du côté bioinformatique, la conception et la modélisation de l'analyse ont été compliquées par l'utilisation de formats de fichiers non standardisés par le logiciel PAML (Yang, 2007). De plus, dans la partie B du flux de travaux (Figure 5.10b), nous avons dû enlever les sites contenant des codons stop en les remplaçant par des gaps. Ces codons stop ne pouvaient être pris en charge par le logiciel PAML. Dans le cas du virus de VIH, évoluant rapidement, plusieurs de ces sites étaient présents dans nos jeux de données (voir Louwagie *et al.*, 1993). Puisque la cohorte de femmes enceintes était assez petite et l'analyse était limitée à cette section du virus du VIH de type 1, une plus grande étude portant sur la pression sélective sur la totalité du génome et sur une plus grande

cohorte est présentement en cours en utilisant le même flux de travaux. On voit ainsi l'avantage de l'utilisation des flux de travaux dans les analyses phylogénomiques, lorsque vient le temps de reproduire l'expérimentation *in silico* sur de nouveaux jeux de données. Pour finaliser l'automatisation du protocole, l'ajout d'un module de calcul de ratio dN/dS (Zhang *et al.*, 2006), réalisé en utilisant le logiciel *MEGA* (Tamura *et al.*, 2011) ainsi que l'ajout de routines en langage *R*[30] (langage gratuit dédié aux calculs mathématiques) pour calculer la probabilité hyper-géométrique, s'avéreraient nécessaires.

5.5 Conclusions

Dans ce chapitre, nous avons présenté deux études touchant le domaine de la phylogénomique, ayant été réalisées en utilisant des flux de travaux exécutés sur la plate-forme *Armadillo*. Pour l'instant, seulement certaines parties des études complètes (Figure 5.1 et 5.7) ont pu être réalisées en utilisant des flux de travaux. Ceci est comparable à ce qui a été observé dans d'autres études par Goderis (2008). La difficulté principale rencontrée dans notre cas pour la réalisation complète de l'expérimentation *in silico* des deux études est le nombre insuffisant d'outils disponibles dans la plate-forme au moment de leur réalisation. Par exemple, nous sommes présentement à inclure dans la plate-forme des outils propres à la génomique et les nouvelles technologies de séquençage à haut débit (*p.ex.* suite de logiciels MAQ, Bowtie 2 et BWA; revus par Bao *et al.*, 2011) permettant la réalisation de telles études sans que l'utilisateur n'ait à « inclure » celles-ci manuellement tel que dans le travail de Leclercq (2012; section 5.2.2.1). De plus, certaines étapes subséquentes à la réalisation des études, telles que la présentation des données sous la forme de graphiques, ne sont pas encore incluses dans la plate-forme *Armadillo*. Néanmoins, puisque la plate-forme permet l'exécution de *scripts* en *R* avec des sorties graphiques, de même que l'ajout de logiciels par l'utilisateur, ces difficultés pourraient être résolues.

Finalement, nous avons aussi présenté un article décrivant l'utilisation de flux de travaux pour fournir des services computationnels via le portail Web *WMP* (Section 5.3 et Figure

[30] http://www.r-project.org/

5.5). Dans ce cas, les flux de travaux sont dissimulés à l'utilisateur qui interagit directement avec la plate-forme via un formulaire Web standard (Figure 5.5).

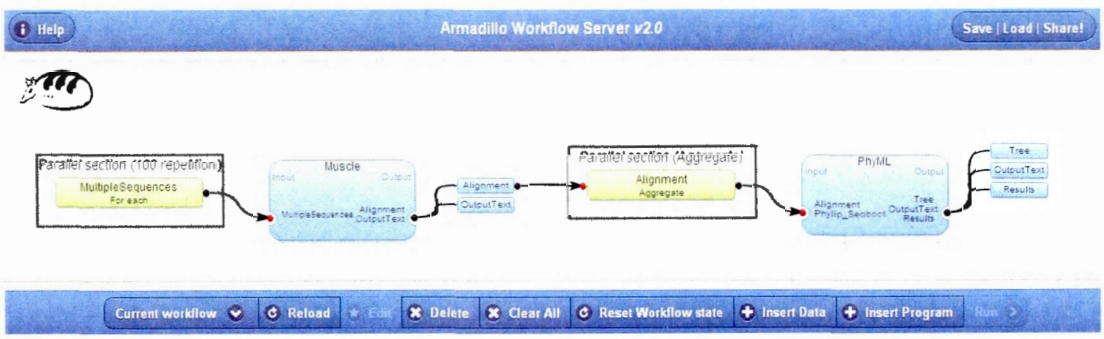


Figure 5.13 Vue de l'interface Web de la plate-forme *Armadillo* version 2.0[31].

Cependant, puisque la programmation de la plate-forme *Armadillo* a été réalisée en langage de programmation *Java* avec la librairie processing[32] (Fry, 2004), elle pourrait être convertie en application Web (*en développement*; Figure 5.13). Une telle plate-forme Web pourra, sans avoir recours à une installation locale, permettre aux scientifiques de partager des données et des protocoles de recherche de manière similaire à la plate-forme Galaxy, en restant orientée vers la phylogénétique et la phylogénomique. Ainsi, les utilisateurs d'un tel portail Web auraient la possibilité de produire eux-mêmes leurs protocoles d'analyses personnalisés.

---

[31] http://www.trex.uqam.ca/~armadillo_workflow/ (Lord *et al. En préparation, version alpha*)
[32] http://processing.org/

CHAPITRE VI

CONCLUSIONS ET PERSPECTIVES

Les flux de travaux en sciences deviennent des techniques de plus en plus exploitées (De Oliveira *et al.*, 2013; Oakley *et al.*, 2014). En sciences de la vie, plusieurs études font mention de plateformes de gestion de flux de travaux Galaxy (Giardine *et al.*, 2005) et Taverna (Oinn *et al.*, 2007). Mais ces plateformes généralistes, n'ont pas un niveau conceptuel précis associé à un domaine comme la phylogénétique (*p.ex.* usage de type de données liées à ce domaine). Cependant, ils apportent divers avantages reliés aux propriétés de réutilisation, de facilité d'utilisation, d'adéquation aux grandes données, de parallélisation et de l'encapsulation et la conceptualisation de méthodes complexes. Dans cette thèse, nous avons modélisé et conçu une plate-forme exploitant la technique des flux de travaux adaptée à un domaine complexe de la bioinformatique (l'analyse phylogénétique). Cette plate-forme de flux de travaux, nommée *Armadillo*, permet la réalisation d'une centaine de procédures distinctes pour des études phylogénétiques et phylogénomiques complexes en un simple clic. Cela est possible grâce à l'automatisation des tâches, l'accès à des bibliothèques d'opérations et des techniques adéquates de visualisation. Dans cette thèse, nous avons mis l'emphase sur les types de flux de travaux et leurs rôles dans la recherche en phylogénomique et en phylogénétique.

Avec le niveau de conceptualisation associé au domaine de la phylogénétique, la plateforme *Armadillo* est adaptée au travail des biologistes qui n'ont aucune base en programmation. Cette plate-forme comporte des types de données définis comme : *Alignement, Sequence, Arbre,* qui facilitent l'intégration des opérations sans se soucier du détail d'interconnexion, des formats des entrées et des sorties et de la gestion des fichiers intermédiaires. Actuellement, plusieurs centaines de flux de travaux décrivant des procédures

phylogénétiques décrites dans la littérature sont implantés. Cependant, de nouvelles fonctionnalités et flux de travaux peuvent êtres ajoutés par les utilisateurs. .

L'instanciation de procédures phylogénétiques (avec des choix de paramètres, de paradigmes et d'algorithmes) pour une problématique donnée entraine souvent une multiplication des flux de travaux associée à cette problématique (voir le chapitre 3). Ainsi, il est souvent important de pouvoir grouper, ordonnancer et/ou sélectionner des flux répondant à un critère défini. Pour y arriver, il est important de disposer de méthode de comparaison de flux de travaux. Par conséquent, le chapitre 4 présente des algorithmes pouvant classifier des flux de travaux, sans *a priori*, et en permettant: 1) un regroupement par mots-clés, 2) un regroupement selon un contexte d'exécution. De plus, nous avons présenté une nouvelle mesure de support qui permet une meilleure compréhension de la classification de différents éléments lors de l'utilisation d'algorithmes de partitionnement. Ce faisant, nous avons validé l'utilisation de la distance cosine et l'utilisation de la méthode *k*-medoids pour le regroupement des flux de travaux par des méthodes de partitionnement. Ainsi, ces approches permettraient d'analyser et classer tous les flux de travaux répertoriés dans la littérature, et de pouvoir en extraire des connaissances associées à ces types de problématiques. Ces connaissances peuvent être une source importante dans la construction de système de recommandation.

Au chapitre 5, nous avons présenté des cas réels de recherches ayant profités de l'utilisation de flux de travaux et de la plate-forme *Armadillo*. Le premier cas concerne l'utilisation de la plate-forme de manière incrémentale en vue d'inclure d'autres méthodes bioinformatiques telles que l'analyse de structure secondaire d'ARN, la prédiction de micro-ARNs, le traitement de données issues du séquençage de nouvelle génération, etc. Ainsi, nous avons conçu une plateforme sur mesure intégrant les éléments précédents en vue de prédire des micro-ARNs associés à la réponse à plusieurs stress abiotiques chez le blé. La seconde application concerne la conception de flux de travaux en vue de l'analyse de pression sélective associée à des protéines de l'enveloppe virale du VIH collectées chez plusieurs femmes enceintes. Ces applications démontrent les larges perspectives de l'utilisation de la plate-forme.

6.1 Principales contributions

- Analyse, conception et implémentation d'une plate-forme de flux de travaux qui permet les études phylogénétiques et phylogénomiques sous plusieurs environnements (Linux, Mac et Windows).

- Modélisation des concepts phylogénétiques en structures et procédures en vue de leur intégration dans un flux de travaux comme types de données définis.

- Proposition et test par simulations de quatre types d'encodage de flux de travaux permettant le regroupement en fonction d'un contexte d'exécution ou par la sélection d'éléments présents dans ceux-ci

- Création d'un nouveau critère de support pour mesurer l'appartenance d'un flux de travaux à une classe donnée. De même, nous avons proposé un critère de support global d'une solution de partitionnement.

- Création de deux jeux de données permettant l'étude des similarités entre flux de travaux dans un contexte d'exécution (présentés au chapitre 4 et en Annexe B).

- Conception et implantation d'une base de données contenant de nouveaux micro-ARNs du blé obtenus à partir de différents flux de travaux intégrés dans la plate-forme *Armadillo*.

- Étude de la pression sélective sur la protéine de l'enveloppe du VIH de type I chez les femmes enceintes à partir de notre flux de travaux.

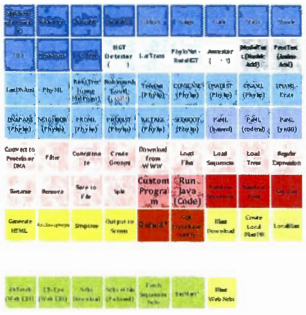
6.2 Perspectives

Dans cette thèse, nous avons décrit une plate-forme de flux de travaux locale, permettant la réalisation d'études phylogénétiques et phylogénomiques. La plateforme en version locale atteint rapidement ses limites lorsqu'il faut manipuler de larges données. Ainsi, il serait important de proposer des dispositifs exploitant des serveurs de calculs distants, incluant les ressources de l'infonuagique (*cloud*). Une version permettant l'utilisation de la plate-forme *Armadillo* directement sur le Web est en cours de développement et devrait permettre ce

genre de déploiement en utilisant une logique de distribution des tâches Min-Max ou encore celle basée sur le principe du chemin critique dynamique (DCP-G; Rahman *et al.*, 2013) (voir la section 5.5). Par ailleurs, il serait important de suivre à la trace l'utilisation et la provenance des données (De Oliveira *et al.*, 2013) et avoir une plus grande visualisation de l'historique des exécutions. De plus, l'exécution de grands jeux de données peut mener à une multitude de flux de travaux (Ramakrishnan et Gannon, 2008). Pour répondre à ces exigences, il serait important d'intégrer dans la plateforme *Armadillo* la classification des flux de travaux développée dans un contexte d'exécution.

Finalement, d'autres domaines scientifiques, pris avec une augmentation dramatique du volume des données, commencent à s'intéresser aux flux de travaux. Ainsi, une étude de Turner et Lambert (2014) fait état de l'usage de flux de travaux dans le domaine des sciences sociales. Plus de 250 flux de travaux se retrouvant sur le portail *myExperiment* portent sur le logiciel *RapidMiner*[33], une plate-forme d'intelligence d'affaires construite autour de l'utilisation de flux de travaux, permettant la répétition d'analyses dans les domaines de l'assurance, des communications, etc. Il serait envisageable d'étendre le « *langage applicatif*» de la plate-forme *Armadillo* (Figure 6.1) pour prendre en compte d'autres contextes d'exécution. Une avenue serait d'y rajouter les concepts associées aux utilitaires systèmes comme la ligne de commande Linux. Il serait alors possible d'effectuer des opérations de surcharge sur les classes de base du *modèle objet* sur lequel repose la plate-forme pour intégrer ces nouvelles fonctionnalités. La conception évolutive de cette plate-forme permet ainsi une expansion facile de son utilisation à d'autres domaines.

[33] http://rapidminer.com/



Phylogénétique

Une sélection de commandes Linux

cd	ls	rm	cp		mv	chown	chmod	chgrp
pwd	mkdir	rmdir	scp	rcp	rmv	su / sudo	date	times
sleep	uniq	nice	slocate	locate	df	bzip2	tar	gzip
man	help	read	paste	find	du	sort	tsort	tree
top	ps	bg	cron	crontab	time	env	export	kill
let	declare	set	unset	uuencode	uudecode	alias	adduser	addgroup
printf	clear	echo	exec	true	false	wget	awk/ gawk	sed
command	function	if	until	eval	more	wc	grep/ egrep	head
case	for	while	continue	break	less	tail	cat	cut

Figure 6.1 Modification du « *langage applicatif* » de la plate-forme *Armadillo* pour un autre domaine.

ANNEXE A

AUTRES FLUX DE TRAVAUX

A.1 Introduction

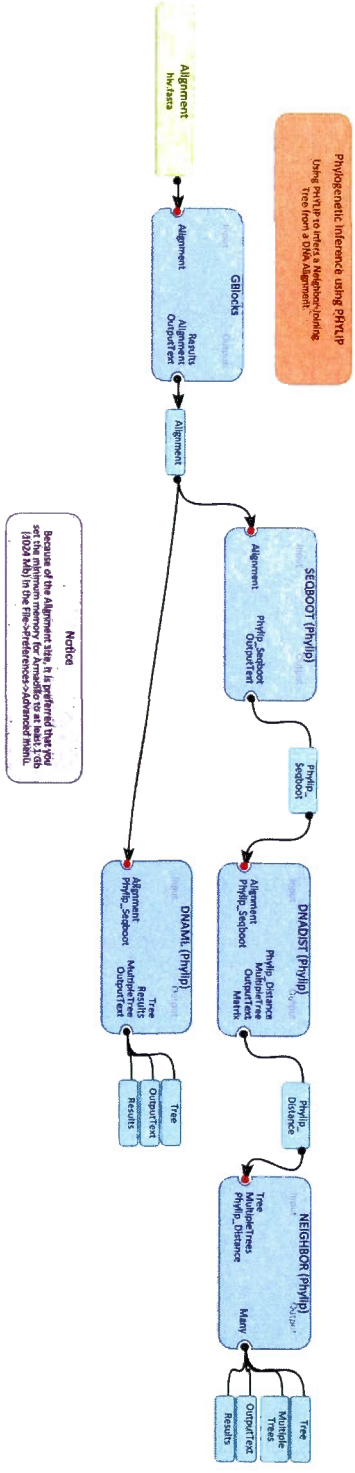
Cette thèse ne serait pas complète sans la présentation de flux de travaux incorporés comme exemples dans la plate-forme *Armadillo* v*1.1*, actuellement disponible sur le Web[34]. Dans cette section, nous donnons un aperçu de ceux-ci comme démonstration des méthodes et structures incluses dans la plate-forme (Tableau C1). Ces flux de travaux présentent ainsi certaines des caractéristiques de la plate-forme qui n'ont pu être mises de l'avant dans les sections précédentes. De plus, on peut ainsi voir la diversité des flux de travaux pouvant être créés et exécutés sur notre plate-forme.

Tableau A.1 Sélection de flux de travaux inclus dans la plate-forme *Armadillo*

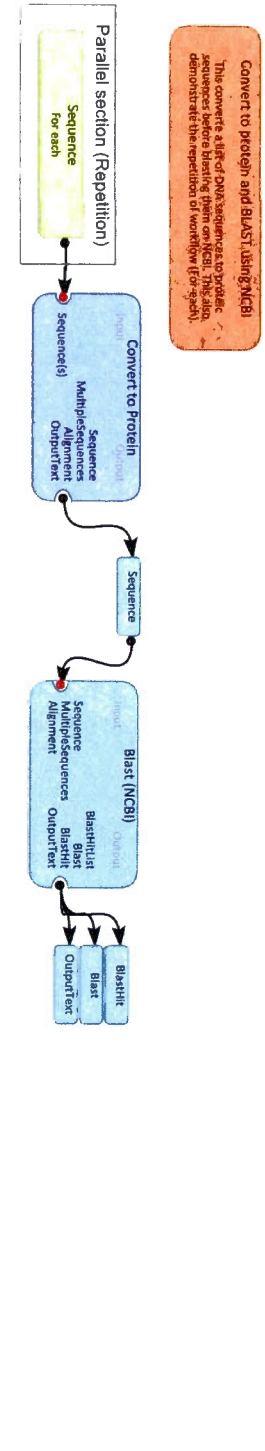
Flux de travaux	**Caractéristiques particulières**
A	Méthode d'inférence phylogénétique par maximum de parcimonie et méthode de distance.
B	Exécution parallèle de BLAST (*version 2.0*).
C	Recherche conditionnelle (*If*) de l'ontologie des gènes (Utilisé dans l'étude de des micro-ARNs du blé, *version 2.0*).
D	Comparaison d'arbres par la méthode de Robinson et Foulds.
E	Méthode d'alignement de séquences Muscle et ClustalW.
F	Flux de travaux génomique (*version 2.0*)

[34] http://adn.bioinfo.uqam.ca/armadillo

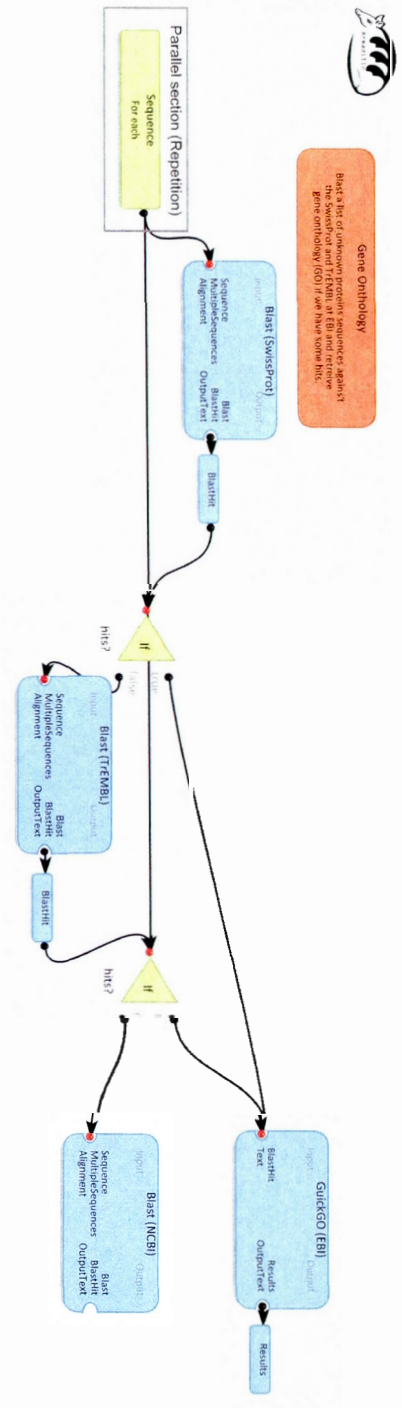
A)

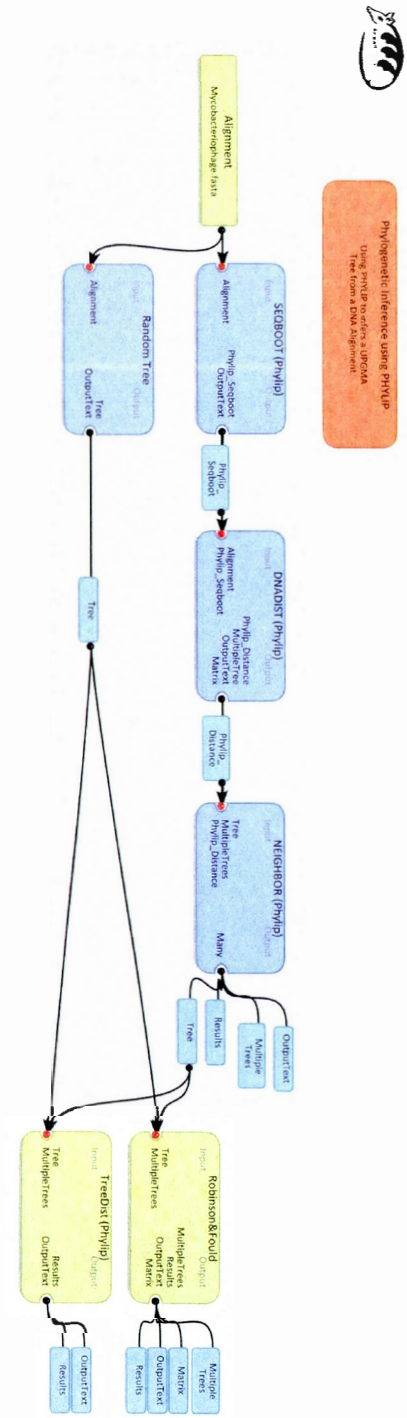

Phylogenetic Inference using PHYLIP
Using PHYLIP to infers a Neighbor-Joining Tree from a DNA Alignment.
Alignment
hiv.fasta
GBlocks
Alignment
Results
Alignment
OutputText
Alignment
SEQBOOT (Phylip)
Alignment
Phylip_Seqboot
OutputText
Phylip_ Seqboot
DNADIST (Phylip)
Alignment
Phylip_Seqboot
Phylip_Distance
MultipleSequence Alignment
OutputText
Matrix
Phylip_ Distance
DNAML (Phylip)
Alignment
Phylip_Seqboot
Tree
Results
MultipleTree
OutputText
Tree
OutputText
Results
NEIGHBOR (Phylip)
Tree
MultipleTrees
Phylip_Distance
Many
Tree
Multiple Trees
OutputText
Results
Notice
Because of the Alignment size, it is preferred that you set the maximum memory allocation to at least 1 GB (1024 MB) in the File->Preferences->Advanced menu.

B)

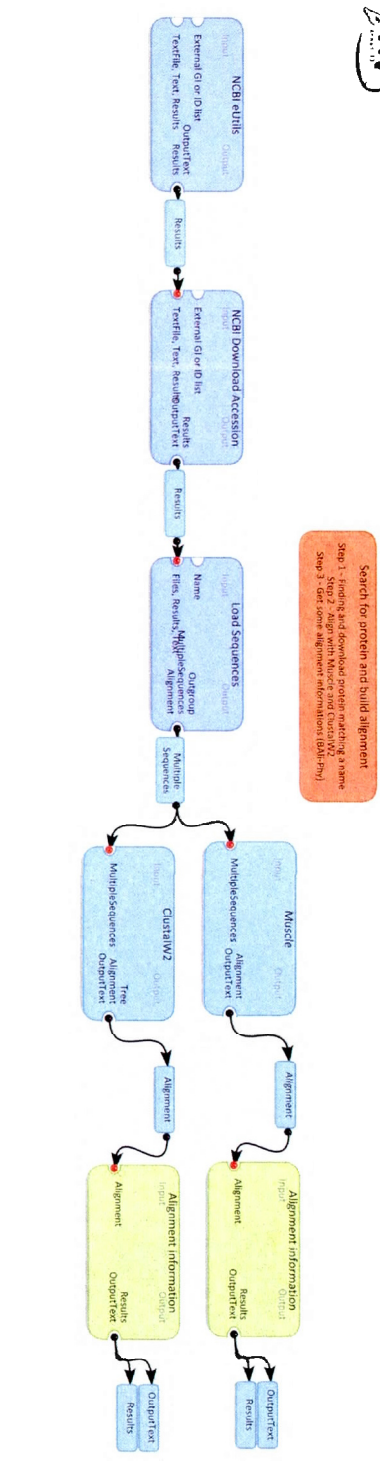

Convert to protein and BLAST using (NCBI)
This converts a list of DNA sequences to protein sequences before blasting them on NCBI. This also demonstrates the repetition of workflow (For each).
Parallel section (Repetition)
Sequence
For each
Convert to Protein
Sequence(s)
Sequence
MultipleSequences Alignment
OutputText
Sequence
Blast (NCBI)
Sequence
MultipleSequences Alignment
BlastHitList
Blast
BlastHit
OutputText
BlastHit
Blast
OutputText

C)


Gene Ontology
Blast a list of unknown proteins sequences against the SwissProt and TrEMBL at EBI and retrieve gene ontology (GO) if we have some hits.
Parallel section (Repetition)
Sequence
For each
Blast (SwissProt)
Sequence
MultipleSequences
Alignment
Blast
BlastHit
OutputText
BlastHit
If
hits?
false
Blast (TrEMBL)
Sequence
MultipleSequences
Alignment
Blast
BlastHit
OutputText
BlastHit
If
hits?
false
Blast (NCBI)
Sequence
MultipleSequences
Alignment
Blast
BlastHit
OutputText
QuickGO (EBI)
BlastHit
Text
Results
OutputText
Results


Phylogenetic inference using PHYLIP
Using mmbPipeFarm's UPGMA
Tree from a DNA Alignment
Alignment
Mycobacteriophage.fasta
SEQBOOT (Phylip)
Alignment
Phylip_Seqboot
OutputText
Phylip_
Seqboot
DNADIST (Phylip)
Alignment
Phylip_Seqboot
Phylip_Distance
MultipleTrees
OutputText
Matrix
Phylip_
Distance
NEIGHBOR (Phylip)
Tree
MultipleTrees
Phylip_Distance
Many
Tree
Results
Multiple
Trees
OutputText
Random Tree
Alignment
Tree
OutputText
Tree
Robinson&Fould
Tree
MultipleTrees
MultipleTrees
Matrix
OutputText
Results
Multiple
Trees
Matrix
OutputText
Results
TreeDist (Phylip)
Tree
MultipleTrees
Results
OutputText
OutputText
Results

D)


Search for protein and build alignment
Step 1 - Finding and download protein matching a name
Step 2 - Align with Muscle and ClustalW2
Step 3 - Get some alignment informations (utili-Phy)
NCBI eUtils
External GI or ID list
TextFile Text Results
OutputText
Results
NCBI Download Accession
External GI or ID list
TextFile Text Results
OutputText
Results
Load Sequences
Name
Files Results
MultipleSequences
Alignment
Outgroup
Multiple Sequences
Muscle
MultipleSequences
Alignment
OutputText
ClustalW2
MultipleSequences
Tree
Alignment
OutputText
Alignment
Alignment information
Alignment
Results
OutputText
Alignment
Results
OutputText
Results
OutputText

E)


Genome
UCSC 100 genomes
(For each) Extract Sequences
MultipleSequences
Genome
Genome
Sequences
Multiple Sequences
Muscle
Input
Output
MultipleSequences
Alignment
OutputText
Alignment
GBlocks
Input
Output
Alignment
Results
Alignment
OutputText
Alignment
RAxML
Input
Output
Alignment
Tree
Results
OutputText
Tree
Results
OutputText

F)

ANNEXE B

TABLEAUX SUPPLÉMENTAIRES DU CHAPITRE 4

Cette section présente plus en détails les flux de travaux décrits au chapitre 4 (voir ci-dessous).

Tableau B.1 Le jeu de données *Armadillo* comprenant 120 flux de travaux et leur classe respective.

Workflow	Classes	Workflow metadata
1	1	ClustalW2
2	1	Baliphy
3	1	Muscle
4	1	Probcons
5	1	Kalign
6	1	ClustalW2 + Kalign
7	1	ClustalW2 + Muscle
8	1	Probcons + BaliPhy
9	1	2 x Muscle + Probcons
10	1	ClustalW2 + Baliphy + Muscle
11	1	Muscle + BaliPhy + Kalign
12	1	Muscle + BaliPhy + Probcons
13	1	Kalign + ClustalW2 + Muscle
14	1	Kalign + ClustalW2 + Muscle + Probcons
15	1	Kalign + ClustalW2 + Muscle + BaliPhy
16	1	2 x Muscle + 2 x BaliPhy + Probcons
17	1	Muscle + ClustalW2 + 2 x BaliPhy + Probcons
18	1	2 x Muscle + 2 x BaliPhy + 2 x Probcons
19	1	2 x Kalign + ClustalW2 + Muscle
20	1	2 x Kalign + ClustalW2 + 3 x Muscle
21	1	Kalign + 2 x ClustalW2 + 2 x Muscle + Probcons + Baliphy
22	1	3 x ClustalW2 + 4 x Muscle
23	1	2 x Kalign + ClustalW2 + 3 x Muscle
24	1	2 x Kalign + ClustalW2 + Muscle + Alignment information
25	1	2x Kalign + ClustalW2 + Muscle + Probcons
26	1	2x Kalign + ClustalW2 + Muscle + BaliPhy
27	1	Kalign + ClustalW2 + Muscle + BaliPhy + Probcons
28	1	2 x Muscle + BaliPhy + Kalign
29	1	2 x ClustalW2 + Baliphy + Muscle
30	1	2 x Kalign + ClustalW2 + Muscle + BaliPhy
31	1	2 x Kalign + ClustalW2 + Muscle + Probcons
32	2	Garly
33	2	PhyML
34	2	ProtML
35	2	ProtPars
36	2	Seqboot + ProtPars
37	2	Garly + PhyML
38	2	Garly + ProtML
39	2	Garly + Seqboot + NeighBor
40	2	PhyML + ProtML
41	2	PhyML + Seqboot + Neighbor
42	2	Garly + PhyML + ProtML
43	2	Garly + PhyML + Seqboot + Neighbor
44	2	PhyML + ProtML + ProtPars
45	2	PhyML + ProtPars + Seqboot + Neighbor
46	2	PhyML + ProtML+ Seqboot + Neighbor
47	2	Garly + ProtPars + ProtML
48	2	Garly + ProtPars + Seqboot + Neighbor
49	2	Garly + ProtML + Seqboot + Neighbor
50	2	ProtML + ProtPars + Seqboot + Neighbor
51	2	Garly + PhyML + ProtML + ProtPars + Seqboot + Neighbor
52	2	Garly + ProtML + ProtPars + PhyML
53	2	Garly + ProtPars + PhyML + Seqboot + Neighbor
54	2	Garly + ProtML + PhyML + Seqboot + Neighbor
55	2	2 x PhyML + ProtPars + Seqboot + Neighbor
56	2	2 x PhyML + ProtML+ Seqboot + Neighbor
57	2	2 x PhyML + ProtML + ProtPars
58	2	2 x Garly + Seqboot + NeighBor
59	2	2 x Garly + PhyML + ProtPars + ProtML
60	2	2x PhyML + ProtML + ProtPars

Table B.1 (*suite*)

Workflow	Classes	Workflow metadata
61	2	7 x PhyML
62	2	2 x Garly + ProtPars + 3 x ProtML
63	2	3 x PhyML + ProtML+ Seqboot + Neighbor
64	2	4 x PhyML + ProtML
65	2	2 x Garly + ProtML + Seqboot + Neighbor
66	2	2 x PhyML + ProtML + Seqboot + Neighbor
67	3	HGT Detector
68	3	Riata
69	3	BLAST
70	3	Robinson and Fould distance (RF)
71	3	Random Tree
72	3	HGT Detector + Riata
73	3	HGT Detector + Random Tree
74	3	Riata + BLAST
75	3	Riata + Random Tree
76	3	HGT Detector + Riata + Blast
77	3	HGT Detector + Riata + Random Tree
78	3	Riata + RF + Random Tree
79	3	2 x Riata + RF + 2 x Random Tree
80	3	HGT Detector + Riata + RF + Random Tree
81	3	HGT Detector + Riata + RF + BLAST
82	3	HGT Detector + BLAST + Riata + RF + Random Tree
83	3	2 x HGT Detector + Riata + RF + Random Tree
84	3	2 x HGT Detector + Riata + RF + BLAST
85	3	2 x HGT Detector + Riata + RF
86	3	2 x Riata + RF + Random Tree
87	3	2 x HGT Detector + RF + Random Tree
88	3	2 x Riata + RF + Random Tree
89	3	2 x HGT Detector + Riata + Random Tree
90	3	RF + 6 x Random Tree
91	3	Riata + 6 x Random Tree
92	3	2 x HGT Detector + Riata + RF + 2 x BLAST
93	3	Riata + 7 x Random Tree
94	4	ClustalW2 + PhyML
95	4	ClustalW2 + Garly
96	4	Muscle + PhyML
97	4	Muscle + Garly
98	4	Muscle + PhyML + HGT Detector
99	4	Muscle + Garly + HGT Detector
100	4	Muscle + Garly + HGT Detector + Random Tree
101	4	Muscle + Garly + HGT Detector + BLAST
102	4	Muscle + PhyML + HGT Detector + Random Tree
103	4	ClustalW2 + PhyML + HGT Detector + Random Tree
104	4	Muscle + 2 x PhyML + HGT Detector + Random Tree
105	4	ClustalW2 + 2 x PhyML + HGT Detector + Random Tree
106	4	PhyML+ HGT Detector + Riata + RF + Random Tree
107	4	Muscle+ HGT Detector + Riata + RF + Random Tree
108	4	2 x HGT Detector + Riata + 2 x Random Tree
109	4	Muscle+ ClustalW2 + Riata + RF + Random Tree
110	4	2 x Muscle + PhyML + HGT Detector
111	4	2 x Muscle + Garly + HGT Detector
112	4	ClustalW2 + PhyML +ClustalW2 + Garly + Muscle + PhyML
113	4	2 x Muscle + Garly + HGT Detector + Random Tree
114	4	2 x Muscle + Garly + HGT Detector + BLAST
115	4	2 x Muscle + PhyML + HGT Detector + Random Tree
116	4	3 x Muscle + PhyML + Alignment Information
117	4	2 x ClustalW2 + PhyML + HGT Detector + Random Tree
118	4	ClustalW2 + Muscle + 2 x PhyML + 2 x HGT Detector + 2 x Random Tree
119	4	4 x ClustalW2 + PhyML
120	4	Alignment information

Tableau B.2 Le jeu de données *myExperiment* comprenant 100 flux de travaux et leur classe respective. Chaque flux de travaux est représenté par des métadonnées (définies par les utilisateurs); de plus, les tâches individuelles ne sont pas indiquées.

myExperiment identifier	Classes	Workflow metadata
162	1	Mygrid, example, tavema, biomoby, , snapdragon, image, annotation
149	1	Biomoby, text mining, gene symbol, disambiguation
150	1	Biomoby, disambiguation, gene symbol, text mining
151	1	Biomoby, disambiguation, gene symbol, text mining, svg
152	1	Biomoby, disambiguation, gene symbol, text mining
174	1	Biomoby, microarray, affymetrix, nugo, R, quality control
175	1	Biomoby, affymetrix, microarray, nugo, R, normalization
1377	1	Pubmed, xml, text mining, ucompare
1013	1	Exon, p53, regex, text mining
1014	1	Exon, p53, regex, text mining
1015	1	Exon, p53, regex, text mining
1016	1	Intron, p53, regex,text mining
1017	1	Intron, p53, regex, text mining
1018	1	Intron, p53, regex, text mining
1428	1	Annotation, example, biomoby, image, mygrid, snapdragon
199	2	EBI, sequence similarity search, ssearch, Smith-Waterman, FASTA
200	2	EBI, sequence similarity search, Smith-Waterman, ssearch,FASTA
220	2	EBI, protein, sequence similarity search, smith-waterman, scanps
218	2	EBI, protein, protein annotation, sequence similarity search, Smith-Waterman, mpsrch
201	3	EBI, sequence similarity search, ncbi blast, BLAST, blastall
1491	3	Sequence, BLAST, ddbj, protein, services
1492	3	Sequence, services, , BLAST, DDBJ, protein
1457	3	Nucleotide, sequence, services
23	3	BLAST, DDBJ, sequence, similarity
21	3	DDBJ, sequence, similarity, BLAST, simplifier
1450	3	DDBJ, protein, services, xml, BLAST
1451	3	DDBJ, BLAST, protein, sequence
329	3	DDBJ, BLAST, protein, sequence
32	3	BLAST, sequence
368	3	Protein, sequence, neuroscience, newcastle, BLAST
1452	3	BLAST, services, protein, sequence
1454	3	BLAST, nucleotide, sequence, services
1455	3	BLAST, nucleotide, sequence, services
1456	3	BLAST, nucleotide, sequence, services
1556	3	BLAST, accession
1525	3	DNA, sequence, alignment, grid, moteur, vbrowser, BLAST, conversion
1528	3	Alignment, BLAST, blat, conversion, DNA, grid, moteur, sequence
202	3	EBI, ncbi blast, sequence similarity search, user_interaction, BLAST blastall
1688	3	Blastall, blastp, EBI, ncbi blast, sequence similarity search, BLAST
1689	3	Blastall, EBI, ncbi blast, sequence similarity search, blastp, BLAST
208	3	BLAST, EBI, ncbi blast, sequence similarity search, PSI-BLAST, blastpgp
1493	3	BLAST, services, sequence, nucleotide
1494	3	BLAST, nucleotide, services
530	3	BLAST,
90	3	BLAST, sequence
197	3	BLAST, wu-blast, EBI, sequence similarity search
198	3	BLAST, wu-blast, EBI, sequence similarity search, user_interaction
1705	4	EBI, clustal, clustalw, multiple sequence alignment
203	4	EBI, multiple sequence alignment, clustal, clustalw
206	4	EBI, clustal, clustalw, tree, phylogenetic tree, neighbor-joining
3370	4	Alignment, clustalw, phylogeny, protein, tree, clustal, sequence
733	4	Phylogenetic tree, BLAST, clustalw, protein, BLAST, clustalw, database
532	4	BLAT
204	5	EBI, interpro, interproscan, GO terms, protein annotation, gff, protein family, protein motif
205	5	EBI, GO terms, interpro, interproscan, protein annotation, user_interaction,
829	5	Network, bioquali, pyquali, interaction
830	5	Bioquali, interaction, network, pyquali
211	6	EBI, protein, protein annotation, transmembrane, transmembrane prediction, phobius, gff
212	6	EBI, interproscan, protein, protein annotation, transmembrane, transmembrane prediction, gff
740	7	Image, compound, compound info + image, , inchi, ncbi, pubchem, pccompound

Table B.2 (*suite*)

myExperiment identifier	Classes	Workflow metadata
737	7	Pubchem, pug, compound info image
2139	7	Compound, structure, open source, open science
215	8	EBI, emboss, emboss tmap, transmembrane, transmembrane prediction,protein, gff
217	8	EBI, alignment, multiple sequence alignment, kalign
219	8	EBI, alignment, multiple sequence alignment, mafft
95	8	Phylogenomics, sifter, automatic function prediction, mafft, tree, moby,
221	8	EBI, alignment, ,multiple sequence alignment, muscle
1477	8	Kbws, g-language, sequence, protein, fasta, blast, muscle, weblogo, togows
222	8	EBI, alignment, multiple sequence alignment, t-coffee,
223	8	Accurate mass, chemspider, mass spectrometry, metabolomics, massbank
1527	8	Alignment, conversion, DNA, grid, moteur, sequence, taverna, vbrowser, blat
158	9	BIOMart, mygrid, taverna, emboss
214	9	EBI, emboss, emboss seqret, sequence retrieval, dbfetch
209	9	EBI, dbfetch
232	9	EBI, dbfetch,fasta format, ncbi, efetch, gi, sequence identifier, sequence retrieval
235	9	EBI, fasta format, gi, interpro, protein, uniparc, uniprot archive, picr, upi, dbfetch
233	10	EBI, pdb, protein, protein structure, structure, pairwise structure comparison
225	10	EBI, pdb, protein, protein structure, structure, structure retrieval, dbfetch
224	10	EBI, protein, structure, pdb, backbone,
165	10	Mygrid, soaplab, sequence, retrieval, rendering
236	11	EBI, gi, picr, protein, sequence identifier, uniparc
237	11	EBI, protein, sequence identifier, sequence retrieval, dbfetch
238	11	EBI, srs, interpro, protein, protein annotation, uniparc
244	11	EBI, nucleotide sequence, protein, censor, repeat masking
809	11	Web service, repeat, opal, example
808	11	Web service, repeat, opal, cluster, example
245	12	Tmascan-se, trnascan, tRNA, nucleotide sequence
36	12	Transformation, transgenic, translation
227	12	EBI, emboss, emboss getorf
4	13	Sanger sequencing, satellite, scaffold
906	13	GeNS, proteins, genomic name server, genomics, proteomics,
907	13	GeNS, proteomics, genomic name server, genomics, protein, convert
912	13	GeNS, database, regex
913	13	GeNS, database, list, regex
914	13	GeNS, database, organisms, regex
1696	14	EBI, interpro, interproscan, protein
1697	14	EBI, interpro, interproscan, protein
380	14	Structure, docking, ligand
39	15	Scaffold, SEG
3945	15	Component

ANNEXE C

LOGICIELS BIOINFORMATIQUES PARALLÉLISÉS

Tableau C.1 Survol de quelques logiciels bioinformatiques parallélisés

Logiciels	**Types**	***API***	**Références**
GARD	Détection de recombinants	MPI	Kosakovsky Pond *et al.* (2006)
fastDNAml	Inférence phylogénétique	MPI, PVM	Stewart *et al.* (2001)
mpiBLAST	Recherche de séquences	MPI	Darling *et al.* (2003)
Clustal-MPI	MSA*	MPI	Li (2003)
pCLUSTAL	MSA	MPI	Cheetham *et al.* (2003)
MSA-CUDA	MSA	CUDA	Liu *et al.* (2009a)
MSAProbs	MSA	CUDA	Liu *et al.* (2010a)
Hybrid ClustalW	MSA	MPI/OpenMP (hybrid)	Tan *et al.* (2005)
CUDA-BLASTP	Recherche de séquences	CUDA	Liu *et al.* (2011a)
mpiCUDA-BLASTP	Recherche de séquences	MPI/CUDA (hybrid)	Liu *et al.* (2011b)
MPI-HMMER	Recherche de séquences	MPI	Walters *et al.* (2007)
GPU-HMMER	Recherche de séquences	CUDA	Walters *et al.* (2009)
PhyML	Inférence phylogénétique	MPI / OpenMP (hybrid)	Guindon *et al.* (2010)
Recodon	Simulation de coalescence	MPI	Arenas et Posada (2007)
PyEvolve	Modélisation de l'évolution des séquences	MPI	Butterfield *et al.* (2004)
CUDASW++	MSA	CUDA	Liu *et al.* (2009b)
CUDA Smith-Waterman	MSA	CUDA	Manavski etValle (2008)

MAFFT	Inférence phylogénétique	Pthreads	Katoh et Toh (2010)
RAxML	Inférence phylogénétique	MPI, Pthreads, CUDA, OpenMP (plusieurs versions)	Stamatakis (2006) (Voir pour revue Pfeiffer and Stamatakis, 2010)
Garli	Inférence phylogénétique	MPI	Zwickl (2006)
MrBayes	Inférence phylogénétique	MPI	Altekar *et al.* (2004)
PAML	Analyse de la pression sélective	MPI	Chen *et al.* (2009)
Tree-Puzzle	Inférence phylogénétique	MPI	Schmidt *et al.* (2002)
PHYLIP	Inférence phylogénétique	MPI	Ropelewski *et al.* (2010)
BEAST	Inférence phylogénétique	MPI	Drummond et Rambaut (2007)
CUDA-MEME	Recherche de motifs	CUDA	Yongchao *et al.* (2010b)
mCUDA-MEME	Recherche de motifs	MPI/CUDA/OpenMP (*hybride*)	Liu *et al.* (2011c)
Dialign-TX	MSA	MPI/OpenMP (*hybride*)	Subramanian *et al.* (2011)
Dialign-P	MSA	MPI	Schmollinger *et al.* (2004)
Autodock4	Recherche d'appariement de structure 3D	MPI	Collignon *et al.* (2011)

* *MSA* (Alignement multiple de séquences), *MPI* (message passing interface), *CUDA* (Compute Unified Device Architecture)

ANNEXE D

SUPPLÉMENT SUR LE NOUVEAU CRITÈRE DE SUPPORT DE CLASSIFICATION

D.1 Introduction

Nous poursuivons, dans cette annexe, l'analyse du critère de support des flux de travaux individuel *PSG* (Section 4) en utilisant deux nouveaux jeux de données externes au domaine des flux de travaux. Ce critère de support en paire (*PSG*), basé sur la comparaison de plusieurs partitions, pourrait ainsi être utilisé pour l'évaluation d'autres types de données par des méthodes de partitionnement. L'élaboration de ce critère fait suite à l'observation que même en supposant un même nombre de classes, les algorithmes tels que *k*-means et *k*-medoids, qui sont des heuristiques, ne produisent pas toujours les mêmes résultats de partitionnement, *c.-à-d.* ne classe pas toujours dans la même classe les différentes paires d'éléments (voir par exemple la Figure D.1).

D.2 Méthodes

Nous avons étudié ce nouveau critère sur deux jeux de données distincts : le jeu de données quantitatives classique de *Iris*[35] (Fisher, 1936), utilisé dans plus de 45 études (Xu et Wunsch, 2005), et le jeu de données de *Zoo*[36] (Forsyth, 1990) utilisé dans 16 études (Bache et Lichman, 2013). Ces deux jeux de données multivariables ont été sélectionnés car ils ne contenaient pas de données manquantes. De plus, ces deux jeux de données peuvent difficilement être traités par des méthodes linéaires de séparation en groupes. Nous avons utilisé dans cette partie de l'étude les algorithmes de partitionnement *k*-means et *k*-medoids

---

[35] https://archive.ics.uci.edu/ml/datasets/Iris
[36] https://archive.ics.uci.edu/ml/datasets/Zoo

avec 1000 répétitions (*random starts*). Nous avons utilisé la distance Euclidienne et, quand applicable, soit l'indice de Calinski-Harabasz, soit l'indice Silhouette comme critères d'optimisation. Les figures des différentes classes ont été générées avec le logiciel ELKI[37] (Achtert *et al.*, 2008). Les statistiques ont été obtenues avec le logiciel Instat v3.0 et le test de Student-Newman-Keuls.

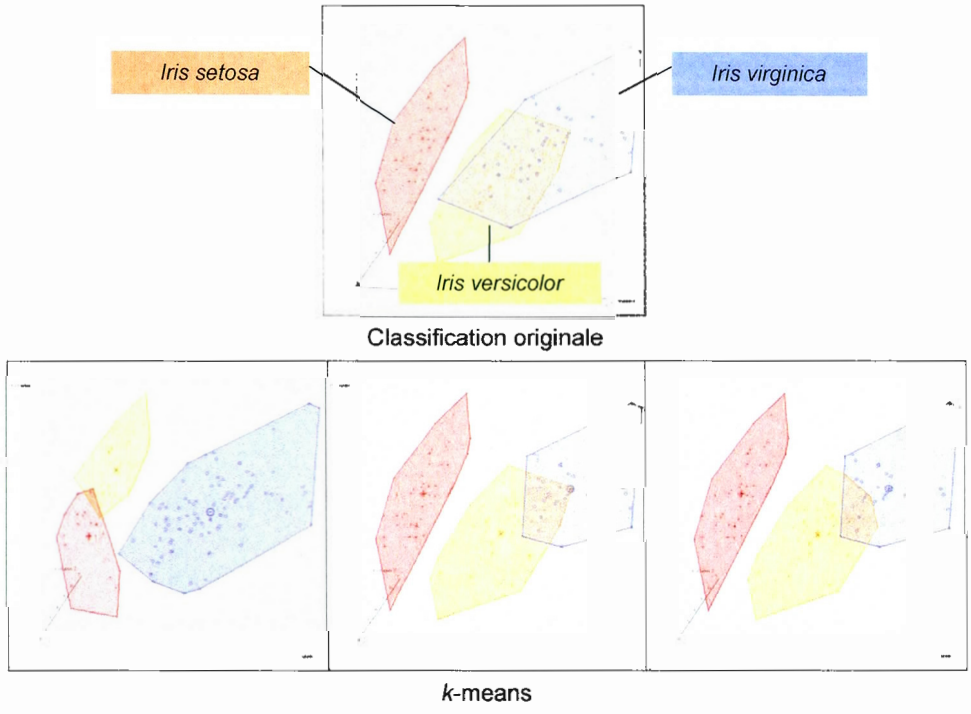


Figure D.1 Différents résultats de l'algorithme *k*-means sur le jeu de données de *Iris*. La figure du haut représente la classification originale des données en trois classes (K=3), selon Fisher (1936). Les différentes classifications du bas ont été obtenues avec différentes exécutions de l'algorithme de regroupement par partitionnement *k*-means en utilisant des *random starts* différents.

[37] http://elki.dbs.ifi.lmu.de/

D.2.1 Jeu de données de *Iris*

Le jeu de données de *Iris* (Fisher, 1936) contient trois (K=3) classes (espèces d'iris : *Iris setosa*, *Iris versicolore* et *Iris virginica*) composées chacune de 50 échantillons. Chacun de ces échantillons est décrit par quatre mesures : longueur des sépales, largeur des sépales, longueur des pétales et largeur des pétales.

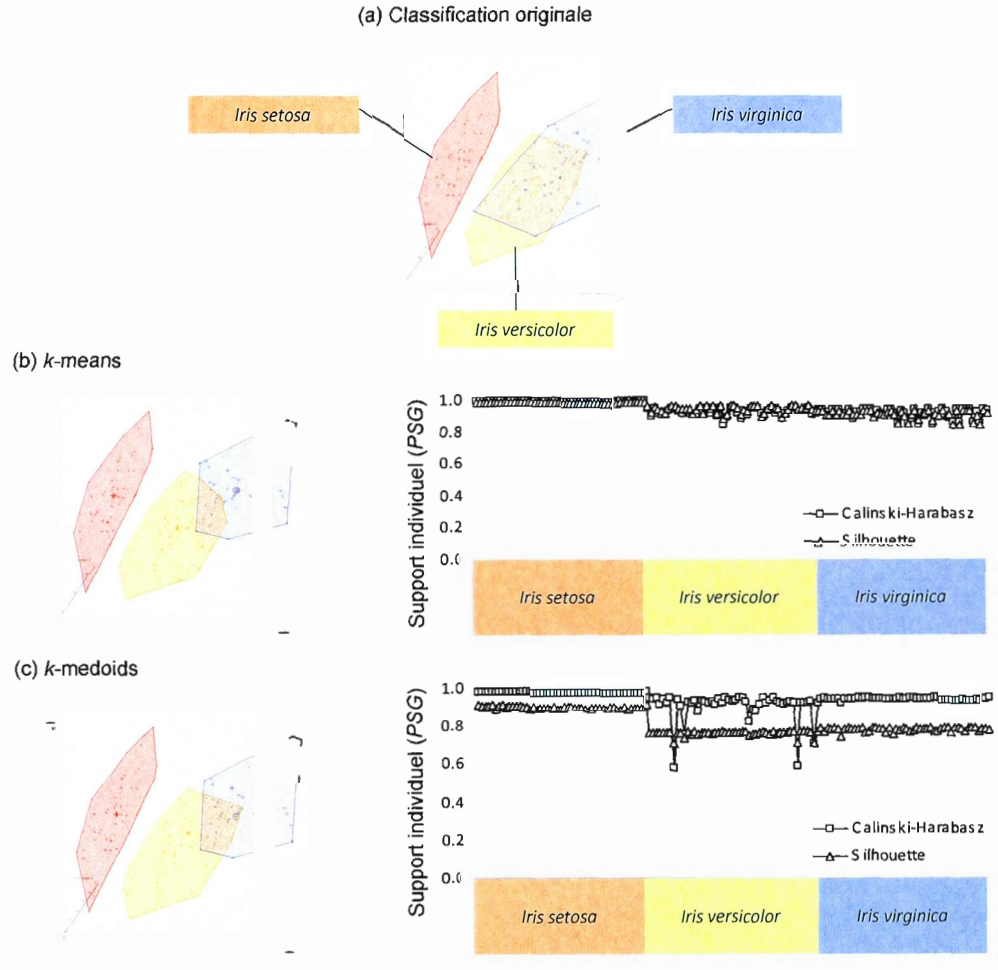


Figure D.2 Jeu de données de *Iris*. En (a) classification originale des données. En (b) classification des données par la méthode *k*-means en utilisant la distance Euclidienne. En (c) classification des données par la méthode *k*-medoids en utilisant la distance Euclidienne. On

retrouve à gauche le résultat du regroupement des échantillons en trois classes (K=3). À droite, de chaque classification obtenue, nous présentons les valeurs de support individuelles (*PSG*) pour chacun des échantillons. L'axe des abscisses indique les classes originales retrouvées par Fisher (1936) pour ce jeu de données.

Ce jeu de données présente des difficultés pour la classification par les algorithmes de types *k*-means car les espèces *Iris versicolore* et *Iris virginica* sont difficilement classifiable en utilisant les méthodes linéaires (Fisher, 1936; Xu et Wunsch, 2005; Gehlenborg et Wong, 2012).

D.2.2 Résultats et conclusions pour le jeu de données de *Iris*

L'analyse des résultats, suite au regroupement et au calcul du support individuel par paires, démontre que suite à l'application des deux algorithmes de partitionnement *k*-means et *k*-medoids, les classes *Iris versicolor* et *Iris virginica* sont effectivement difficilement séparables par ces algorithmes (Figure D.2).

Tableau D.1 Support moyen individuel (*PSG*±SD) des différentes espèces d'iris de la Figure D.2 en fonction du type du regroupement et de l'indice d'optimisation

Méthode de regroupement	**Indices d'optimisation**	***Iris setosa***	***Iris versicolor***	***Iris virginica***
k-mean	Calinski-Harabasz	0.99±0.01	0.92±0.02	0.91±0.03
	Silhouette	0.98±0.01	0.94±0.02	0.90±0.03
k-medoids	Calinski-Harabasz	0.98±0.01	0.90±0.08	0.94±0.01
	Silhouette	0.90±0.01	0.76±0.01	0.78±0.01

Toutefois, le calcul du nouveau critère de support individuel (*PSG*) des différents échantillons d'iris permet de comparer le support de chacune des classes. Ainsi, on peut remarquer que dans le cas de l'algorithme *k*-means, les classes *Iris versicolore* et *Iris virginica* sont beaucoup moins supportées que la classe *Iris setosa* (Figure D.2a). On observe le même phénomène en utilisant l'algorithme *k*-medoids (Figure D.2b). On retrouve, dans le cas de l'algorithme *k*-means, une valeur de support moyen respectif de 0.99 et de 0.98 pour la

classe *Iris setosa* (Tableau D.1) pour les indices Calinski-Harabasz (CH) et Silhouette (Sil). Cependant, le support individuel moyen est plutôt de 0.92 avec CH et de 0.94 avec Sil pour la classe *Iris versicolor* et de 0.91 pour CH et de 0.90 pour Sil pour la classe *Iris virginica* ($p<0.001$; *Iris setosa* vs *Iris versicolor* et *Iris setosa* vs *Iris virginica*). Dans le cas de la méthode de partitionnement *k*-medoids, on peut observer que cette différence est encore plus marquée. Soulignons qu'un support de 1.0 signifie, dans le cas de ce critère, que les paires d'échantillons évaluées se retrouvent toujours dans la même classe (voir le chapitre 4).

En conclusion, dans le cas du jeu de données de *Iris*, le nouveau critère de support par paires permet, sans avoir à recourir à des méthodes de visualisation, telles que les *scatter-plot matrix*, suggérées par Gehlenborg et Wong (2012), de connaître le support de certaines classes lors du regroupement.

D.2.3 Jeu de données de *Zoo*

Le jeu de données de *Zoo* (Forsyth, 1990) est composé des données sur 101 animaux. Ces données sont divisées en 7 classes empiriques (Tableau D.2). Chaque animal dans ce jeu de données zoologique est alors décrit par 16 variables binaires indiquant la présence (1) ou l'absence (0) de certaines caractéristiques de ces espèces. On retrouve ainsi dans ce jeu de données les variables suivantes : présence de poils, les plumes, œufs, lait, volant, aquatique, prédateur, avec dentition, avec colonne vertébrale, respiration, venimeux, palmé, queue, domestique, grosseur d'un chat. De plus, une dernière variable numérique non-continue indique le nombre de pattes que possède chacune des espèces animales.

Tableau D.2 Classes empiriques du jeu de données de *Zoo*

Classes (nombre d'animaux)	**Animaux***
1. Mammifères (41)	aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, **dolphin**, elephant, **fruitbat**, giraffe, **girl**, goat, **gorilla**, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, **porpoise**, puma, pussycat, raccoon, reindeer, **seal**, **sealion**, **squirrel**, **vampire**, vole, **wallaby**,wolf
2. Oiseaux (20)	chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren
3. Reptiles (5)	pitviper, seasnake, slowworm, **tortoise**, **tuatara**
4. Poissons (13)	bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
5. Amphibiens (4)	frog, frog, newt, toad
6. Insectes (8)	flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
7. Invertébrés (10)	clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

* Les noms d'espèces ont été conservés en anglais pour faciliter la comparaison avec d'autres études. En gras, les espèces ayant une valeur de support individuel (*PSG*) plus basse à la Figure D.4.

Comme le jeu de données de *Iris*, le jeu de données multivariables de *Zoo* ne peut pas être bien partitionné à l'aide des méthodes linéaires. Il est utilisé comme jeu de données de référence pour des algorihmes tels que le *k*-Nearest Neighbor (*k-NN*) ou les réseaux de neurones (voir par exemple McKenzie et Forsyth, 1995).

D.2.4 Résultats et discussion pour le jeu de données de Zoo

Nous avons examiné le jeu de données de *Zoo* en utilisant la même méthode d'analyse que le jeu de données de *Iris* (Figure D.3). Cependant, un regroupement hiérarchique par la méthode de Neighbor-Joining (Saitou et Nei, 1987) a aussi été réalisé en utilisant le logiciel Neighbor du *package* PHYLIP (Felsenstein, 2005) pour bien visualiser chacune des classes.

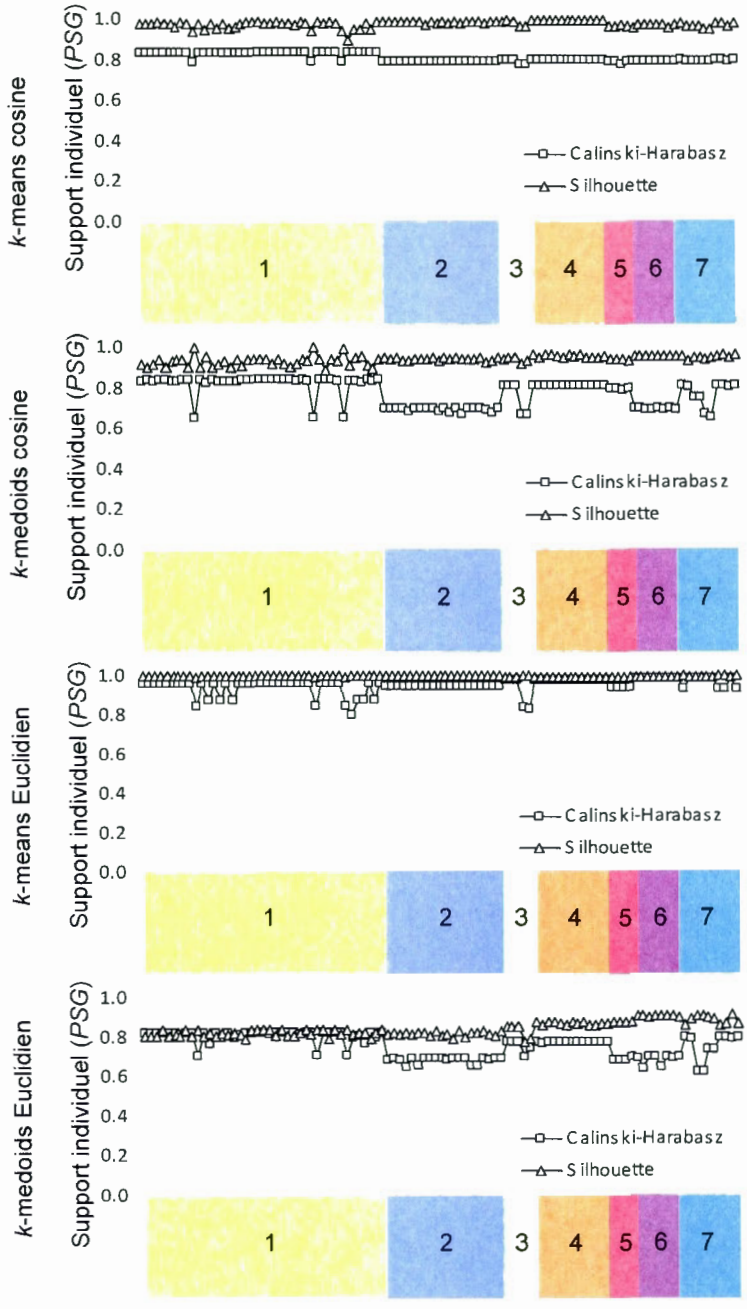


Figure D.3 Valeurs de support individuel, *PSG,* des animaux du jeu de données de *Zoo* pour les différentes méthodes de regroupement par partitionnement, les distances cosine et Euclidienne et les indices de Calinski-Harabasz ou Silhouette. L'axe des abscisses présente les classes originales définies par Forsyth (1990) pour ce jeu de données.

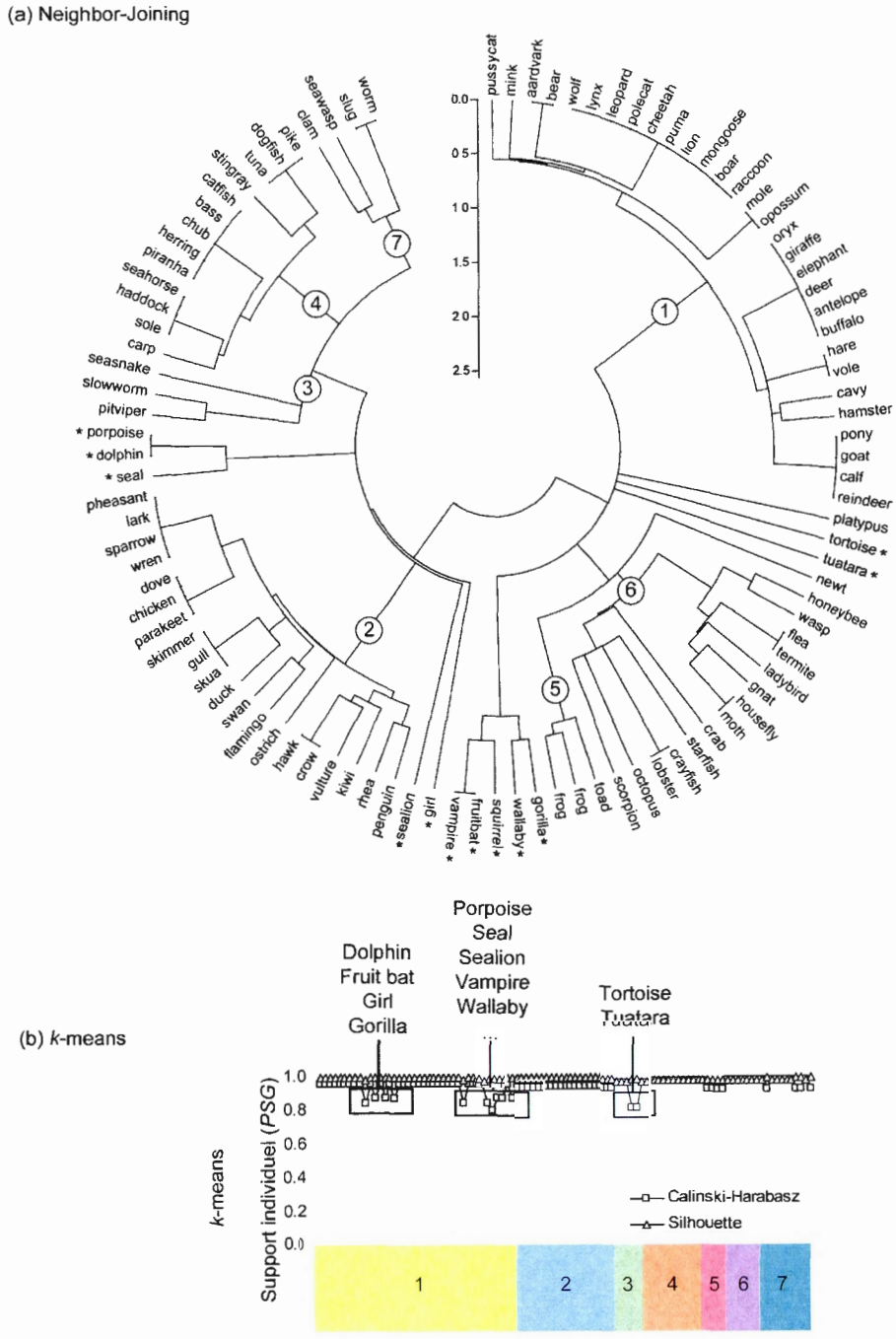


Figure D.4 Regroupement hiérarchique par Neighbor-Joining (a) ou par partitionnement de type *k*-means (b) des animaux du jeu de données de *Zoo*. La distance Euclidienne a été utilisée dans les deux cas. En (b), certaines espèces montrent une valeur de

support individuel *PSG* moins importante par rapport à leur groupe respectif. Elles sont mises en évidence dans la classification hiérarchique en (a) par des étoiles accolées aux noms des espèces. Les grandes classes sont aussi indiquées dans cette classification hiérarchique (numérotées de 1 à 7; voir le Tableau D.2). L'axe des abscisses, en (b), présente les classes originales définies par Forsyth (1990) pour ce jeu de données.

Notre première observation, Figure D.3, est que l'application de certaines méthodes de partitionnement sur ce jeu de données (*p.ex. k*-medoids et distance cosine) résulte en des valeurs de supports individuels (*PSG*) beaucoup plus faibles pour certains groupes lorsque l'on utilise l'indice de Calinski-Harabasz comme critère d'optimisation. Les valeurs de support obtenus en utilisant l'indice Silhouette sont moins affectées dans toutes les conditions (voir Figure D.3). De plus, on peut observer que certains animaux ont une valeur de support moins importante que les autres : *dolphin*, *girl*, *gorilla*, etc. (Figure D.4b; les animaux en gras dans le Tableau D.2). On observe aussi que, en utilisant la distance Euclidienne, ces animaux se retrouvent dans les mêmes sous-arbres dans la classification hiérarchique par Neighbor-Joining (Figure D.4a). Par exemple, le marsouin (*porpoise*), un parent du dauphin de la famille des *Phocoenidae,* est souvent mal identifié comme mammifère (*mammal*) par les algorithmes de types *k-NN* en utilisant la distance Euclidienne (McKenzie et Forsyth, 1995). Bien que ce ne soit pas le cas dans nos regroupements, le marsouin étant classé avec les dauphins dans le cas du regroupement par *k*-means (*non montré*) et Neighbor-Joining (voir Figure D.4), l'information apportée par le nouveau critère de support par paires, *PSG,* permet de croire que le classement de cet animal devrait quand même faire l'objet d'une vérification additionnelle. C'est aussi le cas de *girl* et du gorille dans notre classification qui ont un support beaucoup plus faible que les autres animaux en utilisant l'indice de Calinski-Harabasz (Figure D.4b). Dans ce cas, il pourrait être intéressant de créer une autre classe pour les animaux qui sont plus souvent regroupés ensemble qu'aux autres animaux de leur classe respective.

D.3 Conclusions

Ce nouveau critère, sans avoir à utiliser des techniques telles que le *bootstrap* ou *jacknife* (Henning, 2008), permet de mesurer le support de chacun des éléments à sa classe respective en mesurant un support par paires. Il permet, en outre, de déterminer les éléments pouvant être considérés comme aberrants (*outliers*) de façons algorithmiques. Les éléments aberrants sont connus pour influencer négativement différents types de classification, en outre, la classification par *k*-means et *k*-medoids (Henning, 2007). On emploie normalement pour répondre à cette problématique des techniques graphiques comme le tracé ordonné des données versus une approximation de leur moyenne (*Q-Q plot*) (Wilk et Gnanadesikan, 1968) ou encore on utilise une *scatter-plot matrix* (Gehlenborg et Wong, 2012). Toutefois, de telles méthodes graphiques ne peuvent être appliquées lors de simulations sur de grands jeux de données.

Récemment, une technique pour détecter les biais dans des regroupement hiérarchiques lors de l'inférence d'arbres phylogénétiques, basée sur l'analyse des bipartitions, a été développée pour des jeux de données de milliers d'espèces (Dao *et al.*, 2013). Cependant, à notre connaissance, aucune technique semblable n'existe pour les méthodes de partitionnement pour des ensembles de données d'une telle envergure. Le nouveau critère de support, développé à la base pour les flux de travaux, permet ainsi de détecter les éléments aberrants dans des ensembles de données volumineux. De plus, notre nouvelle approche peut être appliquée en se servant de plusieurs critères d'optimisation (voir le chapitre 4). On peut ainsi l'utiliser avec la méthode de partitionnement *k*-means, dans laquelle le critère de Calinsk-Harabasz est préféré (voir chapitre 5), ou encore avec la méthode de partitionnement *k*-medoids pour laquelle l'indice Silhouette est recommandé. En conclusion, il serait intéressant d'appliquer ce nouveau critère sur des jeux de données très volumineux, mais ayant des biais bien établis, pour pouvoir le comparer aux différentes techniques de détection de valeurs aberrantes existantes dans le cadre des méthodes de partitionnement.

BIBLIOGRAPHIE

Abascal, F., Zardoya, R., et Posada, D. (2005). ProtTest: selection of best-fit models of protein evolution. *Bioinformatics, 21*(9), 2104-2105.

Abouelhoda, M., Alaa, S., et Ghanem, M. (2010). Meta-workflows: pattern-based interoperability between Galaxy and Taverna. Dans *Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science* (Wands '10). ACM, 2-10.

Abouelhoda, M., Issa, S. A., et Ghanem, M. (2012). Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support. *BMC Bioinformatics*, *13*(1), 2-19.

Achtert, E., Kriegel, H. P., Reichert, L., Schubert, E., Wojdanowski, R., et Zimek, A. (2010). Visual evaluation of outlier detection models. In *Database Systems for Advanced Applications* (pp. 396-399). Springer Berlin Heidelberg.

Addario-Berry, L., Hallett, M., et Lagergren, J. (2003). Towards identifying lateral gene transfer events. *Pacific Symposium on Biocomputing, 8*, 279-290.

Afgan, E., Baker, D., Coraor, N., Chapman, B., Nekrutenko, A., et Taylor, J. (2010). Galaxy CloudMan: Delivering Cloud Compute Clusters, *BMC Bioinformatics*, *11*(suppl. 12), S4.

Agharbaoui, Z., Leclercq, M., Remita, M. A., Badawi, M. A., Lord, E., Houde, M., Danyluk, J., Diallo, A. B. et Sarhan, F. (2015). An integrative approach to identify hexaploid wheat miRNAome associated with development and tolerance to abiotic stress. *BMC Genomics, 16*(1), 339.

Altekar, G., Dwarkadas, S., Huelsenbeck, J. P., et Ronquist, F. (2004). Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics, 20*(3), 407-415.

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., et Mock, S. (2004). Kepler: an extensible system for design and execution of scientific workflows. Dans *16th International Conference on Proceedings of Scientific and Statistical Database Management*, IEEE, 423-424.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., et Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, *215*(3), 403-410.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Perez, J.M., et Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243-256.

Archibald, J. M. (2008). The eocyte hypothesis and the origin of eukaryotic cells. *Proceedings of the National Academy of Sciences USA*, *105*(51), 20049-20050.

Arenas, M., et Posada, D. (2007). Recodon: coalescent simulation of coding DNA sequences with recombination, migration and demography. *BMC bioinformatics*, *8*(1), 458.

Arthur, D., et Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. Dans *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (p. 1027-1035). Society for Industrial and Applied Mathematics.

Bache, K. et Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., et Vassilvitskii, S. (2012). Scalable k-means++. *Proceedings of the VLDB Endowment*, *5*(7), 622-633.

Ball, G. H., et Hall, D. J. (1965). ISODATA, a novel method of data analysis and pattern classification. Stanford Research Institute, CA. Disponible à l'adresse : www.dtic.mil/cgi-bin/GetTRDoc?AD=AD0699616.

Bansal, M. S., Burleigh, J. G., Eulenstein, O., et Fernández-Baca, D. (2010). Robinson-Foulds supertrees. *Algorithms for Molecular Biology*, 5(1), 18.

Bao, S., Jiang, R., Kwan, W., Wang, B., Ma, X., et Song, Y. Q. (2011). Evaluation of next-generation sequencing software in mapping and assembly. *Journal of human genetics*, *56*(6), 406-414.

Bapteste, E., Brinkmann, H., Lee, J. A., Moore, D. V., Sensen, C. W., Gordon, P., Duruflé, L., Gaasterland, T., Lopez, P., Müller, M., et Philippe, H. (2002). The analysis of 100 genes supports the grouping of three highly divergent amoebae: Dictyostelium, Entamoeba, and Mastigamoeba. *Proceedings of the National Academy of Sciences*, *99*(3), 1414-1419.

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I. F., Soboleva, A., Tomashevsky, M., et Edgar, R. (2007). NCBI GEO: mining tens of millions of expression profiles—database and tools update. *Nucleic acids research*, *35*(suppl 1), D760-D765.

Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M. H., et Vahi, K. (2008). Characterization of scientific workflows. Dans *IEEE Third Workshop on Workflows in Support of Large-Scale Science*, WORKS 2008, IEEE. (p. 1-10).

Batuwita, R., et Palade, V. (2009). microPred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, *25*(8), 989-995.

Baughman, J. M., Perocchi, F., Girgis, H. S., Plovanich, M., Belcher-Timme, C.A., Sancak, Y., Bao, X.R., Strittmatter, L., Goldberger, O., Bogorad, R.L., Koteliansky, V., et Mootha, V. K. (2011). Integrative genomics identifies MCU as an essential component of the mitochondrial calcium uniporter. *Nature*, *476*(7360), 341-345.

Beaulah, S. A., Correll, M. A., Munro, R. E. J., et Sheldon, J. G. (2008). Addressing informatics challenges in Translational Research with workflow technology. *Drug Discovery Today, 13*(17), 771-777.

Begel, A. (1996). *LogoBlocks: A graphical programming language for interacting with the world.* Electrical Engineering and Computer Science Department, MIT, Boston. Ph.D. thesis. Disponible à l'adresse : http://research.microsoft.com/en-us/um/people/abegel/mit/begel-aup.pdf.

Beiko, R. G., Harlow, T. J., et Ragan, M. A. (2005). Highways of gene sharing in prokaryotes. *Proceedings of the National Academy of Sciences USA*, *102*(40), 14332-14337.

Ben-Hur, A., Elisseeff, A., et Guyon, I. (2001). A stability based method for discovering structure in clustered data. *Pacific symposium on biocomputing*, *7*, 6-17.

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., et Sayers, E. W. (2013). GenBank. *Nucleic Acids Research*, *41*(Database issue), D36-D42.

Berkhin, P. (2006). A survey of clustering data mining techniques. Dans *Grouping multidimensional data* (p. 25-71). Springer Berlin Heidelberg.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers. 272 pages (*réimpression de 2012*).

Bhagat, J., Tanoh, F., Nzuobontane, E., Laurent, T., Orlowski, J., Roos, M., Wolstencroft, K., Aleksejevs, S., Stevens, R., Pettifer, S., Lopez, R., et Goble, C. A. (2010). BioCatalogue: a universal catalogue of Web services for the life sciences. *Nucleic Acids Research*, *38*(Web Server issue), W689-W694.

Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M. H., et Vahi, K. (2008). Characterization of scientific workflows. Dans *Third Workshop on Workflows in Support of Large-Scale Science,* (WORKS 2008), IEEE, 1-10.

Bininda-Emonds, O. R. (2004). The evolution of supertrees. *Trends in Ecology & Evolution*, *19*(6), 315-322.

Blankenberg, D., Taylor, J., et Nekrutenko, A. (2011). Making whole genome multiple alignments usable for biologists. *Bioinformatics*, *27*(17), 2426-2428.

Boc, A., Diallo, A. B., et Makarenkov, V. (2012). T-REX: a web server for inferring, validating and visualizing phylogenetic trees and networks, *Nucleic Acids Research, 40*(Web Server issue), W573-W579.

Boc, A., et Makarenkov, V. (2011). Towards an accurate identification of mosaic genes and partial horizontal gene transfers. *Nucleic acids research, 39*(21), e144-e144.

Boc, A., et Makarenkov, V. (2003). New efficient algorithm for detection of horizontal gene transfer events. Dans *Algorithms in bioinformatics* (p. 190-201). Springer Berlin Heidelberg.

Boc, A., Philippe, H., et Makarenkov, V. (2010). Inferring and validating horizontal gene transfer events using bipartition dissimilarity. Systematic biology, *59*(2)-195-211.

Bock, H. H. (2007). Clustering methods: a history of K-Means algorithms. Dans *Selected contributions in data analysis and classification* (p. 161-172). Springer Berlin Heidelberg.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M. C., Estreicher, A., Gasteiger, E., et Schneider, M. (2003). The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic acids research, 31*(1), 365-370.

Bonnet, E., He, Y., Billiau, K., et Van de Peer, Y. (2010). Tapir, a web server for the prediction of plant microrna targets, including target mimics. *Bioinformatics, 26*(12), 1566-1568.

Boussau, B., et Daubin, V. (2010). Genomes as documents of evolutionary history. *Trends in ecology & evolution, 25*(4), 224-232.

Bowers, S., McPhillips, T., Riddle, S., Anand, M. K., et Ludäscher, B. (2008). Kepler/pPOD: Scientific workflow and provenance support for assembling the tree of life. Dans *Provenance and Annotation of Data and Processes* (p. 70-77). Springer Berlin Heidelberg.

Bradley, P. S., Bennett, K. P. et Demiriz, A. (2000). Constrained k-means clustering. [Rapport technique MSR-TR-2000-65]. Microsoft Research, Redmond, WA.

Brazas, M. D., Yamada, J. T., et Ouellette, B. F. (2010). Providing web servers and training in Bioinformatics: 2010 update on the Bioinformatics Links Directory. *Nucleic Acids Research, 38*(Web Server issue), W3-W6.

Brochier, C., Lopez-Garcia P., et Moreira, D. (2004). Horizontal gene transfer and archaeal origin of deoxyhypusine synthase homologous genes in bacteria. *Gene, 330*, 169-176.

Brochu-Gaudreau, K., Rehfeldt, C., Blouin, R., Bordignon, V., Murphy, B. D., et Palin, M. F. (2010). Adiponectin action from head to toe. *Endocrine, 37*(1), 11-32.

Bunke, H., er Allermann, G. (1983). Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, *1*(4), 245-253.

Bunke, H., Foggia, P., Guidobaldi, C., Sansone, C., et Vento, M. (2002). A comparison of algorithms for maximum common subgraph on randomly connected graphs. Dans *Structural, Syntactic, and Statistical Pattern Recognition* (p. 123-132). Springer Berlin Heidelberg.

Bunke, H., et Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern recognition letters*, *19*(3), 255-259.

Burge, S. W., Daub, J., Eberhardt, R., Tate, J., Barquist, L., Nawrocki, E. P., Eddy, S. R., Gardner, P. P., et Bateman, A. (2013). Rfam 11.0: 10 years of RNA families. *Nucleic Acids Research*, *41*(Database issue), D226-D232.

Butterfield, A., Vedagiri, V., Lang, E., Lawrence, C., Wakefield, M.J., Isaev, A., et Huttley, G.A. (2004). PyEvolve: a toolkit for statistical modelling of molecular evolution. *BMC Bioinformatics*, *5*(1), 1.

Caliński, T., et Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*(1), 1-27.

Callaghan, S., Deelman, E., Gunter, D., Juve, G., Maechling, P., Brooks, C., Vahi, K., Milner, K., Graves, R., Field, E., Okaya, D., et Jordan, T. (2010). Scaling up workflow-based applications. *Journal of Computer and System Sciences*, *76*(6), 428-446.

Campbell, V., Legendre, P., et Lapointe, F. J. (2009). Assessing congruence among ultrametric distance matrices. *Journal of classification*, *26*(1), 103-117.

Castresana, J. (2000). Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Molecular Biology and Evolution*, *17*(4), 540-552.

Carollo, V., Matthews, D. E., Lazo, G. R., Blake, T. K., Hummel, D. D., Lui, N., Hane, D. L. et Anderson, O. D. (2005). GrainGenes 2.0. An improved resource for the small-grains community. *Plant physiology*, *139*(2), 643-651

Cavalli-Sforza, L. L. et Edwards, A. W. (1967). Phylogenetic analysis: Models and estimation procedures. *American Journal of Human Genetics*, *19*(3 part 1), 233-257.

Cheetham, J., Dehne, F., Pitre, S., Rau-Chaplin, A., et Taillon, P. J. (2003). Parallel clustal w for pc clusters. Dans *Computational Science and Its Applications*, ICCSA 2003, (p. 300-309), Springer Berlin Heidelberg.

Chen, S. H., Su, S. Y., Lo, C. Z., Chen, K. H., Huang, T.J., Kuo, B. H., et Lin, C. Y. (2009). PALM: a paralleled and integrated framework for phylogenetic inference with automatic likelihood model selectors. *PLoS One*, *4*(12), e8116.

Chen, W., Silva, R. F. D., Deelman, E., et Sakellariou, R. (2013) Balanced task clustering in scientific workflows. Dans *9th International Conference on eScience*: 22-25 Octobre 2013, Beijing. Los Alamitos, IEEE Computer Society, 188-195.

Milligan, G. W., et Cheng R (1996) Measuring the influence of individual data points in a cluster analysis. *Journal of Classification*, *13*(2), 315-335.

Chevenet, F., Croce, O., Hebrard, M., Christen, R., et Berry, V. (2010). ScripTree: scripting phylogenetic graphics. *Bioinformatics*, *26*(8), 1125-1126.

Childs, K. L., Hamilton, J. P., Zhu, W., Ly, E., Cheung, F., Wu, H., Rabinowicz, P. D., Town, C. D., Buel, R., et Chan, A. P. (2007). The TIGR plant transcript assemblies database. *Nucleic Acids Research*, *35*(suppl. 1), D846-D851.

Choi, S. S., Cha, S. H., et Tappert, C. C. (2010). A survey of binary similarity and distance measures. Journal of Systemics, *Cybernetics and Informatics*, *8*(1), 43-48.

Churches, D., Gombas, G., Harrison, A., Maassen, J., Robinson, C., Shields, M., Taylor, I., et Wang, I. (2006). Programming Scientific and Distributed Workflow with Triana Services. *Concurrency and Computation: Practice and Experience*, *18*(Special Issue: Workflow in Grid Systems), 1021-1037.

Ciccarelli, F. D., Doerks, T., Von Mering, C., Creevey, C. J., Snel, B., et Bork, P. (2006). Toward automatic reconstruction of a highly resolved tree of life. *Science*, *311*(5765), 1283-1287.

Cohen-Boulakia, S., Chen, J., Missier, P., Goble, C., Williams, A. R., et Froidevaux, C. (2014). Distilling structure in Taverna scientific workflows: a refactoring approach. *BMC bioinformatics*, *15*(suppl. 1), S12.

Collignon, B., Schulz, R., Smith, J. C., et Baudry, J. (2011). Task-parallel message passing interface implementation of Autodock4 for docking of very large databases of compounds using high-performance super-computers. *Journal of Computational Chemistry*, *32*(6):1202-1209.

Conte, D., Foggia, P., Sansone, C., et Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, *18*(3), 265-298.

Conte, D., Foggia, P., et Vento, M. (2007). Challenging complexity of maximum common subgraph detection algorithms: a performance analysis of three algorithms on a wide database of graphs. *Journal of Graph Algorithms and Applications, 11*(1), 99-143.

Conte, D., Guidobaldi, C., et Sansone, C. (2003). A comparison of three maximum common subgraph algorithms on a large database of labeled graphs. Dans *Graph Based Representations in Pattern Recognition,* (p. 130-141). Springer Berlin Heidelberg.

Cordella, L. P., Foggia, P., Sansone, C., et Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,*26*(10), 1367-1372.

Cortez, D., Delaye, L., Lazcano, A., et Becerra, A. (2009). Composition-based methods to identify horizontal gene transfer. Dans *Horizontal Gene Transfer* (p. 215-225). Humana Press.

Costa, F., de Oliveira, D., Ogasawara, E., Lima, A. A., et Mattoso, M. (2012). Athena: text mining based discovery of scientific workflows in disperse repositories. Dans *Resource Discovery* (p. 104-121). Springer Berlin Heidelberg.

Da Silva, R. F., Juve, G., Deelman, E., Glatard, T., Desprez, F., Thain, D., Tovar, B., et Livny, M. (2013). Toward fine-grained online task characteristics estimation in scientific workflows. Dans *Proceedings of the 8th Workshop on Workflows in Support of Large-Scale Science*, ACM, 58-67.

Dagan, T. (2011). Phylogenomic networks. *Trends in microbiology*, *19*(10), 483-491.

Dagan, T., et Martin, W. (2007). Ancestral genome sizes specify the minimum rate of lateral gene transfer during prokaryote evolution. *Proceedings of the National Academy of Sciences USA*, *104*(3), 870-875.

Darling, A. E., Carey, L., et Feng, W. (2003) The Design, Implementation, and Evaluation of mpiBLAST. Dans *ClusterWorld Conference & Expo and the 4th International Conference on Linux Clusters: The HPC Revolution 2003*. Récupéré de: http://www.upf.edu/scb/_pdf/mpiBLAST.pdf.

Dao, D., Flouris, T., et Stamatakis, A. (2013). Automated plausibility analysis of large phylogenies. [Rapport technique]. Karlsruhe Institute of Technology. Récupéré de: http://sco.h-its.org/exelixis/pubs/Exelixis-RRDR-2013-6.pdf.

Darty, K., Denise, A., et Ponty, Y. (2009). Varna: Interactive drawing and editing of the rna secondary structure. *Bioinformatics*, *25*(15), 1974-1975.

Darwin, C. (1859). On the origins of species. John Murray, London.

David, M., Dzamba, M., Lister, D., Ilie, L., et Brudno, M. (2011). SHRiMP2: sensitive yet practical short read mapping. *Bioinformatics*, *27*(7), 1011-1012.

Davies, D. L., et Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*(2), 224-227

Day, W. H. E. (1985). Optimal algorithms for comparing trees with labeled leaves. *Journal of Classification*, *2*(1), 7-28.

Dayhoff., M. O., Eck. R. V., et Park, C. M. (1972). A model of evolutionary change in proteins. Dans *Atlas of Protein Sequence and Structure*, M. O. Dayhoff, (dir.), Volume 5. National Biomedical Research Foundation, Washington, D.C, pages 89-99.

De Oliveira, D., Ocaña, K. A., Ogasawara, E., Dias, J., Gonçalves, J., Baião, F., et Mattoso, M. (2013). Performance evaluation of parallel strategies in public clouds: A study with phylogenomic workflows. *Future Generation Computer Systems*, *29*(7), 1816-1825.

DeLong, E. F. (2009). The microbial ocean from genomes to biomes. *Nature*, *459*(7244), 200-206.

Deelman, E., Singh, G., Su, M. H., Blythe, J., Gil, Y., Kesselman, C., Mehtaa, G., Vahia, K., Berrimanb, G.B., Goodb, J., Laityb, A., Jacob, J. C., et Katz, D. S. (2005) Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, *13*(3), 219-237.

Delsuc, F., Brinkmann, H., et Philippe, H. (2005). Phylogenomics and the reconstruction of the tree of life. *Nature Reviews Genetics*, *6*(5), 361-375.

Dereeper A, Guignon, V., Blanc, G., Audic, S., Buffet, S., Chevenet, F., Dufayard, J.-F., Guindon, S., Lefort, V., Lescot, M., Claverie, J.-M., et Gascuel, O. (2008). Phylogeny.fr: robust phylogenetic analysis for the non-specialist. *Nucleic Acids Research, 36*(suppl. 2), W465-W469.

Dereeper, A., Audic, S., Claverie, J. M., et Blanc, G. (2010). BLAST-EXPLORER helps you building datasets for phylogenetic analysis. *BMC evolutionary biology*, *10*(1), 8.

Desgraupes, B. (2013). Clustering Indices. University Paris Ouest Lab Modal'X. Récupéré de: cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf.

Dijkman, R., Dumas, M., et García-Bañuelos, L. (2009). Graph matching algorithms for business process model similarity search. Dans *Business process management* (p. 48-63). Springer Berlin Heidelberg. Récupéré de: http://math.ut.ee/~dumas/pubs/bpm09.pdf (vérifé le 15 Août 2014).

Ding, C., et He, X. (2004). K-means clustering via principal component analysis. Dans *Proceedings of the twenty-first international conference on Machine learning* (p. 29). ACM.

Dinov, I. D., Van Horn, J. D., Lozev, K. M., Magsipoc, R., Petrosyan, P., Liu, Z., MacKenzie-Graham, A., Eggert, P., Parker, D. S., et Toga, A. W. (2009). Efficient, distributed and interactive neuroimaging data analysis using the LONI pipeline. *Frontiers in neuroinformatics*, *3*, 22.

Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., Zamanyan, A., Chakrapani, S.,Van Horn, J., Parker, D. S., Magsipoc, R., Leung, K., Gutman, B., Woods, R., et Toga, A. W. (2010). Neuroimaging study designs, computational analyses and data provenance using the LONI pipeline. *PloS One, 5*(9), e13070.

Dinov, I. D., Torri, F., Macciardi, F., Petrosyan, P., Liu, Z., Zamanyan, A., Eggert, P., Pierce, J., Genco, A., Knowles, J., A., Clark, A. P., Van Horn, J. D., Ames, J., Kesselman, C., et Toga, A. W. (2011). Applications of the pipeline environment for visual informatics and genomics computations. *BMC Bioinformatics*, *12*(1), 304.

Dinsdale, E. A., Edwards, R. A., Hall, D., Angly, F., Breitbart, M., Brulc, J.M., Furlan, M., Desnues, C., Haynes, M., Li, L., McDaniel, L., Moran, M. A., Nelson, K. E., Nilsson, C., Olson, R., Paul, J., Brito, B. R., Ruan, Y., Swan, B. K., Stevens, R., Valentine, D. L., Thurber, R. V., Wegley, L., White, B. A., et Rohwer, F. (2008). Functional metagenomic profiling of nine biomes. *Nature*, *452*(7187), 629-632.

Do, C. B., Mahabhashyam, M. S., Brudno, M., et Batzoglou, S. (2005). ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, *15*(2), 330-340.

DonVito, G., Vicario, S., Notarangelo, P., et Balech, B. (2012). The BioVeL Project: Robust phylogenetic workflows running on the GRID. Dans *Proceedings of the EGI Community Forum 2012/EMI Second Technical Conference (EGICF12-EMITC2)*. Munich, Germany (Volume 1,p. 29). Récupéré de: http://pos.sissa.it/archive/conferences/162/029/EGICF12-EMITC2_029.pdf.

Doolittle, W. F. (1999). Phylogenetic classification and the universal tree. *Science*, *284*(5423), 2124-2128.

Drineas, P., Frieze, A. M., Kannan, R., Vempala, S., et Vinay, V. (1999). Clustering in Large Graphs and Matrices. Dans *SODA,*99, 291-299. Récupéré de: http://www.researchgate.net/publication/220780006_Clustering_in_Large_Graphs_and_Matrices/file/60b7d524315e36358d.pdf.

Drummond, A. J., Ashton, B., Cheung, M., Heled, J., Kearse, M., *et al.* (2009). Geneious, v. 5.5; Disponile à l'adresse: http://www.geneious.com/.

Drummond, A. J., et Rambaut, A. (2007). BEAST: Bayesian evolutionary analysis by sampling trees. *BMC evolutionary biology*, *7*(1), 214.

Dubey, G. P., et Ben-Yehuda, S. (2011). Intercellular nanotubes mediate bacterial communication. *Cell*, *144*(4), 590-600.

Duda, R. O., Hart, P.E., et Stork, D. G. (1999), *Pattern classification*. John Wiley & Sons, 680 pages.

Dudley, J. T., et Butte, A. J. (2009). A quick guide for developing effective bioinformatics programming skills. *PLoS computational biology*, *5*(12), e1000589.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, *4*(1), 95-104.

Dunn, C. W., Hejnol, A., Matus, D. Q., Pang, K., Browne, W. E., Smith, S. A., Seaver, E., Rouse, G. W., Obst, M., Edgecombe, G. D., Sørensen, M. V., Haddock, S. H., Schmidt-Rhaesa, A., Okusu. A., Kristensen, R. M., Wheeler, W. C., Martindale, M. Q., et Giribet, G. (2008). Broad phylogenomic sampling improves resolution of the animal tree of life. *Nature*, *452*(7188), 745-749.

Dunn, C. W., Howison, M., et Zapata, F. (2013). Agalma: an automated phylogenomics workflow. *BMC bioinformatics*, *14*(1), 330.

Durbin, R, Eddy, S., Krogh, A., et Mitchison, G. (2006). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.360 pages.

Eddy, S. R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Informatics*, *23*(1), 205-211.

Edgar, R. (2004). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, *5*(1), 113.

Edwards, S. V., Jennings, W. B., et Shedlock, A. M. (2005). Phylogenetics of modern birds in the era of genomics. *Proceedings of the Royal Society B: Biological Sciences*, *272*(1567), 979-992.

Eisen, J. A. (1998). Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome research*, *8*(3), 163-167.

Ester, M., Kriegel, H. P., Sander, J., et Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Dans *Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press, Menlo Park, CA, p. 226-231.

Everitt, B. S., Landau, S., et Leese, M. (2001). Cluster Analysis. Wiley, 5e Édition, London, 346 pages.

Fang, Y., et Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, *56*(3), 468-477.

Farris, J. S. (1977). Phylogenetic analysis under Dollo's Law. *Systematic Biology*, *26*(1), 77-88.

Farris, J. S., Källersjö, M., Kluge, A. G., et Bult, C. (1994). Testing significance of incongruence. *Cladistics*, *10*(3), 315-319.

Fernández, M.-L., et Valiente, G. (2001) A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, *22*(6), 753-758.

Felsenstein, J. (1984). Distance methods for inferring phylogenies: a justification. *Evolution*, 16-24.

Felsenstein, J. (1989). PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, *5* 164-166.

Felsenstein, J. (2004). Inferring phylogenies, 2nd edition, Sinauer Associates, Sunderland, Mass, 664 pages.

Felsenstein, J. (2005, 2006). PHYLIP (phylogeny inference package) distribué par l'auteur (version 3.6), Department of Genome Sciences, University of Washington, Seattle [Software] Available: http://evolution.genetics.washington.edu/phylip.html.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics*, *7*(Part II), 179-188.

Fitch, W. M., et Margoliash, E. (1967). Construction of phylogenetic trees. *Science*, *155*(3760), 279-284.

Fletcher, W., et Yang, Z. (2009). INDELible: a flexible simulator of biological sequence evolution. *Molecular biology and evolution*, *26*(8), 1879-1888.

Forslund, K., Ceric, G., Clements, J., Heger, A., Holm, L., Sonnhammer, E. L., Eddy, S. R., Bateman, A., et Finn, R. D. (2012). The Pfam protein families database. *Nucleic Acids Research*, *40*(Database issue), D290-D301.

Forsyth, R. S. (1990). Neural learning algorithms: Some empirical trials. *Proceedings of the Third Conference on Neural Nets and their Applications* (pp. 301-317). Nanterre, France.

McKenzie, D. P., et Forsyth, R. S. (1995). Classification by similarity: An overview of statistical methods of case-based reasoning. *Computers in Human Behavior*, *11*(2), 273-288.

Fowlkes, E. B., et Mallows, C. L. (1983). A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, *78*(383), 553-569.

Frahling, G., et Sohler, C. (2008). A fast k-means implementation using coresets. *International Journal of Computational Geometry & Applications*, *18*(06), 605-625.

Frickey, T., et Lupas, A. N. (2004). PhyloGenie: automated phylome generation and analysis. *Nucleic Acids Research, 32*(17), 5231-5238.

Friesen, N., et Rüping, S. (2010). Workflow Analysis Using Graph Kernels. Dans *Proceedings of the ECML/PKDD Workshop on Third-Generation Data Mining: Towards Service-Oriented Knowledge Discovery* (SoKD 2010), Barcelona, Spain. Récupéré de: http://www.stefan-rueping.de/publications/friesen-rueping-2010.pdf.

Fry. B. J. (2004). Computational information design. MIT, Boston, thèse de doctorat, Récupéré de: http://benfry.com/phd/.

Garg, B., Puranik, S., Tuteja, N., et Prasad, M. (2012). Abiotic stress-responsive expression of wali1 and wali5 genes from wheat. *Plant signaling & behavior, 7*(11), 1393.

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), 685-695.

Ge, F., Wang, L. S., et Kim, J. (2005). The cobweb of life revealed by genome-scale estimates of horizontal gene transfer. *PLoS biology, 3*(10), e316.

Gehlenborg, N., et Wong, B. (2012). Points of view: Power of the plane. *Nature methods, 9*(10), 935-935.

Giardine, B., Riemer, C., Hardison, R.C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert, I., Taylor, J., Miller, W., Kent, W.J., et Nekrutenko, A. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome Research, 15*(10):1451-1455.

Gil, Y., Ratnakar, V., Kim, J., Gonzalez-Calero, P., Groth, P., Moody, J., et Deelman, E. (2010). Wings: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems*, 62-72.

Gilbert, D. G. (2010). Readseq par D.G. Gilbert, (version. 2.1.30). Récupéré de: http://iubio.bio.indiana.edu/soft/molbio/readseq/java/.

Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., Borkum, M., Bechhofer, S., Roos, M., Li, P., et De Roure, D. (2010). MyExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research, 38* (suppl. 2), W677-W682.

Goderis, A., De Roure, D., Goble, C., Bhagat, J., Cruickshank, D., Fisher, P., Michaelides, D., et Tanoh, F. (2008). Discovering scientific workflows: The myexperiment benchmarks. Récupéré de: http://core.kmi.open.ac.uk/download/pdf/36591.pdf.

Goderis, A., Sattler, U., Lord, P., et Goble, C. (2005). Seven bottlenecks to workflow reuse and repurposing. Dans *The Semantic Web,* (ISWC 2005), (p. 323-337). Springer Berlin Heidelberg.

Goderis, A. (2008). Workflow re-use and discovery in bioinformatics, these de doctorat, University of Manchester, Uk, 223 pages. Récupéré de: http://www.myexperiment.org/files/139/download.

Goderis, A., Li, P., et Goble, C. (2008). Workflow Discovery: Requirements from E-science and a Graph-based Solution. *International Journal of Web Services Research*, *5*(4), 32-58.

Goloboff, P. A. (2014). Oblong, a program to analyse phylogenomic data sets with millions of characters, requiring negligible amounts of RAM. *Cladistics*, *30*(3), 273-281.

Goloboff, P. A., Catalano, S. A., Marcos Mirande, J., Szumik, C. A., Salvador Arias, J., Källersjö, M., et Farris, J.S. (2009). Phylogenetic analysis of 73 060 taxa corroborates major eukaryotic groups. *Cladistics*, *25*(3), 211-230.

Goloboff, P. A., Farris, J. S., et Nixon, K. C. (2008). TNT, a free program for phylogenetic analysis. *Cladistics*, *24*(5), 774-786.

Goodman, A., Pepe, A., Blocker, A. W., Borgman, C. L., Cranmer, K., Crosas, M., Di Stefano, R., Gil, Y., Groth, P., Hedstrom, M., Hogg, D. W., Kashyap, V., Mahabal, A., Siemiginowska, A., et Slavkovic, A. (2014). 10 Simple Rules for the Care and Feeding of Scientific Data. *arXiv preprint arXiv:1401.2134*.

Goto, H., Dickins, B., Afgan, E., Paul, I. M., Taylor, J., Makova, K. D., et Nekrutenko, A. (2011). Dynamics of mitochondrial heteroplasmy in three families investigated via a repeatable re-sequencing study. *Genome Biology*, *12*(6), R59.

Goujon, M., McWilliam, H., Li, W., Valentin, F., Squizzato, S., Paern, J., et Lopez, R. (2010). A new bioinformatics analysis tools framework at EMBL–EBI. *Nucleic acids research*, *38*(suppl. 2), W695-W699.

Guindon, S., et Gascuel, O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Systematic biology*, *52*(5), 696-704.

Guindon, S., Dufayard, J.F., Lefort, V., Anisimova, M., Hordijk, W., et Gascuel, O. (2010). New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Systematic biology*, *59*(3), 307-321.

Günter, S., et Bunke, H. (2002). Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters*, *23*(4), 405-417.

Hayden, E. C. (2014). Is the $1,000 genome for real? *Nature* News, doi:10.1038/nature.2014.14530.

Hallett, M. T., et Lagergren, J. (2001). Efficient algorithms for lateral gene transfer problems. Dans *Proceedings of the fifth annual international conference on Computational biology*, ACM, (p. 149-156).

Hanekamp, K., Bohnebeck, U., Beszteri, B., et Valentin, K. (2007). PhyloGena—a user-friendly system for automated phylogenetic annotation of unknown sequences. *Bioinformatics*, *23*(7), 793-801.

Hartigan, J. A. (1975). Clustering algorithms. Wiley, New York

Hasegawa, M., Kishino, H., et Yano, T. (1985). Dating of human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, *22*(2), 160-174.

Hathaway, R. J., Bezdek, J. C., et Hu, Y. (2000). Generalized fuzzy c-means clustering strategies using L p norm distances. *IEEE Transactions on Fuzzy Systems*, *8*(5), 576-582.

Henikoff, S., et Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences USA*, *89*(22), 10915-10919.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, *52*(1), 258-271.

Hennig, C. (2008). Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods. *Journal of multivariate analysis*, *99*(6), 1154-1176.

Hernandez, P., Martis, M., Dorado, G., Pfeifer, M., Gálvez, S., Schaaf, S., Jouve, N., Šimková, H., Valárik, M., Doležel, J., et Mayer, K. F. (2012). Next-generation sequencing and syntenic integration of flow-sorted arms of wheat chromosome 4A exposes the chromosome structure and gene content. *The Plant Journal*, *69*(3), 377-386.

Hettne, K. M., Wolstencroft, K., Belhajjame, K., Goble, C. A., Mina, E., Dharuri, H., et Roos, M. (2012). Best Practices for Workflow Design: How to Prevent Workflow Decay. in SWAT4LS. Récupéré de: http://ceur-ws.org/Vol-952/paper_23.pdf.

Higgins, D. et Lemey, P. (2009). Multiple sequence alignment. Dans *The Phylogenetic Handbook*, 2nd édition. P. Lemey, M. Salemi et A-M. Vandamme (dir.). Cambridge University Press, UK, 724 pages.

Hils, D. D. (1992). Visual languages and computing survey: Data flow visual programming languages. *Journal of Visual Languages & Computing*, *3*(1), 69-101.

Hinchcliffe, M., et Webster, P. (2011). In silico analysis of the exome for gene discovery. *Methods in Molecular Biology*, *760*, 109-128.

Hipp, D.R., et Kennedy, D. (2003). SQLite. An Embeddable SQL Database Engineer. Récupéré de: http://www.sqlite.org.

Hofacker, I. L. (2009). RNA secondary structure analysis using the Vienna RNA package. *Current Protocols in Bioinformatics*, *12*(2), 1-16.

Holland, R. C., Down, T. A., Pocock, M., Prlić, A., Huen, D., James, K., Foisy, S., Dräger, A., Yates, A., Heuer, M. et Schreiber, M. J. (2008). BioJava: an open-source framework for bioinformatics. *Bioinformatics, 24*(18), 2096-2097.

Hollingsworth, D. (1995). Workflow Management Coalition: The Workflow Reference Model, WFMC-TC-1003, 19-Jan-95. Disponible en ligne à l'adresse http://www.wfmc.org/docs/tc003v11.pdf.

Hong-Tao, B., Li-li, H., Dan-tong, O., Zhan-shan, L., et He, L. (2009). K-means on commodity gpus with CUDA. Dans *WRI World Congress on Computer Science and Information Engineering*, IEEE. (Vol. 3, p. 651-655).

Hoon, S., Ratnapu, K. K., Chia, J. M., Kumarasamy, B., Juguang, X., Clamp, M., Stabenau, A., Potter, S., Clarke, L., et Stupka, E. (2003). Biopipe: a flexible framework for protocol-based bioinformatics analysis. *Genome Research, 13*(8), 1904-1915.

Houde, M., et Diallo, A. O. (2008). Identification of genes and pathways associated with aluminum stress and tolerance using transcriptome profiling of wheat near-isogenic lines. *BMC genomics, 9*(1), 400.

Houde, M., Belcaid, M., Ouellet, F., Danyluk, J., Monroy, A. F., Dryanova, A., Gulick, P., Bergeron, A., Laroche, A., Links, M. G., MacCarthy, L., Crosby, W. L., et Sarhan, F. (2006). Wheat EST resources for functional genomics of abiotic stress. *BMC genomics, 7*(1), 149.

Howison, M., Sinnott-Armstrong, N. A., et Dunn, C. W. (2012). BioLite, a Lightweight Bioinformatics Framework with Automated Tracking of Diagnostics and Provenance. Dans *Theory and Practive of Provenance (TaPP)*. Récupéré de: www.usenix.org/system/files/conference/tapp12/tapp12-final5.pdf.

Huelsenbeck, J. P., Bull, J. J., et Cunningham, C. W. (1996). Combining data in phylogenetic analysis. *Trends in Ecology & Evolution, 11*(4), 152-158.

Huelsenbeck, J. P., et Ronquist, F. (2001). MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics, 17*(8), 754-755.

Ioannidis J. P., Allison, D. B., Ball, C. A., Coulibaly, I., Cui, X., Culhane, A. C., et van Noort, V. (2009). Repeatability of published microarray gene expression analyses. *Nature genetics, 41*(2), 149-155.

Iwasaki, W., et Takagi, T. (2009). Rapid pathway evolution facilitated by horizontal gene transfers across prokaryotic lineages. *PLoS Genetics, 5*(3), e1000402.

Jaccard, P. (1901). Étude comparative de la distribuition florale dans une portion des Alpes et des Jura, *Bulletin de la Société vaudoise des sciences naturelles, 37*, 547-579.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651-666.

Jamieson, D. J., Theiler, R. N., et Rasmussen, S. A. (2006). Emerging infections and pregnancy. *Emerging infectious diseases*, *12*(11), 1638.

Johnson, M., Zaretskaya, I., Raytselis, Y., Merezhuk, Y., McGinnis, S., et Madden, T. L. (2008). NCBI BLAST: a better web interface. *Nucleic Acids Research, 36*(Web Server issue), W5-W9.

Jones, D. T., Taylor, W. R., et Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Computer applications in the biosciences CABIOS*, *8*(3), 275-282.

Jones, M. O., Koutsovoulos, G. D., et Blaxter, M. L. (2011). iPhy: an integrated phylogenetic workbench for supermatrix analyses. *BMC bioinformatics*, *12*(1), 30.

Jones-Rhoades, M. W. et Bartel, D. P. (2004). Computational identification of plant micrornas and their targets, including a stress-induced mirna. *Molecular cell*, *14*(6), 787-799.

Jones-Rhoades, M. W., Bartel, D. P., et Bartel, B. (2006). MicroRNAs and their regulatory roles in plants. *Annual Review of Plant Biology*, *57*, 19-53.

Johnson, R. A., et Wichern, D. W. (1992). *Applied multivariate statistical analysis* (Vol. 4). Englewood Cliffs, NJ: Prentice hall.

Jordan, G. E., et Piel, W. H. (2008). PhyloWidget: web-based visualizations for the tree of life. *Bioinformatics*, *24*(14), 1641-1642.

Jukes, T. H. et Cantor, C. R. (1969). Evolution of Protein Molecules. New York: Academic Press. (p. 21–132).

Jung, J. Y., et Bae, J. (2006). Workflow clustering method based on process similarity. Dans *Computational Science and Its Applications* (ICCSA 2006) (p. 379-389). Springer Berlin Heidelberg.

Kadri, S., Hinman, V., et Benos, P. V. (2009). HHMMiR: efficient de novo prediction of microRNAs using hierarchical hidden Markov models. *BMC bioinformatics*, *10*(suppl. 1), S35.

Karypis, G. 2002. CLUTO a clustering toolkit. [Rapport technique 02-017], Département d'informatique, Universityof Minnesota. Récupéré de: http://glaros.dtc.umn.edu/gkhome/views/cluto.

Kastner, M., Saleh, M. W., Wagner, S., Affenzeller, M., et Jacak, W. (2009). Heuristic methods for searching and clustering hierarchical workflows. Dans *Computer Aided Systems Theory* (EUROCAST 2009) (p. 737-744). Springer Berlin Heidelberg.

Katoh, K., Misawa, K., Kuma, K. I., et Miyata, T. (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, *30*(14), 3059-3066.

Katoh, K., et Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, *30*(4), 772-780.

Katoh, K., et Toh, H. (2010). Parallelization of the MAFFT multiple sequence alignment program. *Bioinformatics*. *26*(15), 1899-1900.

Kaufman, L. R., et Rousseeuw, P. (1990). Finding groups in data: An introduction to cluster analysis. Hoboken, NJ, John Wiley & Sons Inc

Kelleher, C., et Pausch, R. (2005). Lowering the barriers to Programming : a survey of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR) Surveys*, *7*(2), 83-137.

Kim, J., et Warnow, T. (1999). Tutorial on phylogenetic tree estimation. *Intelligent Systems for Molecular Biology*, Heidelberg. Récupéré de: http://www.cs.utexas.edu/users/tandy/tutorial.pdf.

Kimura, M. (1984). *The neutral theory of molecular evolution*. Cambridge University Press.

Kluge, A. G. (1998). Total evidence or taxonomic congruence: Cladistics or consensus classication. *Cladistics*, *14*(2), 151-158.

Kosakovsky Pond, S. K., Wadhawan, S., Chiaromonte, F., Ananda, G., Chung, W. Y., Taylor, J., et Nekrutenko, A. (2009). Windshield splatter analysis with the Galaxy metagenomic pipeline. *Genome research*, *19*(11), 2144-2153.

Kosakovsky Pond, S. L., et Frost, S. D. (2005). Not so different after all: a comparison of methods for detecting amino acid sites under selection. *Molecular biology and evolution*, *22*(5), 1208-1222.

Kosakovsky Pond, S.L., Posada, D., Gravenor, M. B., Woelk, C. H., et Frost, S. D. (2006). GARD: a genetic algorithm for recombination detection. *Bioinformatics*, *22*(24), 3096-3098.

Kozomara, A., et Griffiths-Jones, S. (2014). miRBase: annotating high confidence microRNAs using deep sequencing data. *Nucleic acids research*, *42*(D1), D68-D73.

Kumar, S., Skjæveland, Å., Orr, R. J., Enger, P., Ruden, T., Mevik, B. H., Burki, F., Botnen, A., et Shalchian-Tabrizi, K. (2009). AIR: A batch-oriented web program package for construction of supermatrices ready for phylogenomic analyses. *BMC bioinformatics*, *10*(1), 357.

Kunin, V., Goldovsky, L., Darzentas, N., et Ouzounis, C. A. (2005). The net of life: reconstructing the microbial phylogenetic network. *Genome Research*, *15*(7), 954-959.

Kurtoglu, K. Y., Kantar, M., et Budak, H. (2014). New wheat microRNA using whole-genome sequence. *Functional & integrative genomics*, *14*(2), 1-17.

Kwok, Y. K., et Ahmad, I. (1996). Dynamic critical-path scheduling: An effective technique for allocating task graphs to multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, *7*(5), 506-521.

Löytynoja, A., et Goldman, N. (2008). Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, *320*(5883), 1632-1635.

Lamprecht, A-L., Margaria, T., et Steffen, B. (2009a). From Bio-jETI Process Models to Native Code. Dans *14th IEEE International Conference on Engineering of Complex Computer Systems*, IEEE, (p. 95-101).

Lamprecht, A.-L., Margaria, T. et Steffen, B. (2009b). Bio-jETI: a framework for semantics-based service composition. *BMC bioinformatics*, *10*(suppl. 10), S8.

Lanave, C., Preparata, G., Sacone, C., et Serio, G. (1984). A new method for calculating evolutionary substitution rates. *Journal of molecular evolution*, *20*(1), 86-93.

Lange, T., Roth, V., Braun, M. L., et Buhmann, J. M. (2004). Stability-based validation of clustering solutions. *Neural computation*, *16*(6), 1299-1323.

Lange, T., Roth, V., Braun, M. L., et Buhmann, J. M. (2004). Stability-based validation of clustering solutions. *Neural computation*, *16*(6), 1299-1323.

Lapierre, P., Lasek-Nesselquist, E., et Gogarten, J. P. (2014). The impact of HGT on phylogenomic reconstruction methods. *Briefings in bioinformatics*, *15*(1), 79-90.

Lapointe, F. J., et Rissler, L. J. (2005). Congruence, consensus, and the comparative phylogeography of codistributed species in California. *The American Naturalist*, *166*(2), 290-299.

Larkin, M. A., Blackshields, G., Brown, N. P., Chenna, R., McGettigan, P. A., McWilliam, H., Valentin, F.,Wallace, I. M., Wilm, A., Lopez, R., Thompson, J. D., Gibson, T. J., et Higgins, D. G. (2007). Clustal W and Clustal X version 2.0. *Bioinformatics*, *23*(21), 2947-2948.

Lassmann, T., et Sonnhammer, E.L. (2005). Kalign - an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, *6*(1), 298.

Leal, É., Janini, M., et Diaz, R. S. (2007). Selective pressures of human immunodeficiency virus type 1 (HIV-1) during pediatric infection. *Infection, Genetics and Evolution*, *7*(6), 694-707.

Leclercq, M., Diallo, A. B., et Blanchette, M. (2013). Computational prediction of the localization of micrornas within their pre-mirna. *Nucleic acids research*, *41*(15), 7200-7211.

Leclercq, M. (2012). *Identification et caractérisation de microARNs dans les ESTs du blé par des méthodes bioinformatiques*. Mémoire. Montréal (Québec, Canada), Université du Québec à Montréal, Maîtrise en informatique.

Lecointre, G., et Le Guyader, H. (2001). Classification phylogénétique du vivant (3ième éd.). Belin. 562 pages.

Lee, R., Feinbaum, R., et Ambros, V. (1993), The *C.* elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14, *Cell*, *75*(5), 843-854.

Lee, W. J., et Duin, R. P. (2009). A labelled graph based multiple classifier system. Dans *Multiple Classifier Systems* (p. 201-210). Springer Berlin Heidelberg.

Leigh, J. W., Susko, E., Baumgartner, M., et Roger, A. J. (2008). Testing congruence in phylogenomic analysis. *Systematic biology*, *57*(1), 104-115.

Leigh, J. W., Lapointe, F. J., Lopez, P., et Bapteste, E. (2011). Evaluating phylogenetic congruence in the post-genomic era. *Genome biology and evolution*, *3*, 571-587.

Li, K. B. (2003). ClustalW-MPI: ClustalW analysis using distributed and parallel computing. *Bioinformatics*, *19*(12), 1585-1586.

Li, H., et Durbin, R. (2014) *Mapping and Assembly with Quality(MAQ).* disponible à l'adresse: http://maq.sourceforge.net/.

Limaye, B., Banerjee, R., Datta, A., Inamdar, H., Vats, P., Dahale, S., et Joshi, R. (2012). Anvaya: A workflows environment for automated genome analysis. *Journal of bioinformatics and computational biology*, *10*(4). Récupéré de: http://www.worldscientific.com/doi/abs/10.1142/S0219720012500060.

Lin, J., Ho, C., Sadiq, W., et Orlowska, M. E. (2001). On workflow enabled e-learning services. Dans *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, IEEE. (p. 349-352). doi: 10.1109/ICALT.2001.943942.

Lin, C., Lu, S., Fei, X., Chebotko, A., Pai, D., Lai, Z., Fotouhi, F., et Hua, J. (2009). A reference architecture for scientific workflow management systems and the VIEW SOA solution. *IEEE Transactions on Services Computing*, *2*(1), 79-92.

Lin, H., Ma, X., Feng, W., et Samatova, N. F. (2011). Coordinating computation and I/O in massively parallel sequence search. *IEEE Transactions on Parallel and Distributed Systems*, *22*(4), 529-543.

Lipkus, A. H. (1999). A proof of the triangle inequality for the Tanimoto distance. *Journal of Mathematical Chemistry*, *26*(1-3), 263-265.

Liu, F. G., Miyamoto, M. M., Freire, N. P., Ong, P. Q., Tennant, M. R., Young, T. S., et Gugel, K. F. (2001). Molecular and morphological supertrees for eutherian (placental) mammals. *Science*, *291*(5509), 1786-1789.

Liu, K., Raghavan, S., Nelesen, S., Linder, C. R, et T. Warnow. (2009a). Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, *324*(5934), 1561-1564.

Liu, Y., Maskell, D. L., et Schmidt, B. (2009b). CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units. *BMC research notes*, *2*(1), 73.

Liu, Y., Schmidt, B., et Maskell, D. L. (2010a). MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities. *Bioinformatics*, *26*(16), 1958-1964.

Liu, Y., Schmidt, B., Liu, W., et Maskell, D. L. (2010b). CUDA–MEME: Accelerating motif discovery in biological sequences using CUDA-enabled graphics processing units. *Pattern Recognition Letters*, *31*(14), 2170-2177.

Liu, W., Schmidt, B., et Müller-Wittig, W. (2011a). CUDA-BLASTP: accelerating BLASTP on CUDA-enabled graphics hardware. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *8*(6), 1678-1684.

Liu, W., Schmidt, B., Liu, Y., Voss, G., et Müller-Wittig, W. (2011b). Mapping of BLASTP Algorithm onto GPU Clusters. Dans *IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE. (p. 236-243).

Liu, Y., Schmidt, B., et Maskell, D. L. (2011c). An ultrafast scalable many-core motif discovery algorithm for multiple GPUs. Dans *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, IEEE. (p. 428-434).

Lloyd, S. (1957, publié 1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129-137.

Lord, E., Leclercq, M., Boc, A., Diallo, A.B., et Makarenkov, V. (2012). Armadillo 1.1: an original workflow platform for designing and conducting phylogenetic analysis and simulations. *PloS One, 7*(1), e 29903.

Lord, E., Diallo, A. B., et Makarenkov, V. (2015). Classification of bioinformatics workflows using weighted versions of partitioning and hierarchical clustering algorithms. *BMC Bioinformatics*, *16*(1), 68.

Lord, E., Remita, M. A., Agharbaoui, Z., Leclercq, M., Badawi, M. A., Makarenkov, V., Sarhan, F., et Diallo, A. B. (2015b). WMP: Wheat MiRNA web-Portal. A novel comprehensive wheat miRNA database, including related bioinformatics software (*soumis*).

Louwagie, J., McCutchan, F. E., Peeters, M., Brennan, T. P., Sanders-Buell, E., Eddy, G. A., van der Groen, G., Fransen, K., Gershy-Damet, G.-M., Deleys, D., et Burke, D. S. (1993). Phylogenetic analysis of gag genes from 70 international HIV-1 isolates provides evidence for multiple genotypes. Aids, 7(6), 769-780.

Lu, Y., Lu, S., Fotouhi, F., Deng, Y., et Brown, S. J. (2004). FGKA: A fast genetic k-means clustering algorithm. Dans *Proceedings of the 2004 ACM symposium on Applied computing* (p. 622-623). ACM.

Ludäscher, B., Weske, M., Mcphillips, T., et Bowers, S. (2009). Scientific Workflows: Business as Usual?. Dans *Proceedings of the 7th International Conference on Business Process Management (BPM '09)*, Umeshwar Dayal, Johann Eder, Jana Koehler, et Hajo A. Reijers (dir.). Springer-Verlag, Berlin, Heidelberg, (p. 31-47).

Luks, E. M.(1982). Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, *25*(1), 42-65.

Lushbough, C. M., Bergman, M. K., Lawrence, C. J., et Jennewein, D. (2008). Implementing bioinformatic workflows within the BioExtract Server. *International journal of computational biology and drug design*, *1*(3), 302-312.

Lushbough, C. M., Jennewein, D. M., et Brendel, V. P. (2011). The BioExtract Server: a web-based bioinformatic workflow platform. *Nucleic acids research, 39*(suppl. 2), W528-W532.

Ma, J., Shaw, E., et Kim, J. (2010). Computational workflows for assessing student learning. Dans *Intelligent Tutoring Systems*. Springer Berlin Heidelberg, (p. 188-197).

McGregor, J. J. (1982). Backtrack search algorithms and the maximal common subgraph problem. *Software: Practice and Experience*, *12*(1), 23-34.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Dans *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. University of California Press, Berkeley, California, pp 281-297.

Maddison, W. P., et Maddison, D. R. (2011). Mesquite: a modular system for evolutionary analysis. (Version 2.75) Récupéré de: http://mesquiteproject.org.

Maddison, W. P. (1997). Gene trees in species trees. *Systematic Biology*, *46*(3), 523-536.

Majithia, S., Shields, M. S., Taylor, I. J. et Wang, I. (2004). Triana: A Graphical Web Service Composition and Execution Toolkit Dans *Proceedings of the IEEE International Conference on Web Services* (ICWS'04), IEEE, (p. 514-524).

Makarenkov, V. (2001). T-REX: reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics*, *17*(7), 664-668.

Makarenkov, V., Boc, A., et Legendre, P. (2014). A New Algorithm for Inferring Hybridization Events Based on the Detection of Horizontal Gene Transfers. Dans *Clusters, Orders, and Trees: Methods and Applications* (p. 273-293). Springer New York.

Makarenkov, V., et Leclerc, B. (2000). Comparison of additive trees using circular orders. *Journal of Computational Biology*, *7*(5), 731-744.

Makarenkov, V., et Legendre, P. (2001). Optimal variable weighting for ultrametric and additive trees and K-means partitioning: Methods and software. *Journal of Classification*, *18*(2), 245-271.

Makarenkov, V., Boc, A., Xie, J., Peres-Neto, P., Lapointe, F. J., et Legendre, P. (2010). Weighted bootstrapping: a correction method for assessing the robustness of phylogenetic trees. *BMC evolutionary biology*, *10*(1), 250.

Manavski, S. A., et Valle, G. (2008). CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. *BMC bioinformatics*, *9*(suppl. 2), S10.

Margaria, T., Nagel, R., et Steffen, B. (2005). Remote integration and coordination of verification tools in jETI. Dans *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, (ECBS'05), IEEE. (p. 431-436).

May, R. M. (1988). How many species are there on earth?. *Science*, *241*(4872), 1441-1449.

McGregor, J. J. (1982). Backtrack search algorithms and the maximal common subgraph problem. *Software: Practice and Experience*, *12*(1), 23-34.

McMahon, M. M., et Sanderson, M. J. (2006). Phylogenetic supermatrix analysis of GenBank sequences from 2228 papilionoid legumes. *Systematic biology*, *55*(5), 818-836.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, *12*(2), 153-157.

Meyers, B. C., Axtell, M. J., Bartel, B., Bartel, D. P., Baulcombe, D., Bowman, J. L., Cao, X., Carrington, J. C., Chen, X., Green, P. J., Griffiths-Jones, S., Jacobsen, S. E., Mallory, A. C., Martienssen, R. A., Poethig, R. S., Qi, Y., Vaucheret, H., Voinnet, O., Watanabe, Y., Weigel, D., Zhu, J. K.(2008). Criteria for annotation of plant microrna. *The Plant Cell Online*, *20*(12), 3186-3190.

Migliorini, S., Gambini, M., La Rosa, M. et ter Hofstede, A. H. M. (2011) Pattern-Based Evaluation of Scientific Workflow Management Systems. BPM Center Report BPM-11-03, BPMcenter.org, 2011. Disponible en ligne à l'adresse http://eprints.qut.edu.au/39935/.

Milligan, G. W., et Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*(2), 159-179.

Milligan, G. W. (1989). A validation study of a variable weighting algorithm for cluster analysis. *Journal of Classification*, 6(1), 53-71.

Mirkin, B. G., Fenner, T. I., Galperin, M. Y., et Koonin, E. V. (2003). Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. *BMC evolutionary biology*, *3*(1), 2.

Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., William, A., Oinn, T., et Goble, C. (2010). Taverna, reloaded. Dans *Scientific and Statistical Database Management*, Springer Berlin Heidelberg. (p. 471-481).

Mora, C., Tittensor, D. P., Adl, S., Simpson, A. G., et Worm, B. (2011). How many species are there on Earth and in the ocean?. *PLoS biology*, *9*(8), e1001127.

Nadeem, F., et Fahringer, T. (2009). Using templates to predict execution time of scientific workflow applications in the grid. Dans *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, IEEE Computer Society. (p. 316-323).

Nei, M., et Gojobori, T. (1986). Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Molecular biology and evolution*, *3*(5), 418-426.

Nevill, P. G., Wallace, M. J., Miller, J. T., et Krauss, S. L. (2013). DNA barcoding for conservation, seed banking and ecological restoration of Acacia in the Midwest of Western Australia. *Molecular Ecology Resources*. *13*(6), 1033-1042.

Ng, R. T., et Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, *14*(5), 1003-1016.

Nozawa, M., Miura, S., et Nei, M. (2012). Origins and evolution of microRNA genes in plant species. *Genome biology and evolution*, *4*(3), 230-239.

Oakley, T. H., Alexandrou, M. A., Ngo, R., Pankey, M. S., Churchill, C. K., Chen, W., et Lopker, K. B. (2014). Osiris: accessible and reproducible phylogenetic and phylogenomic analyses within the Galaxy workflow management system. *BMC bioinformatics*, *15*(1), 230.

Oinn, T., Greenwood, M., Addis, M., Alpdemir, M. N., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M. R., Senger, M., Stevens, R., Wipat, A., et Wroe, C. (2006). Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, *18*(10), 1067-1100.

Oinn, T., Li, P., Kell, D. B., Goble, C., Goderis, A., Greenwood, M., Hull, D., Stevens, R., Turl, D., et Zhao, J. (2007). Taverna/myGrid: aligning a workflow system with the life sciences community. Dans *Workflows for e-Science* (p. 300-319). Springer London.

Okonechnikov, K., Golosova, O., et Fursov, M. (2012). Unipro UGENE: a unified bioinformatics toolkit. *Bioinformatics*, *28*(8), 1166-1167.

Olsen, G. J., Matsuda, H., Hagstrom, R., et Overbeek, R. (1994). fastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Computer applications in the biosciences*, *10*(1), 41-48.

Orvis, J., Crabtree, J., Galens, K., Gussman, A., Inman, J. M., Lee, E., Nampally, S., Riley, D., Sundatam, J. P., Felix, V., Whitty, B., Mahurkar, A., Wortman, J., White, O., et Angiuoli, S. V. (2010). Ergatis: a web interface and scalable software system for bioinformatics workflows. *Bioinformatics*, *26*(12), 1488-1492.

Ossowski, S., Schwab, R., et Weigel, D. (2008). Gene silencing in plants using artificial microRNAs and other small RNAs. *The Plant Journal*, *53*(4), 674-690.

Pagani, I., Liolios, K., Jansson, J., Chen, I. M., Smirnova, T., Nosrat, B., Markowitz, V. M., et Kyrpides, N. C. (2012). The Genomes OnLine Database (GOLD) v.4: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, *40*(Database issue), D571-D579.

Pagel, M. (1999). Inferring the historical patterns of biological evolution. *Nature*, *401*(6756), 877-884.

Parastatidis, S. (2009). A Platform for All That We Know: Creating a Knowledge Driven Research Infrastructure. Dans *The fourth paradigm: data-intensive scientific discovery*. Microsoft Research Redmond, WA, p. 165-172.

Pardi, F., Guillemot, S., et Gascuel, O. (2010). Robustness of phylogenetic inference based on minimum evolution. *Bulletin of mathematical biology*, *72*(7), 1820-1839.

Pattengale, N. D., Gottlieb, E. J., et Moret, B. M. (2007). Efficiently computing the Robinson-Foulds metric. *Journal of Computational Biology*, *14*(6), 724-735.

Peters, R. S., Meyer, B., Krogmann, L., Borner, J., Meusemann, K., Schütte, K., Niehuis, O., et Misof, B. (2011). The taming of an impossible child: a standardized all-in approach to the phylogeny of Hymenoptera using public database sequences. *BMC biology*, *9*(1), 55.

Pfeiffer, W. et Stamatakis, A. (2010). Hybrid MPI/Pthreads parallelization of the RAxML phylogenetics code. Dans *IEEE International Symposium on Parallel & Distributed Processing, Workshops and and Forum*,(IPDPSW 2010), IEEE, p. 1-8.

Philippe, H., Brinkmann, H., Lavrov, D. V., Littlewood, D. T., Manuel, M., Wörheide, G., et Baurain, D. (2011). Resolving difficult phylogenetic questions: why more sequences are not enough. *PLoS Biology*, *9*(3), e1000602.

Philippe, H., Zhou, Y., Brinkmann, H., Rodrigue, N., et Delsuc, F. (2005). Heterotachy and long-branch attraction in phylogenetics. *BMC evolutionary biology*, *5*(1), 50.

Pisani, D., Benton, M. J., et Wilkinson, M. (2007). Congruence of morphological and molecular phylogenies. *Acta biotheoretica*, *55*(3), 269-281.

Pop, M., et Salzberg, S. L. (2008). Bioinformatics challenges of new sequencing technology. *Trends in Genetics*, *24*(3), 142-149.

Popa, O., et Dagan, T. (2011). Trends and barriers to lateral gene transfer in prokaryotes. *Current opinion in microbiology*, *14*(5), 615-623.

Posada, D. (2008). jModelTest: phylogenetic model averaging. *Molecular biology and evolution*, *25*(7), 1253-1256.

Price, M. N., Dehal, P. S., et Arkin, A. P. (2010). FastTree 2–approximately maximum-likelihood trees for large alignments. *PloS One*, *5*(3), e9490.

Punta, M., Coggill, P.C., Eberhardt, R. Y., Mistry, J., Tate, J., Boursnell, C., Pang, N., Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., et Glöckner, F. O. (2013). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. Opens external link in new window. *Nucleic Acids Research*, *41*(D1), D590-D596.

Rahman, M., Hassan, R., Ranjan, R., et Buyya, R. (2013). Adaptive workflow scheduling for dynamic grid and cloud computing environment. *Concurrency and Computation: Practice and Experience*, *25*(13), 1816-1842.

Ramakrishnan, L., et Gannon, D. (2008). *A survey of distributed workflow characteristics and resource requirements*. Indiana University. Récupéré de: http://www.cs.indiana.edu/l/www/ftp/techreports/TR671.pdf.

Rambaut, A., et Grass, N. C. (1997). Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Computer applications in the biosciences*, *13*(3), 235-238.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*(336), 846-850.

Ransy, D. G., Lord, E., Caty, M., Lapointe, N., Boucher, M., Diallo, A. B., Soudeyns, H. (2015). Subtle Differences in Selective Pressures Applied on the Envelope Gene of HIV-1 in Pregnant Versus Non-Pregnant Women. (soumis).

Rannala, B., et Yang, Z. (1996). Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *Journal of molecular evolution*, *43*(3), 304-311.

Ray, D. K., Mueller, N. D., West, P. C., et Foley, J. A. (2013). Yield trends are insufficient to double global crop production by 2050. *PLoS One*, *8*(6), e66428.

Raymond, J. W., et Willett, P. (2002). Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of computer-aided molecular design*, *16*(7), 521-533.

Ren, F., Tanaka, H., et Yang, Z. (2009). A likelihood look at the supermatrix-supertree controversy. *Gene*, *441*(1), 119-125.

Rex, D. E., Ma, J. Q., et Toga, A. W. (2003). The LONI pipeline processing environment. *Neuroimage*, *19*(3), 1033-1048.

Reynolds, A.P., Richards, G., et Rayward-Smith, V. J. (2004). The application of k-medoids and pam to the clustering of rules. Dans *Intelligent Data Engineering and Automated Learning,* (IDEAL 2004), (p. 173-178). Springer Berlin Heidelberg

Richards, K. D., Snowden, K. C., et Gardner, R. C. (1994). Wali6 and wali7. Genes induced by aluminum in wheat (Triticum aestivum L.) roots. Plant *physiology*, *105*(4), 1455.

Riesen, K., et Bunke, H. (2009). Graph classification based on vector space embedding. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(06), 1053-1081.

Rivas, E., et Eddy, S. R. (2008). Probabilistic phylogenetic inference with insertions and deletions. *PLoS computational biology*, 4(9), e1000172.

Robbertse, B., Yoder, R. J., Boyd, A., Reeves, J., et Spatafora, J. W. (2011). Hal: an automated pipeline for phylogenetic analyses of genomic data. *PLoS currents*, *3*, RRN1213.

Robinson, D. F., et Foulds, L. R. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences*, *53*(1), 131-147.

Robinson, D. M., Jones, D. T., Kishino, H., Goldman, N., et Thorne, J. L. (2003). Protein evolution with dependence among codons due to tertiary structure. *Molecular Biology and Evolution*, *20*(10), 1692-1704.

Robles-Kelly, A., et Hancock, E. R. (2005). Graph edit distance from spectral seriation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(3), 365-378.

Rogers, K., et Xuemei, C. (2013). Biogenesis, turnover, and mode of action of plant microRNAs. *The Plant Cell Online*, *25*(7), 2383-2399.

Rokas, A., et Holland, P. W. (2000). Rare genomic changes as a tool for phylogenetics. *Trends in Ecology & Evolution*, *15*(11), 454-459..

Romano, P. (2008). Automation of in-silico data analysis processes through workflow management systems. *Briefings in Bioinformatics*, *9*(1), 57-68.

Romero-Severson, E., Skar, H., Bulla, I., Albert, J., et Leitner, T. (2014). Timing and order of transmission events is not directly reflected in a pathogen phylogeny. *Molecular biology and evolution*, *31*(9), 2472-2482.

Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., Huelsenbeck, J. P. (2012). MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Systematic biology*, *61*(3), 539-542.

Ronquist, F., van der Mark, P., et Huelsenbeck, J. P. (2009). Bayesian phylogenetic analysis using MrBayes. Dans *The phylogenetic handbook*. Cambridget University Press. (p. 210-266).

Ropelewski, A. J., Nicholas, H. B., et Gonzalez Mendez, R. R. (2010). MPI-PHYLIP: parallelizing computationally intensive phylogenetic analysis routines for the analysis of large protein families. *PLoS One*, *5*(11), e13999.

Roure, B., Rodriguez-Ezpeleta, N., et Philippe, H. (2007). SCaFoS: a tool for selection, concatenation and fusion of sequences for phylogenomics. *BMC evolutionary biology*, *7*(suppl. 1), S2.

Roure, B., et Philippe, H. (2011). Site-specific time heterogeneity of the substitution process and its impact on phylogenetic inference. *BMC evolutionary biology*, *11*(1), 17.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53-65.

Rumble, S. M., Lacroute, P., Dalca, A. V., Fiume, M., Sidow, A., et Brudno, M. (2009). SHRiMP: accurate mapping of short color-space reads. *PLoS computational biology*, *5*(5), e1000386.

Sanfeliu, A., et Fu, K. S. (1983). A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, *13*(3), 353-362.

Saitou, N., et Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, *4*(4), 406-425.

Sánchez, R., Serra, F., Tárraga, J., Medina, I., Carbonell, J., Pulido, L., de María, A., Capella-Gutíerrez, S., Huerta-Cepas, J., Gabaldón, T., Dopazo, J., et Dopazon, H. (2011). Phylemon 2.0: a suite of web-tools for molecular evolution, phylogenetics, phylogenomics and hypotheses testing. *Nucleic acids research*, *39*(suppl. 2), W470-W474.

Sanderson, M. J., Donoghue, M. J., Piel, W. H., et Eriksson, T. (1994). TreeBASE: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life. *American Journal of Botany*, *81*(6), 183.

Santos, E., Lins, L., Ahrens, J. P., Freire, J., et Silva, C. T. (2008). A first study on clustering collections of workflow graphs. Dans *Provenance and Annotation of Data and Processes* (p. 160-173). Springer Berlin Heidelberg.

Savolainen, V., et Chase, M. W. (2003). A decade of progress in plant molecular phylogenetics. *Trends in Genetics*, *19*(12), 717-724.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, *1*(1), 27-64.

Schatz, M. C. (2010). The missing graphical user interface for genomics. *Genome Biology*, *11*(8), 128.

Schmidt, H. A., Strimmer, K., Vingron, M., von Haeseler, A. (2002). TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics*, *18*(3), 502-504.

Schmollinger, M., Nieselt, K., Kaufmann, M., et Morgenstern, B. (2004). DIALIGN P: Fast pair-wise and multiple sequence alignment using parallel processors. *BMC bioinformatics*, *5*(1), 128.

Seibel, P. N., Krüger, J., Hartmeier, S., Schwarzer, K., Löwenthal, K., Mersch, H., Dandekar, T. et Giegerich, R. (2006). XML schemas for common bioinformatic data types and their application in workflow systems. *BMC bioinformatics*, *7*(1), 490.

Sharma, A., Rai, A., et Lal, S. (2013). Workflow management systems for gene sequence analysis and evolutionary studies - A Review. *Bioinformation*, *9*(13), 663-672.

Silva, V., Chirigati, F., Maia, K., Ogasawara, E., Oliveira, D., Braganholo, V., Murta, L., et Mattoso, M. (2011). Similarity-based workflow clustering. *Journal of Computational Interdisciplinary Sciences*, *2*(1), 23-35.

Simonsen, M., Mailund, T., et Pedersen, C. N. (2008). Rapid neighbour-joining. Dans *Algorithms in Bioinformatics* (p. 113-122). Springer Berlin Heidelberg.

Singh, G., Su, M. H., Vahi, K., Deelman, E., Berriman, B., Good, J., Katz, D. S., et Mehta, G. (2008). Workflow task clustering for best effort systems with Pegasus. Dans *Proceedings of the 15th ACM Mardi Gras conference: From lightweight mash-ups to lambda grids: Understanding the spectrum of distributed computing requirements, applications, tools, infrastructures, interoperability, and the incremental adoption of key capabilities* ACM. P.9.

Smillie, C. S., Smith, M. B., Friedman, J., Cordero, O. X., David, L. A., et Alm, E. J. (2011). Ecology drives a global network of gene exchange connecting the human microbiome. *Nature*, *480*(7376), 241-244.

Smith, S.A., et Donoghue, M. J. (2008). Rates of molecular evolution linked to life history in flowering plants. *Science*, *322*(5898), 86-89.

Smith, S. A., Beaulieu, J. M., et Donoghue, M. J. (2009). Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. *BMC evolutionary biology*, *9*(1), 37.

Smith, W., Taylor, V., et Foster, I. (1999). Using run-time predictions to estimate queue wait times and improve scheduler performance. Dans *Job Scheduling Strategies for Parallel Processing* (p. 202-219). Springer Berlin Heidelberg.

Snowden, K. C, et Richard, C. G. (1993). Five genes induced by aluminum in wheat (Triticum aestivum L.) roots. *Plant Physiology*, *103*(3), 855-861.

Sokal, R., et Michener, C. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38(issue), 1409-1438.

Sorek, R., Zhu, Y., Creevey, C. J., Francino, M. P., Bork, P., et Rubin, E. M. (2007). Genome-wide experimental determination of barriers to horizontal gene transfer. *Science*, *318*(5855), 1449-1452.

Sroka, J., Hidders, J., Missier, P., et Goble, C. (2009). A formal semantics for the Taverna 2 workflow model, *Journal of Computer and System Sciences*, 76(issue), 490-508.

Sroka, J., Krupa, Ł., Kierzek, A. M., et Tyszkiewicz, J. (2011). CalcTav—integration of a spreadsheet and Taverna workbench. *Bioinformatics*, *27*(18), 2618-2619.

Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G., Korf, I., Lapp, H., Lehväslaiho, H., Matsalla, C., Mungall, C. J., Osborne, B. I., Pocock, M. R., Schattner, P., Senger, M., Stein, L. D., Stupka, E., Wilkinson, M. D., et Birney, E. (2002). The Bioperl toolkit: Perl modules for the life sciences. *Genome research*, *12*(10), 1611-1618.

Stamatakis, A. (2006). RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, *22*(21), 2688-2690.

Stamatakis, A., Blagojevic, F., Nikolopoulos, D. S., et Antonopoulos, C. D. (2007). Exploring new search algorithms and hardware for phylogenetics: RAxML meets the IBM cell. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, *48*(3), 271-286.

Steinley, D. (2003). Local optima in K-means clustering: what you don't know may hurt you. *Psychological methods, 8*(3) 294-304.

Steinley, D. (2006). Profiling local optima in K-means clustering: Developing a diagnostic technique. *Psychological methods*, *11*(2), 178-192.

Steinley, D. (2008). Stability analysis in *K*-means clustering. *British Journal of Mathematical and Statistical Psychology*, *61*(2), 255-273.

Stevens, R., Zhao, J., et Goble, C. (2007). Using provenance to manage knowledge of in silico experiments. *Briefings in bioinformatics*, *8*(3), 183-194.

Stevens, R., Goble, C., Baker, P., et Brass, A. (2001). A classification of tasks in bioinformatics. *Bioinformatics*, *17*(2), 180-188.

Stevens, R. D., Robinson, A. J., et Goble, C. A. (2003). myGrid: personalised bioinformatics on the information grid. *Bioinformatics*, *19*(suppl. 1), i302-i304.

Strimmer, K., et Moulton, V. (2000). Likelihood analysis of phylogenetic networks using directed graphical models. *Molecular Biology and Evolution*, *17*(6), 875-881.

Stewart, C. A., Hart, D., Berry, D. K., Olsen, G. J., Wernert, E. A., et Fischer, W. (2001). Parallel implementation and performance of fastDNAml—a program for maximum likelihood phylogenetic inference. Dans *ACM/IEEE 2001 Conference on Supercomputing*, (p. 32-32).

Subramanian, A. R., Kaufmann, M., et Morgenstern, B. (2008). DIALIGN-TX: greedy and progressive approaches for segment-based multiple sequence alignment. *Algorithms in Molecular Biology*, *3*,6.

Suchard, M. A., et Redelings, B. D. (2006). BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics*, *22*(16), 2047-2048.

Sunkar, R., et Zhu, J. K. (2004). Novel and stress-regulated microRNAs and other small RNAs from Arabidopsis. *The Plant Cell Online*, *16*(8), 2001-2019.

Swofford, D. L., Olsen, G. J., Waddell, P. J., et Hillis, D. M. (1996). Phylogenetic Inference. Dans *Molecular Systematics*, Hiillis DM, Moritz D, and Mable BK, editors, Sinauer Associates, Sunderland, Massachusetts., p. 407-514.

Swofford, D. L. (2002). PAUP*: phylogenetic analysis using parsimony (* and other methods). Version 4, Massachusetts: Sinauer Associates, Sunderland, Février. 2002.

Talavera, G., et Castresana, J. (2007). Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Systematic biology*, *56*(4), 564-577.

Tamura, K., Peterson, D., Peterson, N., Stecher, G., Nei, M., et Kumar, S. (2011). MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. *Molecular biology and evolution*, *28*(10), 2731-2739.

Tan, G., Feng, S., et Sun, N. (2005, July). Parallel multiple sequences alignment in SMP cluster. Dans *Proceedings. Eighth International Conference on High-Performance Computing in Asia-Pacific Region*, IEEE. (p. 431-437).

Tan, W., Zhang, J., et Foster, I. (2010). Network Analysis of scientific workflows : a gateway to reuse. *IEEE Computer*, *43*(10), 54-61.

Tanimoto, T. T. (1957). An Elementary Mathematical theory of Classification and Prediction. *Internal IBM Technical Report*.

Tang, Z., Choi, J. H., Hemmerich, C., Sarangi, A., Colbourne, J. K., et Dong, Q. (2009). ESTPiper–a web-based analysis pipeline for expressed sequence tags. *BMC genomics*, *10*(1), 174.

Taylor, I. J. et Schutz, B.F. (1997). The Grid Signal Processing System. Dans *Astronomical Data Analysis Software and Systems VI, ASP Conference Series*, *125*, 18-21.

Taylor, I., Shields, M., Wang, I., et Harrison, A. (2007). The triana workflow environment: Architecture and applications. Dans *Workflows for e-Science* (p. 320-339). Springer London.

Taylor, R. C. (2010). An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC bioinformatics*, *11*(suppl. 12), S1.

Than, C., Ruths, D., et Nakhleh, L. (2008). PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC bioinformatics*, *9*(1), 322.

Tsai, Y. L, Huang, K. C., Chang, H. Y., Ko, J., Wang, E. T., et Hsu, C. H. (2012). Scheduling Multiple Scientific and Engineering Workflows through Task Clustering and Best-Fit Allocation. Dans *IEEE Eighth World Congress on Services*: 24-29 Juin 2012, Honolulu, HI. Los Alamitos, IEEE Computer Society, 1-8.

Thompson, K. (1968). Programming techniques: Regular expression search algorithm. *Communications of the ACM*, *11*(6), 419-422.

Thompson, J. D., Higgins, D. G., et Gibson, T. J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, *22*(22), 4673-4680.

Turi, D., Missier, P., Goble, C., De Roure, D., et Oinn, T. (2007, December). Taverna workflows: Syntax and semantics. Dans *IEEE International Conference on e-Science and Grid Computing*, IEEE, (p. 441-448).

Turner, K. J., et Lambert, P. S. (2014). Workflows for quantitative data analysis in the social sciences. *International Journal on Software Tools for Technology Transfer*, 1-18.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327-352.

Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM*, *23*(1), 31-42.

van Nimwegen, E. (2007). Finding regulatory elements and regulatory motifs: a general probabilistic framework. *BMC Bioinformatics*, *8*(suppl. 6), S4.

van der Aalst, W. M. P., et van Hee, K. (2002). *Workflow managment : Models, Methods, and Systems*. MIT Press, Cambridge, Massachussetts, 368 pages.

van der Aalst, W. M. P., et Stahl, C. (2011). *Modeling Business Processes, a petri net-oriented approach*. MIT Press, Cambridge, Massachussetts . 386 pages.

van der Veen, J., Jones, V., et Collis, B. (2000). Using workflow for projects in higher education. *Computer Science Education*, *10*(3), 283-301.

van Rijsbergen. C. J. (1979). *Information Retrieval* (2^{e} éd.). Butterworth-Heinemann, Newton, MA, USA.

von Haeseler, A., et Churchill, .G. A. (1993). Network models for sequence evolution. *Journal of Molecular Evolution*, *37*(1), 77-85.

Vairavanathan, E., Al-Kiswany, S., Costa, L. B., Zhang, Z., Katz, D. S., Wilde, M., et Ripeanu, M. (2012). A workflow-aware storage system: An opportunity study. Dans *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (ccgrid 2012) (p. 326-334). IEEE Computer Society.

Valentin, F., Squizzato, S., Goujon, M., McWilliam, H., Paern, J., et Lopez, R. (2010). Fast and efficient searching of biological data resources—using EB-eye. *Briefings in bioinformatics*, *11*(4), 375-384.

Vesanto, J., et Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, *11*(3), 586-600.

Vicari, D., et Vichi, M. (2009). Structural classification analysis of three-way dissimilarity data. *Journal of Classification*, *26*(2), 121-154.

Vicario, S., Hardisty, A., et Haitas, N. (2011). Biovel: biodiversity virtual e-laboratory. *EMBnet journal*, *17*(2), 5.

Vouk, M. A., Bitzer, D., et Klevans, R. L. (1999). Workflow and end-user quality of service issues in web-based education. *IEEE Transactions on Knowledge and Data Engineering*, *11*(4), 673-687.

Wagstaff, K., Cardie, C., Rogers, S., et Schrödl, S. (2001). Constrained k-means clustering with background knowledge. Dans *International Conference on Machine Learning* (Vol. 1, p. 577-584). Récupéré de: https://web.cse.msu.edu/~cse802/notes/ConstrainedKmeans.pdf.

Wallis, W. D., Shoubridge, P., Kraetz, M., et Ray, D. (2001). Graph distances using graph union. *Pattern Recognition Letters*, *22*(6), 701-704.

Walters, J. P., Balu, V., Kompalli, S., et Chaudhary, V. (2009). Evaluating the use of GPUs in liver image segmentation and HMMER database searches. Dans *IEEE International Symposium on Parallel & Distributed Processing*, 2009. (IPDPS 2009). IEEE. (p. 1-12).

Walters, J. P., Meng, X., Chaudhary, V., Oliver, T., Yeow, L. Y., Schmidt, B., Nathan, D., et Landman, J. (2007). MPI-HMMER-boost: distributed FPGA acceleration. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, *48*(3), 223-238.

Weerawarana, S., Curbera, F., Leymann, F., Storey, T., et Ferguson, D. F. (2005). *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR. 456 pages.

Whelan, S., et Goldman, N. (2001). A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Molecular biology and evolution*, *18*(5), 691-699.

Wheeler, T. J. (2009). Large-scale neighbor-joining with NINJA. Dans *Algorithms in Bioinformatics* (p. 375-389). Springer Berlin Heidelberg.

Wieczorek, M., Prodan, R., et Fahringer, T. (2005). Scheduling of scientific workflows in the ASKALON grid environment. *ACM SIGMOD Record*, *34*(3), 56-62.

Wieczorek, M., Hoheisel, A., et Prodan, R. (2009). Towards a general model of the multi-criteria workflow scheduling on the grid. *Future Generation Computer Systems*, *25*(3), 237-256.

Wilk, M. B., et Gnanadesikan, R. (1968). Probability plotting methods for the analysis for the analysis of data. *Biometrika*, *55*(1), 1-17.

Wilkens, S. J., Janes, J., et Su, A. I. (2005). HierS: hierarchical scaffold clustering using topological chemical graphs. *Journal of medicinal chemistry*,*48*(9), 3182-3193.

Witten, D. M., et Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, *105*(490), 713–726.

Wollrath, A., Riggs, R., et Waldo, J. (1996). A Distributed Object Model for the Java™ System. *Computing Systems*, *9*, 265-290.

Wombacher, A. (2006). Evaluation of technical measures for workflow similarity based on a pilot study. Dans *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE* (p. 255-272). Springer Berlin Heidelberg.

Wombacher, A., et Li, C. (2010). Alternative approaches for workflow similarity. Dans *IEEE International Conference on Services Computing* (SCC), IEEE, (p. 337-345).

Wong, K. M., Suchard, M. A., et Huelsenbeck, J. P. (2008). Alignment uncertainty and genomic analysis. *Science*, *319*(5862), 473-478.

Woollard, D., Medvidovic, N., Gil, Y., et Mattmann, C. A.. (2008). Scientific Software as Workflows: From Discovery to Distribution. *IEEE Software*, *25*(4), 37-43.

Woollard, P. M. (2010). Asking complex questions of the genome without programming. Dans *Genetic Variation* (p. 39-52). Humana Press.

World Health Organization, UNAIDS, UNICEF. 2011. Global HIV/AIDS response, Epidemic update and health sector progress towards Universal Access.

Wu, X., Yang, Z. Y., Li, Y., Hogerkorp, C. M., Schief, W. R., Seaman, M. S., et Mascola, J. R. (2010). Rational design of envelope identifies broadly neutralizing human monoclonal antibodies to HIV-1. *Science*, *329*(5993), 856-861.

Xu, R., et Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, *16*(3), 645-678.

Yamakawa, T., Horio, K., et Hoshino, M. (2006). Self-organizing map with input data represented as graph. Dans *Neural Information Processing* (p. 907-914). Springer Berlin Heidelberg.

Yang, Z. (2007). PAML 4: phylogenetic analysis by maximum likelihood. *Molecular biology and evolution*, *24*(8), 1586-1591.

Yang, Z., Nielsen, R., Goldman, N., et Pedersen, A. M. K. (2000). Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics*, *155*(1), 431-449.

Yang, Z., Nielsen, R., et Hasegawa, M. (1998). Models of amino acid substitution and applications to mitochondrial protein evolution. *Molecular Biology and Evolution*, *15*(12), 1600-1611.

Yang, Z., Wong, W. S., et Nielsen, R. (2005). Bayes empirical bayes inference of amino acid sites under positive selection. *Molecular Biology and Evolution, 22*(4), 1107-1118.

Yuan, D., Yang, Y., Liu, X., et Chen, J. (2010). A data placement strategy in scientific cloud workflows. *Future Generation Computer Systems*, *26*(8), 1200-1214.

Zeng, Q. Y., Yang, C. Y., Ma, Q. B., Li, X. P., Dong, W. W., et Nian, H. (2012). Identification of wild soybean miRNAs and their target genes responsive to aluminum stress. *BMC plant biology*, *12*(1), 182.

Zeng, Z., Tung, A. K., Wang, J., Feng, J., et Zhou, L. (2009). Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment*, *2*(1), 25-36.

Zhang, F. E., et Li, K. (2009). Application of Workflow Technology in Graduate Education Management. Dans *EEEE'09. International Conference on E-Learning, E-Business, Enterprise Information Systems, and E-Government*, IEEE. (p. 263-266).

Zhang, J., Chiodini, R., Badr, A., et Zhang, G. (2011). The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics*, *38*(3), 95-109.

Zhang, Q., et Couloigner, I. (2005). A new and efficient k-medoid algorithm for spatial clustering. Dans *Computational Science and Its Applications–ICCSA 2005* (p. 181-189). Springer Berlin Heidelberg.

Zhang, Z., Li, J., Zhao, X. Q., Wang, J., Wong, G. K. S., et Yu, J. (2006). KaKs_Calculator: calculating Ka and Ks through model selection and model averaging. *Genomics, proteomics & bioinformatics*, *4*(4), 259-263.

Zhao, Y., Karypis, G., et Fayyad, U. (2005).Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery, 10*(2),141-168.

Zhao, H., et Sakellariou, R. (2003). An experimental investigation into the rank function of the heterogeneous earliest finish time scheduling algorithm. Dans *Euro-Par 2003 Parallel Processing* (p. 189-194). Springer Berlin Heidelberg.

Zhao, J., Gomez-Perez, J. M., Belhajjame, K., Klyne, G., Garcia-Cuesta, E., Garrido, A., Hettne, K., Roos, M., De Roure, D., et Goble, C. (2012). Why workflows break—Understanding and combating decay in Taverna workflows. Dans *IEEE 8th International Conference on E-Science (e-Science)*, IEEE. (p. 1-9).

Zmasek, C. M., et Eddy, S. R. (2001). ATV: display and manipulation of annotated phylogenetic trees. *Bioinformatics*, *17*(4), 383-384.

Zwickl, D. J. (2006). Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. Thèse de doctorat, L'Université du Texas à Austin. Récupéré de: http://molevol.lysine.umiacs.umd.edu/molevolfiles/garli/zwicklDissertation.pdf.