UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MÉTHODES ET LOGICIEL POUR LE TRAITEMENT EFFICACE DES DONNÉES DE CRIBLAGE À HAUT DÉBIT

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR
PABLO ZENTILLI

SEPTEMBRE 2007

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche, M. Vladimir Makarenkov, pour ses conseils, ses suggestions, ses encouragements et son suivi.

Je remercie également mes collègues, Alix Boc, Abdoulaye Baniré Diallo, Alpha Boubacar Diallo, Andrei Gagarin et Dmytro Kevorkov pour leurs conseils, leur aide et leur soutien qui m'ont permis d'avancer quand la voie était difficile.

Mes remerciements s'adressent aussi à ma famille pour son soutien et à celle de ma conjointe pour m'avoir accepté parmi eux. Je remercié ma conjointe, Yaël Wojcik, pour sa patience lors de ces longues années d'études.

J'adresse ma reconnaissance à Génome Québec qui a contribué au financement de ce projet d'étude.

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet, qu'ils trouvent ici mes remerciements les plus sincères.

TABLE DES MATIÈRES

LISTE DES FIGURES

Note : dans le chapitre 5, présentant le logiciel « HTS Corrector », pour alléger la présentation nous avons numéroté seulement les figures montrant un résultat ou présentant un intérêt particulier.

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AICD	« Average Inter Cluster Distance ». Distance moyenne entre clusters.
B Score	« Better Score ». Méthode de correction proposée par Brideau *et al.* (2003).
Clustering	Groupement. Se dit des méthodes permettant de faire des groupements de données « proches », selon une mesure de distance prédéfinie.
FN	Faux Négatif. Élément rejeté à tort.
FP	Faux Positif. Élément accepté à tort.
« Hit »	Composé prometteur ayant une activité apparente par rapport à un seuil prédéfini.
K-means	Groupement utilisant la méthode *k-means* (voir Legendre et Legendre, 1998).
SASD	« Sum of the Average Squared inside-cluster Distances ». Somme des distances inter-cluster moyennes.
Well Correction	Correction par puits.
HTS	« High-Throughput Screening » ou criblage à haut débit.

RÉSUMÉ

Dans ce mémoire, nous abordons le problème de la correction d'erreurs systématiques et de la recherche des composés prometteurs (*i.e.* « hits ») dans les procédures de criblage à haut débit (HTS). Nous introduisons une nouvelle approche pour la correction des erreurs systématiques dans les procédures HTS et la comparons à quelques méthodes couramment utilisées. La nouvelle méthode, appelée « *well correction* » ou *correction par puits*, procède par une analyse des erreurs systématiques localisées au niveau des puits, à travers toute la procédure de criblage. Cette méthode permet une amélioration des résultats obtenus lors de la sélection des « hits », par des méthodes utilisant un seuil prédéfini. La *correction par puits* à montré des résultats supérieurs aux méthodes suggérées dans la littérature telles que : *correction par soustraction de l'arrière-plan* (« *background correction* » : Kevorkov et Makarenkov, 2005a, 2005b); « *median-polish* » et « *B score* » (Brideau *et al.*, 2003; Malo *et al.*, 2006).

Nous avons également comparé trois méthodes de recherche des « hits » utilisant des approches de groupement (*i.e.* « clustering ») : k-mean; somme des distances inter-cluster moyennes (SASD) et distance moyenne entre clusters (AICD). Ces méthodes proposent des algorithmes différents pour mesurer la distance entre les données provenant du criblage. Les méthodes de groupement utilisant *k-means* et SASD ont montré des résultats intéressants, mais aucune des méthodes étudiées n'a montré des performances pouvant justifier son utilisation dans tous les cas de figure.

Un logiciel, « HTS Corrector », a été développé dans le cadre de ce travail. Il intègre toutes les méthodes étudiées dans ce mémoire. D'autres fonctionnalités auxiliaires, pouvant aider le praticien dans l'analyse des résultats provenant d'une procédure HTS, ont aussi été intégrées.

Mots clés : criblage à haut débit, high-throughput screening, erreurs systématiques, correction de données, méthodes de groupement, recherche de hits, normalisation de données.

INTRODUCTION

La présence d'erreurs aléatoires et systématiques dans les données produites par un criblage à haut débit, représente un problème majeur dans la recherche de nouveaux médicaments. Lors de la sélection des composés prometteurs (« hits »), la présence de ces erreurs peut produire des faux positifs (*i.e.* des valeurs retenues alors qu'elles ne le méritaient pas) ou de faux négatifs (*i.e.* des valeurs écartées alors qu'elles auraient pu mener au développement d'un médicament). Les erreurs aléatoires ne peuvent être corrigées que par des procédures plus strictes, du matériel plus performant et par des mesures multiples (ex : double échantillonnage). Par contre, les erreurs systématiques peuvent être détectées et minimisées par des procédures statistiques appliquées aux données résultantes du criblage (Zhang, Chung et Oldenburg, 1999, 2000; Heuer, Haenel et Prause, 2003; Brideau *et al.*, 2003; Kevorkov et Makarenkov, 2005a, 2005b; Malo *et al.*, 2006; etc.). Ces auteurs proposent diverses solutions pour ce problème.

Dans nos travaux nous nous intéressons au processus de criblage dès la production de résultats numériques. Nous écartons toute méthode s'appliquant au matériel ou aux procédures de laboratoire. Notre approche propose des outils informatiques pour aider le praticien dans la sélection de composés prometteurs, que se soit lors de l'analyse des données, de leur correction ou de la recherche des « hits ».

Nous nous concentrons sur l'analyse de quelques méthodes existantes et recommandées. Nous les comparons entre elles, en mesurant leurs performances sur

des données simulées et des données réelles. L'utilisation de données simulées nous permet de contrôler tous les paramètres de l'expérience et d'avoir ainsi une référence unique pour toutes les méthodes testées. Nous pouvons ainsi les classifier selon leur capacité à détecter et à corriger les erreurs systématiques et selon le nombre de faux positifs et de faux négatifs qu'elles produisent. Leur application à des données réelles (jeux de données du concours de l'Université McMaster) nous permet de les tester, dans des conditions semblables, sur des données types pouvant provenir d'une procédure de criblage. Mon travail a consisté principalement au développement de la nouvelle méthode proposée par notre équipe, la *correction par puits* (ou « well correction »), ainsi qu'a la création des programmes de simulation nécessaires à son développement et à la comparaison avec les autres méthodes proposées par la littérature. Finalement, j'ai réalisé la nouvelle version du logiciel HTS Corrector (voir ci-après).

Une des priorités guidant nos travaux est le développement d'un logiciel intégrant les méthodes testées. Lors de nos recherches, nous avons remarqué que les méthodes proposées dans la littérature étaient implémentées individuellement par chaque laboratoire, ou que les laboratoires développaient en interne leurs propres méthodes. Cette façon de faire nous semble coûteuse en termes de ressources, et est sujette à des erreurs. Les logiciels développés par des équipes de recherche académiques ou privées sont souvent peu évolutifs et fortement dépendants du savoir-faire de la personne chargée de sa réalisation (parfois un praticien n'ayant pas forcement toutes les connaissances nécessaires en développement logiciel). Ces logiciels sont souvent abandonnés dès que leur concepteur quitte l'équipe et sont rarement mis à jour (voir aussi Geldenhuys *et al.*, 2006). Nous ne voulions pas que le logiciel réalisé par notre équipe, « HTS Corrector », subisse le même sort et nous avons essayé d'y apporter le soin nécessaire pour qu'il puise continuer à évoluer.

Ce mémoire comporte, au premier chapitre, une introduction au processus de criblage à haut débit permettant de se familiariser avec le domaine et sa nomenclature. Nous présentons un bref historique du processus, suivi des étapes du développement d'un nouveau médicament. Suivent quelques remarques sur les avantages et inconvénients du criblage pour finir avec une présentation plus technique du matériel utilisé.

Le deuxième chapitre traite des méthodes de normalisation et de sélection des « hits ». Il introduit les méthodes utilisées ainsi que les notions mathématiques et statistiques nécessaires à leur compréhension.

Le troisième chapitre traite spécifiquement de la problématique des erreurs systématiques pouvant affecter les données obtenues lors du criblage. Il introduit les causes probables de ces erreurs et les méthodes pouvant être utilisées pour leur correction. Quelques résultats sont aussi présentés.

Le quatrième chapitre est composé de divers articles que nous avons cosignés. Notamment : une comparaison entre les méthodes de correction des erreurs systématiques au moyen de la *correction par soustraction de l'arrière-plan* et de la *correction par puits*; un article de présentation du logiciel « HTS Corrector »; une comparaison de trois techniques de recherche des « hits » par des méthodes de groupement et, finalement, un article présentant diverses méthodes de correction de données et les évaluant entre elles.

Le cinquième chapitre est consacré à la présentation du logiciel « HTS Corrector » et de ces principales fonctionnalités.

Nous concluons par une synthèse du travail effectué et par des pistes de recherche pour l'amélioration de nos méthodes.

CHAPITRE I

LE CRIBLAGE À HAUT DÉBIT (HTS)

1.1 Historique du HTS

Le « High-Throughput Screening » (HTS), ou criblage à haut débit, est le fruit d'une évolution des méthodes utilisées par les compagnies pharmaceutiques pour le développement des médicaments. L'échantillonnage existait avant l'apparition du HTS mais se faisait manuellement. Le processus était très long (environ 200 composés analysés par semaine) et sensible aux erreurs de manipulation (Nelson et Yingling, 2004). Faute de moyens, et de connaissances suffisantes, la détection de molécules bioactives se limitait, avant les années 70, à de simples essai-erreurs, à très petite échelle. Les mécanismes biologiques impliqués n'étaient pas suffisamment compris pour pouvoir cibler plus précisément les recherches.

Durant les années 70 et 80, le développement de la génétique, de l'électronique et de l'ingénierie, augmentèrent sensiblement la productivité des laboratoires de recherche. Les progrès techniques intégrés aux instruments de laboratoire et une meilleure connaissance des mécanismes de biologie cellulaire, ont permis, pour la première fois, une approche de développement ciblée (Macarron, 2006; Chen, 2006; Wu et Doberstein, 2006).

Ces développements se sont poursuivis dans les années 90, en tirant parti des avancées effectuées par l'informatique et, plus récemment, par la robotique. Grâce à l'automatisation poussée des diverses étapes nécessaires au criblage (stockage, préparation, maturation, manipulation et mesure), les capacités de traitement sont passées à plus de 100'000 composés par jour. Une recherche typique, utilisant les possibilités du HTS, peut produire jusqu'à 50 millions de mesures par an (Heuer, Haenel et Prause, 2003). Actuellement, la génération des résultats dépasse souvent les capacités d'analyse de l'équipe.

Pour le futur, les recherches sont orientées vers une meilleure validation des résultats obtenus par le criblage, l'anticipation des effets des composés étudiés, avant même leur test (conformités ADMET, Virtual HTS), et le développement de composés sur mesure (chimiothèques développées *in silico*), selon les qualités recherchées et les sites ciblés (Fox *et al.*, 2006; Chen, 2006). La miniaturisation des appareils de manipulation et de mesure, pourrait permettre la création d'équipes de recherches plus souples et locales, pouvant procéder, en interne, à leurs propres criblages, et ce, pour des coûts et des délais réduits. La miniaturisation des plateaux, servant aux manipulations et aux mesures des composés, permettrait aussi une réduction des coûts (quantités utilisées plus faibles) et des délais (augmentation des composés testés dans chaque plateau).

L'industrie pharmaceutique cherche à réduire ses coûts de recherche et de développement, pour maximiser les revenus produits par les nouveaux médicaments, avant l'expiration du délai de protection des brevets. Elle cherche aussi à réduire les délais de recherche et d'acceptation des nouveaux médicaments, pour pouvoir faire face à l'apparition de nouvelles maladies, dans des délais raisonnables.

1.2 Survol du processus de recherche d'un médicament

La recherche d'un nouveau médicament se divise, généralement, en plusieurs phases (voir figure 1.1). Elle part d'une première étape de sélection de composés prometteurs (choix des composés à analyser et criblage primaire pour la sélection des « hits »), suivie d'étapes de sélection de plus en plus rigoureuses. Le but étant d'éliminer les composés inefficaces et de restreindre le nombre des composés devant faire l'objet d'analyses plus poussées. La première étape de ce cycle est communément faite par criblage à haut débit.

Le criblage à haut débit est une méthode de dépistage d'une activité biologique donnée dans une banque de composés chimiques (chimiothèques). Ce qui la distingue des méthodes plus traditionnelles est le nombre de composés testés à la fois et la rapidité du processus. La gestion de la sélection des composés selon les résultats doit de plus en plus être assistée par des logiciels permettant de compenser la faillibilité humaine et surtout capables de traiter les énormes quantités de données produites.

Le criblage primaire (voir figure 1.1 : « Primary screen ») vise à faire une première sélection de candidats parmi une ou plusieurs chimiothèques complètes en utilisant un test biologique sommaire mais rapide. Typiquement, on n'utilisera qu'une ou peu de concentrations pour chaque composé à tester. Les candidats retenus (ci-après « hits ») sont déterminés, le plus souvent, par comparaison avec un seuil choisi statistiquement à l'aide de valeurs de contrôle du test biologique. Ils sont ensuite retracés puis déplacés selon une grille plus appropriée aux étapes subséquentes (« cherry picking »).

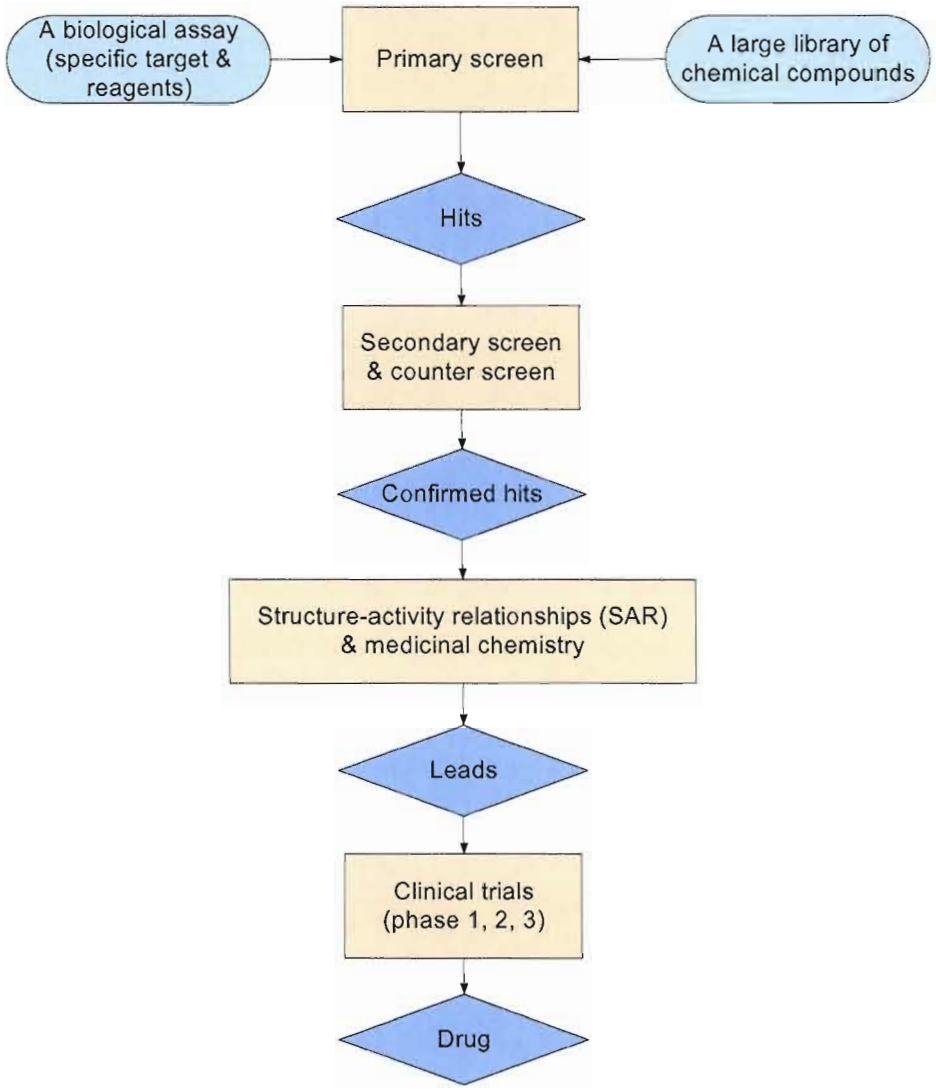


Figure 1.1 : Diagramme du processus de développement de médicaments (tiré de Malo *et al.*, 2006).

Le criblage secondaire (la plupart du temps non HTS) consiste à tester l'effet des candidats retenus, à des concentrations décroissantes, sur le système biologique étudié. Seules les molécules montrant une activité sous une concentration prédéterminée sont retenues pour les étapes subséquentes. Les candidats sont également testés sur un système similaire pour déterminer si l'effet du candidat est spécifique.

Une fois les candidats confirmés, des expériences de chimie fonctionnelle sont tentées. Des composés similaires aux « hits » confirmés sont synthétisés ou achetés, et sont testés pour les deux systèmes biologiques (spécifique et non-spécifique). Les différences dans les résultats obtenus pour chacun des composés, permettent de déterminer quelle zone de la molécule candidate est importante pour maximiser l'effet biologique recherché. Elles permettent également de faire le design d'une nouvelle molécule similaire à la molécule parent, mais ayant une plus grande action sur le système ciblé ou une plus grande spécificité pour ce dernier (avec moins d'effets secondaires potentiels).

Une fois que l'activité thérapeutique et la spécificité des candidats est satisfaisante, des études précliniques (*in vivo*) et cliniques sont lancées sur les candidats les plus prometteurs. Les études précliniques testent généralement l'efficacité des produits dans un contexte initialement cellulaire (lignées cellulaires appropriées à la thérapie), puis systémique (organismes vivants complets, tels que les rongeurs et les simiens). La toxicité du produit est également vérifiée lors de ces essais.

Les études cliniques sortent du cadre de ce mémoire, mais leurs grandes lignes peuvent être résumées. Elles visent à vérifier l'efficacité des meilleurs candidats à guérir ou à soulager l'être humain (volontaire) atteint d'une maladie. Dans un second temps, à s'assurer qu'il n'y a pas d'effets secondaires à son administration et, dans le cas contraire, les identifier et mesurer leur gravité. À partir des données précédentes, des tests sont effectués pour déterminer le dosage optimal pouvant maximiser les bienfaits et minimiser les effets secondaires indésirables. Le médicament est également testé sur des cohortes d'individus sains. Selon les résultats des essais cliniques, une décision stratégique d'entreprise est prise pour choisir lequel des différents composés sera mis en production et commercialisé. Les coûts de production et de mise en marché sont également pris en compte (et pèsent lourd) lors de cette décision (Drews, 2000, 2001; Dove, 2003).

1.3 Avantages et limites du processus

Jusqu'au milieu du 19ème siècle, les méthodes de recherche des médicaments se basaient sur l'observation et l'expérimentation de produits naturels ou nouvellement synthétisés (essor de l'industrie chimique à la fin du 19ème), ainsi que sur l'étude de remèdes traditionnels. Les chercheurs ne faisant que trouver les concentrations appropriées ou sélectionner les composés actifs pour mieux les synthétiser. L'expertise (et la chance) du chercheur avaient une grande influence sur les résultats des recherches. La plupart des médicaments « faciles » ayant déjà été trouvés par ces méthodes, l'industrie pharmaceutique se trouvait face à une impasse.

Au cours du 20ème siècle, plus particulièrement vers sa fin (années 80 et suivantes), les progrès techniques et génomiques ont ouvert la voie à des recherches utilisant des méthodes plus « scientifiques » et systématiques. Les nouvelles approches de chimie combinatoire ont grandement contribué au développement de nouvelles méthodes dans la recherche des médicaments. Actuellement, on peut produire des composés « à volonté » selon les souhaits du client, en faisant varier leur formule chimique molécule par molécule, voir atome par atome. Suite à ces progrès techniques, l'industrie créa des chimiothèques composées de millions de produits chimiques sans se soucier à priori de leur action biologique. Il restait à analyser tous ces composés.

Le processus de sélection, par criblage à haut débit, permet de faire un premier tri parmi ces nombreuses molécules. En les mettant en présence de cibles thérapeutiques, l'interaction entre les composés chimiques et la cible permettra de mesurer leurs effets, de manière rapide et efficace. Par le traitement de milliers de composés, on pourra mettre en évidence un pourcentage (en général faible et aux environs de 1%) de composés pouvant mener à la découverte et à la réalisation d'un nouveau médicament.

Cette nouvelle approche permet de se soustraire aux limites des anciennes procédures qui partaient d'un effet constaté pour ensuite rechercher les éléments chimiques le produisant. Le cycle est maintenant inversé, on procède aux tests, par force brute, de plusieurs composés et on espère constater un effet exploitable dans le traitement d'une maladie (Heyse, 2002)

Le processus de découverte et de commercialisation d'un médicament est long et couteux (voir figure 1.2). Pour dix millions de composés étudiés (dans la figure 1.2, le 10'000 correspond à l'étape de criblage secondaire), avec un peu de chance, on peut commercialiser un médicament. Le processus complet peut prendre, cependant, entre 10 à 15 ans (parfois plus) et coûter plusieurs centaines de millions de dollars sans garantie de succès commercial (voir figure 1.2).

Figure 1.2 : Phases, coûts et délais du cycle de développement d'un médicament. Montant des coûts en livres. (tiré du site www.schoolscience.co.uk).

L'utilisation du HTS dans les phases préliminaires de la recherche d'un médicament, permettent de raccourcir un peu les délais et de trouver des médicaments utilisant des composés chimiques que l'on ne serait pas tentés de tester au premier abord. En augmentant le nombre de composés étudiés, on s'attend à avoir un nombre grandissant de molécules prometteuses menant à la réalisation de nouveaux médicaments. Le HTS a connu une première étape de frénésie (due à l'apport massif de fonds et à

l'échec des méthodes traditionnelles pour contrer de nouvelles maladies) où la quantité et la rapidité primaient sur la qualité (fin des années 80 jusqu'à tard dans les années 90. Voir Macarron, 2006). Actuellement, nous sommes entrés dans une phase où l'accent est mis sur la qualité des processus et des mesures. Le nombre de médicaments développés grâce à la technique du HTS est en progression mais, à cause du cycle de découverte et de développement pouvant prendre jusqu'à 15 ans, les possibles médicaments trouvés grâce à HTS sont encore en développement (les essais cliniques pour des médicaments provenant du HTS commencent au début du millénaire).

Une étude récente, faite sur 58 laboratoires HTS par Fox *et al.* (2006), montre l'importance grandissante du nombre de « leads » trouvés par cette méthode, qui est passé de 328 en 2000 à 746 en 2004. Elle indique aussi que 104 produits provenant de « leads » trouvés par criblage à haut débit sont en phase de candidature pour des essais cliniques et que quatre médicaments sont actuellement commercialisés (ils ont été criblés en 1984 (1), 1989 (2), et en 1999 (1)). Cette étude met en avant le fait que les récents progrès dans la mise en œuvre de la méthodologie de criblage (entre autres des criblages plus ciblés) devraient augmenter le pourcentage de découvertes et de succès ainsi que réduire le délai entre le criblage et la mise en marché des médicaments.

Après un optimisme sans limites sur l'efficacité du processus HTS (on s'attendait à la sortie d'un médicament dans les mois suivant la publication d'un article), on s'est heurté aux limites de l'approche. Les critiques ironisent, en assimilant la recherche d'un médicament à la recherche d'une aiguille dans une meule de foin et en prétendant que les méthodes utilisées par le criblage à haut débit correspondraient à ajouter du foin avant de chercher l'aiguille (Macarron, 2006).

La création de chimiothèques spécialisées permet de réduire les essais *in vitro* inutiles en délaissant les produits visiblement toxiques et / ou inefficaces et en répondant à des mesures de biocompatibilité plus poussées. Pour cela on doit utiliser

toutes les connaissances sur les propriétés biochimiques souhaitées du médicament et sur les effets néfastes connus de certaines molécules, pour exclure un maximum de molécules « parasites ». Les molécules biocompatibles doivent, généralement, se conformer à certaines propriétés telles que les propriétés ADMET (« absorption, distribution, metabolism, elimination / excretion and toxicity »), le règles de Lipinski (Lipinski *et al.*, 2000, 2001), ou répondre à des métriques « drug-like » (Sirois *et al.*, 2004).

Les processus de recherche de nouveaux médicaments par criblage à haut débit se justifient par l'apparition de nouvelles maladies (nouveaux virus ou virus résistants aux anciens médicaments, maladies dues au vieillissement de la population), par l'absence de médicaments pour des maladies telles le cancer, l'Alzheimer ou l'arthrite, et par la prolifération de maladies comme l'asthme. Ils se justifient aussi par les retours sur investissement plus rapides pour les nouveaux médicaments, dus à un cycle de recherche et développement réduit.

Le coût estimé des maladies non guéries était de 646 milliards de dollars en 2000 (source : Pharmaceutical Research and Manufacturers of America, 2003, Industry profile, PhRMA, Washington, D.C.)

Le nouveau défi posé par la sélection des composés prometteurs, au moyen du criblage à haut débit, est la gestion des inévitables erreurs aléatoires et / ou systématiques inhérentes à tout processus industriel. C'est ce problème qui est à la base de nos recherches actuelles. Il sera présenté dans le 3ème chapitre de ce mémoire.

1.4 Présentation du matériel

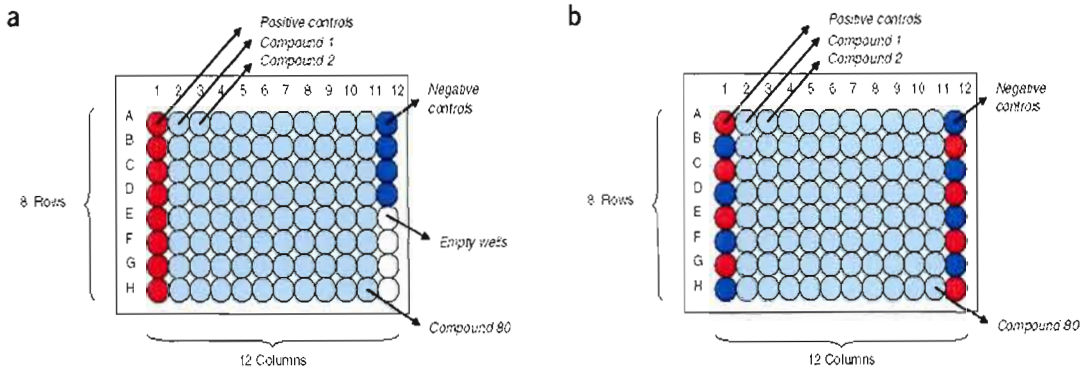


Figure 1.3 : Compositions classiques des plateaux utilisés dans une campagne HTS (tiré de Malo *et al.*, 2006).

Pour effectuer une analyse HTS, on place des composés cibles dans des éprouvettes ou puits (« wells »), intégrés dans des plateaux (voir figure 1.3). Ils permettent leur traitement automatisé par des robots manipulateurs. Les standards de l'industrie sont les plateaux à 96 puits (8 x 12) de 40 à 150 µL, à 384 puits (4 x 96 puits) de 20 à 70 µL et, plus récemment, à 1536 puits (4 x 384 puits) jusqu'à 10 µL. On trouve aussi les deux derniers en version faible volume (LV) de 5 à 30 µL pour les plateaux à 384 puits et jusqu'à 5 µL pour ceux à 1536 puits. Des versions propriétaires existent, mais elles ne sont pas recommandées. La miniaturisation des puits se justifie par le traitement de plus de composés pour chaque plateau (gain de temps) et par les volumes inferieurs qui permettent de réduire les coûts en composés chimiques (typiquement moins de 50 $ pour 1 à 5 mg de composé, mais les quantités utilisées font en sorte que le prix global est important (Fassina, 2006)). Par contre, les très faibles volumes véhiculés dans les pipettes de remplissage se heurtent à des problèmes de dynamique des fluides propres aux faibles volumes, ce qui nécessite du matériel plus performant et peut induire plus d'erreurs lors du traitement. On parle de « Ultra HTS » dès que le traitement avoisine les 100'000 composés par jour.

Les composés cibles doivent faire l'objet d'une maturation en milieux contrôlé pendant une durée définie, dépendant du composé et de l'effet recherché. On ajoute ensuite le ou les composés chimiques à tester et on procède à une deuxième maturation. Selon le procédé de mesure, on peut procéder à un filtrage et possiblement à un nettoyage des mélanges avant lecture. Pour pouvoir contrôler la qualité des mesures effectuées, on ajoute, typiquement, des puits de contrôle ayant des propriétés connues (voir figure 1.3). Les mesures effectuées sur ces puits permettront de savoir si des erreurs de mesure se sont produites et d'estimer leur ordre de grandeur. On pourra ainsi écarter les résultats d'un criblage si les valeurs rencontrées ne correspondent pas aux attentes (pour plus de détails voir les chapitres 2 et 3).

Les méthodes de mesures généralement utilisées pour le criblage à haut débit sont : la fluorescence; l'absorption; la luminescence; la polarisation fluorescente (FP); fluorescence temporelle (TRF : « time-resolved fluorescence »); fluorescence par résonance énergétique de transfert (FRET : « fluorescence resonance energy transfer »). Pour une liste plus détaillée voir l'article de Wu et Doberstein (2006).

Toutes sortes d'erreurs, se produisant lors de la préparation ou lors de la manipulation des plateaux, peuvent fausser les résultats. Les sources d'erreurs les plus fréquentes, lors de la préparation, sont : les délais d'incubation non respectés; les conditions de stockage non-conformes; les erreurs de volumes lors du pipetage ou dues à l'évaporation des composants; les contaminations éventuelles. Les erreurs peuvent provenir aussi des appareils de mesure (erreurs intrinsèques à l'appareil ou la méthode de mesure, erreurs de positionnement, conditions ambiantes défavorables, etc.).

Pour maximiser les chances d'obtenir des résultats concluants, la plupart des procédures doivent être automatisées. Elles sont effectuées par des robots dans des lieux confinés et contrôlés. L'industrie de l'équipement propose des solutions pour tous les besoins, que se soit la manipulation, le stockage ou la lecture. Les solutions

proposées sont souvent modulaires et permettent « l'agrandissement » du système selon la croissance des besoins sans devoir forcement se départir de l'existant. Actuellement, il existe des mini laboratoires autonomes à des coûts relativement modérés (voir figure 1.4 et 1.5). Ces laboratoires ne représentent plus qu'une fraction des coûts d'une procédure HTS.

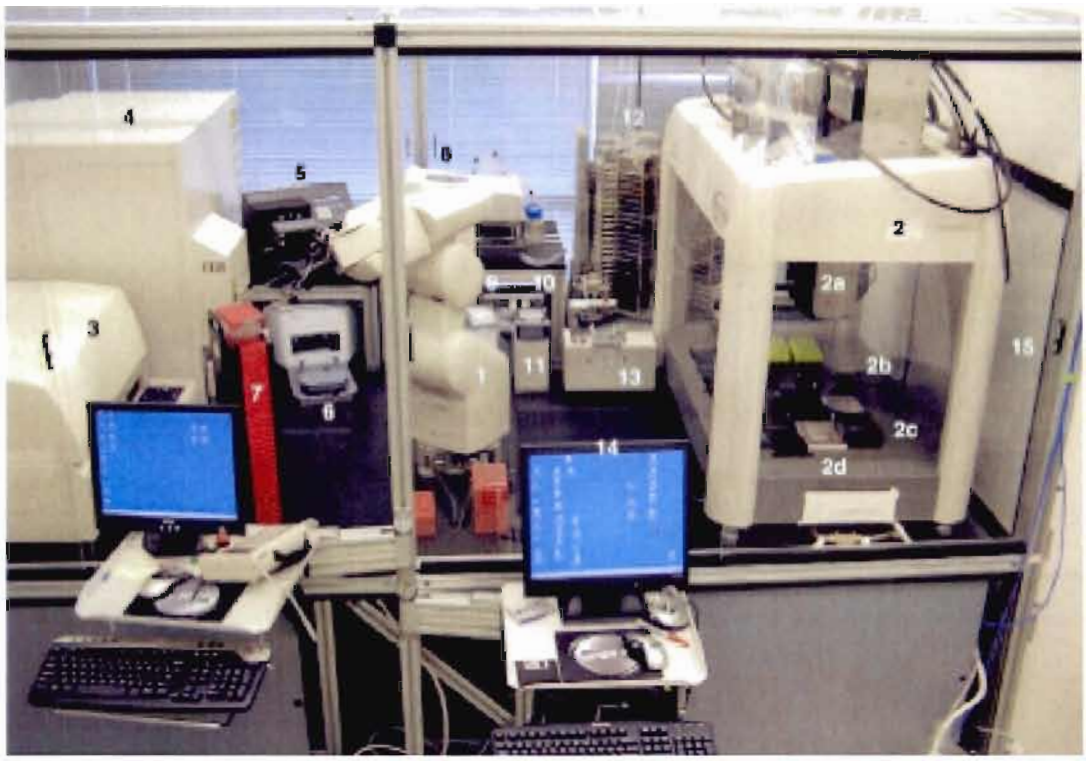


Figure 1.4 : Système de criblage robotisé entièrement intégré. (1) Robot manipulateur; (2) Module comprenant : (2a) tête de distribution par pipettes pour 96 ou 384 puits, (2b) mélangeur, (2c) système de filtrage par pression négative et (2d) une station de lavage par ultrasons; (3) Appareil de lecture acceptant 11 modes de lecture; (4) Système de rangement en milieu contrôlé; (5) Nettoyeur de plateaux; (6) Centrifugeuse; (7) Pile de rangement; (8) Distributeur de réactifs; (9) Imprimante de codes barres; (10) Lecteur de codes barres; (11) Module de réorientation des plateaux; (12) Incubateur à température ambiante; (13) Module de gestion des couvercles des plateaux (tiré de Wu et Doberstein, 2006).

Figure 1.5 : Systèmes de criblage robotisé entièrement intégré (tiré des catalogues des fabricants).

Figure 1.6 : Systèmes de gestion des solutions (tiré de Fassina, 2006).

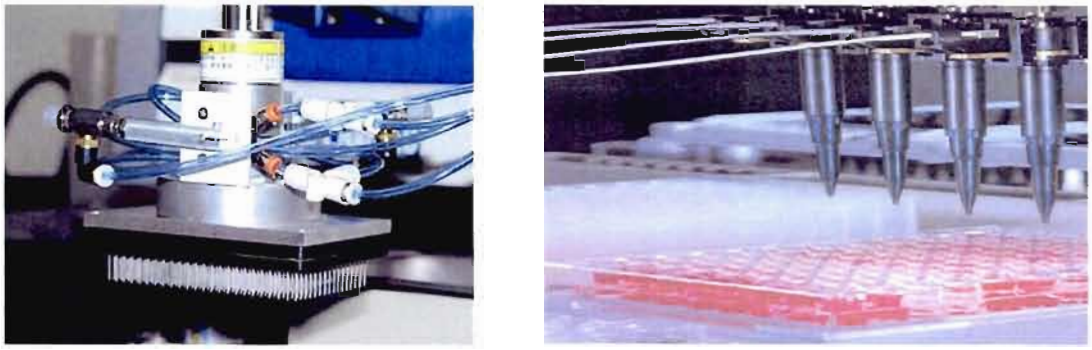


Figure 1.7 : Têtes de pipettes (tiré de Fassina, 2006).

CHAPITRE II

NORMALISATION ET SÉLECTION DES « HITS »

2.1 Normalisation

Une fois le processus de mesure du criblage effectué, il faut procéder à l'analyse des données brutes produites, en vue d'identifier les composés prometteurs pouvant mener au développement d'un nouveau médicament. Pour analyser les données produites par les appareils de mesure, il faut avant tout les transformer pour pouvoir les comparer. Les données brutes peuvent présenter de légers biais ou des ordres de grandeurs différents selon le plateau, les conditions ambiantes et le moment auquel on été faites les mesures. Ceci peut venir de temps d'incubation différents, de la préparation des plateaux par familles de composés, de volumes de solution changeants, etc. (pour plus de détails voir la littérature, notamment : Zhang, Chung et Oldenburg (1999, 2000); Heuer, Haenel et Prause (2003); Brideau *et al.* (2003); Kevorkov et Makarenkov (2005a, 2005b); Malo *et al.* (2006)). La moyenne et l'écart type des valeurs mesurées peuvent varier d'un plateau à l'autre. La pratique veut que l'on normalise les données brutes pour les rendre comparables et pouvoir ainsi procéder à leur analyse statistique.

Il existe plusieurs techniques de normalisation pouvant être utilisées dans le cadre d'une campagne HTS. Si la distribution des mesures est gaussienne, on peut procéder à une transformation logarithmique avant de procéder à la normalisation (Kevorkov et Makarenkov, 2005a). Nous présentons ci-après les méthodes de normalisation conseillées par la plupart des auteurs, notamment Brideau *et al.* (2003), Kevorkov et Makarenkov (2005a, 2005b) et Malo *et al.* (2006).

Normalisation Centrée Réduite (appelée aussi « Z Score ») : C'est la normalisation classique utilisée dans tous les manuels de statistiques. Les données ainsi normalisées auront une moyenne de zéro et un écart-type de un. Les valeurs sont normalisées selon la formule suivante :

$$x'_i = \frac{x_i - \mu}{\sigma}, \tag{1}$$

où la moyenne μ de chaque plateau est calculée comme suit :

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i , \tag{2}$$

et l'écart-type σ du plateau comme suit :

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1}}, \tag{3}$$

où x'_i est la valeur normalisée correspondant à la valeur mesurée x_i. Ces calculs sont effectués pour les n valeurs de chaque plateau.

Makarenkov *et al.* (2007) proposent de prendre la moyenne et l'écart-type de l'ensemble de la procédure HTS si les conditions sont semblables. Cette façon de faire permet de calculer les paramètres en utilisant plus de valeurs que si on se limite à un seul plateau. Typiquement, dans nos travaux, 80 valeurs par plateau contre : 80 valeurs/plateaux x 1250 plateaux = 100 000 valeurs. Le nombre accru des valeurs considérées permet d'obtenir des résultats moins sensibles aux valeurs extrêmes et plus proches des moyennes et écart types réels. Cependant, dans la pratique, on ne peut pas garantir que des déviations ne se soient produites lors des mesures de chaque plateau. Ces types de déviations peuvent se produire par à-coups (par exemple si les conditions de mesure ont changé, si les mesures sont effectuées sur une longue période ou pour des familles différentes de composés) ou graduellement (ce qui produit à la longue un écart important entre les premières et les dernières mesures). Si on ne peut garantir l'homogénéité des mesures tout au long de la procédure, la méthode classique, qui effectue ces calculs par plateaux, est plus sûre.

La normalisation par intervalle : On peut aussi faire une normalisation par rapport à deux bornes (ci-après L_{max} et L_{min}) pour s'assurer que les valeurs seront comprises dans l'intervalle $[L_{min}, L_{max}]$. On utilise donc la transformation suivante :

$$x'_i = \frac{(x_i - x_{\min})(L_{\max} - L_{\min})}{x_{\max} - x_{\min}} + L_{\min}, \qquad (4)$$

où x_{max} et x_{min} sont respectivement les valeurs maximum et minimum des mesures et x'_i la valeur normalisée correspondant à la valeur mesurée x_i.

Kevorkov et Makarenkov (2005a) ont proposé de fixer [L_{min}, L_{max}] à [-1, 1], mais dans la pratique on utilise plutôt les valeurs maximum (ou la moyenne) des puits de contrôle positifs et minimum (ou la moyenne) des puits de contrôle négatifs selon la formule suivante :

$$x_i' = \frac{H - x_i}{H - L} * 100\%, \tag{5}$$

où H est la moyenne des contrôles positifs et L la moyenne des contrôles négatifs et x'_i la valeur normalisée correspondant à la valeur mesurée x_i.

Les méthodes ci-dessus sont sensibles aux valeurs extrêmes à cause de l'utilisation des moyennes. Elles sont plus fortement affectées si le nombre de mesures traitées est faible. Différents auteurs proposent de faire une première normalisation sur l'ensemble des données de chaque plateau, puis de soustraire à ces données les valeurs qui dépassent un certain seuil (typiquement trois écarts types) et, finalement, de refaire une deuxième normalisation sur les données restantes (voir notamment Kevorkov et Makarenkov (2005a, 2005b) pour des détails, ou Malo *et al.* (2006) pour la méthode « jackknife » Z score). Cette façon de faire minimise les variations dues à des valeurs extrêmes tout en restant simple du point de vue de la compréhension (mathématiques et statistiques de base) et facilement implémentable dans n'importe quel langage de programmation sans besoin de connaissances approfondies. Les méthodes de normalisation présentées ci-dessus sont facilement réalisables dans un chiffrier électronique (ou tableur), par exemple une feuille Excel.

Les critiques de ces méthodes, notamment, Brideau *et al.* (2003), Gribbon *et al.* (2005), et Malo *et al.* (2006), leur préfèrent des méthodes plus robustes qui utilisent des médianes au lieu des moyennes, par exemple le B score qui utilise la méthode itérative de « median polish » à deux critères de Tukey (1977).

La méthode « median polish » permet d'enlever les erreurs systématiques de lignes et de colonnes dans une matrice de valeurs. Elle s'apparente à un calcul d'analyse de variance (ANOVA) sans en avoir les inconvénients. L'ANOVA part du principe d'une distribution centrée réduite des erreurs et est peu résistante aux valeurs extrêmes (Brideau *et al.*, 2003). « Median polish » présente néanmoins les inconvénients des procédures itératives : résultats pouvant varier d'un calcul à l'autre pour des données initiales identiques; seuils d'arrêt des itérations arbitraires; mise en application plus complexe; etc. On calcule les résidus, parfois nommés erreurs, des valeurs mesurées par plateaux selon la formule suivante :

$$r_{ijp} = y_{ijp} - \left(\hat{\mu}_p + \hat{R}_{ip} + \hat{C}_{jp}\right), \tag{6}$$

où y_{ijp} est la valeur mesurée dans la ligne i, colonne j du plateau p ; $\hat{\mu}_p$ est la moyenne estimée du plateau p, $\hat{R}_{ip}$ l'erreur due à la ligne i du plateau p et $\hat{C}_{jp}$ l'erreur due à la colonne j du plateau p; r_{ijp} est le « résidu » de la mesure y_{ijp}, soit l'estimation de la valeur une fois les erreurs de ligne et de colonne enlevées.

Pour chaque plateau on ajuste les valeurs en les divisant par la déviation médiane absolue (« median absolute deviation » MAD) selon les formules suivantes :

$$r'_{ijp} = \frac{r_{ijp}}{MAD_p}, \tag{7}$$

où $MAD_p = médiane\{ \mid r_{ijp} - médiane(r_{ijp}) \mid \}$ (8)

Si les données suivent une loi normales, on multiplie le MAD par un coefficient correcteur de 1.4826 ce qui le réajuste de manière à ce qu'il corresponde à une approximation de l'écart-type.

Nos récentes études (Makarenkov *et al.*, 2007 reproduit au chapitre 4 section 4) ont montré que cette méthode n'est pas aussi performante que semblent le croire les auteurs l'ayant proposée (Brideau *et al.*, 2003) et leur défenseurs. D'après nos études (et simulations), B Score détecte parfaitement des erreurs sur une ou deux lignes (ou colonnes) par plateau, mais dans des conditions plus sévères (par exemple : erreurs possibles sur chaque ligne et chaque colonne des plateaux), elle se montre moins performante. Elle ne détecte pas les erreurs que nous avons introduites selon ce modèle et la « correction » effectuée tend même à empirer la qualité des données. Ce problème ayant été mis en évidence lors de nos derniers travaux, nous ne sommes pas encore en mesure d'en expliquer la cause. Une cause probable pourrait être l'atteinte par « median polish », et donc par B score dont elle est une étape préliminaire, de minimums locaux qui arrêteraient la procédure itérative avant d'atteindre le minimum global recherché. Une autre cause pourrait être la faible quantité de données par plateaux, nos essais s'étant portés sur des plateaux de 96 éléments, soit 80 mesures une fois enlevées les valeurs de contrôle. Il est possible que cette quantité ne permette pas de trouver une valeur médiane vraiment représentative des valeurs de chaque plateau ou qu'elle soit fortement dépendante de la dispersion des valeurs des éléments mesurés. Ces dernières remarques ne sont que des conjonctures qui doivent être analysées plus en détail par des travaux subséquents.

2.2 Facteur Z'

Il existe aussi un facteur proposé par Zhang, Chung et Oldenburg (1999, 2000) : le facteur Z'. Il est très utilisé pour définir la qualité globale d'une analyse.

Il correspond au rapport définissant la bande de séparation entre les contrôles positifs et négatifs :

$$Z' = 1 - \frac{3\sigma_{c+} + 3\sigma_{c-}}{\left|\mu_{c+} - \mu_{c-}\right|}, \tag{9}$$

où σ_{c+} et σ_{c-} sont les écarts types des contrôles positifs et négatifs, respectivement, et μ_{c+} et μ_{c-} leurs moyennes.

On catégorise la qualité d'un criblage HTS selon la valeur du facteur Z' obtenue :

- pour 1 > Z' > 0.9 : le criblage est d'excellente qualité ;
- pour 0.9 > Z' > 0.7 : le criblage est bon ;
- pour 0.7 > Z' > 0.5 : toute amélioration dans les processus de mesure pourra rendre la sélection des « hits » plus exacte ;
- finalement 0.5 = Z' : le minimum acceptable pour une campagne HTS.

On peut procéder au calcul du facteur Z', au début du criblage, pour un petit nombre de plateaux (environ 10 plateaux) pour estimer la qualité de la procédure avant de la lancer au complet. En cours de criblage il est intéressant de le revérifier périodiquement pour permettre des corrections sur les appareils ou pour arrêter la procédure si ses résultats semblent compromis.

2.3 Sélection des « hits »

Une fois les données normalisées, et donc comparables entre elles, on peut procéder à la recherche des composés prometteurs (« hits »). Les composés, sélectionnés lors du criblage primaire, seront étudiés dans les phases subséquentes en vue d'obtenir des renseignements pouvant conduire à la découverte ou à la création d'un médicament. Généralement, les « hits » seront choisis parmi les composés qui présentent des valeurs inferieures (pour un criblage d'inhibition) à un certain seuil, établi en fonction des paramètres et particularités de la campagne HTS. Typiquement, on s'attend à trouver entre 0.01 et 1% de « hits » parmi les valeurs résultantes d'un criblage (Woodward *et al.*, 2006).

La stratégie couramment utilisée pour la sélection des « hits » propose de sélectionner comme valeur active (« hit »), toute valeur inférieure (pour un test d'inhibition) à un seuil prédéfini, typiquement la moyenne moins trois écarts types. Ce seuil garantit qu'environ 99.7% de l'ensemble des valeurs seront comprises entre $\mu - 3\sigma$ et $\mu + 3\sigma$ (pour une distribution des valeurs suivant une loi normale) ce qui, dans un cas d'une campagne HTS d'inhibition, nous laisse environ 0.3% / 2, soit 0.15%, de valeurs à analyser lors de la phase suivante, soit le criblage secondaire (voir figure 1.1 au premier chapitre).

Woodward *et al.* (2006) proposent de choisir parmi les éléments ayant un effet de 40% par rapport aux valeurs neutres (considérées ayant un effet de 0%). D'autres procédures proposent de sélectionner un nombre, ou pourcentage, prédéfini de valeurs. Par exemple, on choisira 1% de valeurs, parmi les plus basses obtenues lors du criblage HTS. Des approches dictées par les capacités d'analyse des laboratoires, proposent de sélectionner uniquement le nombre de composés pouvant être raisonnablement analysés lors des opérations subséquentes.

Ces deux dernières approches sont guidées par des contraintes financières et par les ressources disponibles. Elles ne trouvent aucune justification statistique; le nombre de « hits » n'étant pas connu à l'avance et pouvant changer selon les composés, ou les familles de composés, testés. Elles peuvent mener à un gaspillage de ressources si le seuil est trop permissif (nombre élevé de faux positifs), ou au contraire, à perdre des composés très prometteurs (faux négatifs). Ces derniers étant écartés simplement parce que leurs valeurs n'entrent pas dans une zone choisie. Cependant, ces composés pourraient être efficaces dans d'autres conditions ou concentrations.

D'autres stratégies sont aussi mises en avant comme le groupement de valeurs (« clustering ») (voir Gagarin, Makarenkov et Zentilli, 2006 et Schnecke et Boström, 2006). Ces stratégies se basent sur le fait que les valeurs mesurées pour des composés actifs sont significativement éloignées de celles mesurées pour des composés inactifs. On pourra ainsi chercher deux groupes, les composés actifs et les composés inactifs (le groupe le plus volumineux).

Typiquement, il existe deux approches pour procéder à un groupement, l'approche ascendante (« bottom-up » ou « agglomerative »), qui part d'un élément et cherche des voisins, ou l'approche descendante (« top-down » ou « divisive »), qui part d'un groupe unique et tente de le diviser en sous-groupes aux propriétés semblables. La première solution est en général plus lente pour des grandes quantités d'éléments car pour chaque élément elle mesure sa distance avec tous les autres éléments. Elle limite souvent la quantité d'éléments pouvant être analysés. La deuxième approche peut analyser des ensembles de plus grande taille et est en général plus rapide. Les méthodes de groupement se différencient aussi par l'approche utilisée dans la mesure de la distance entre les éléments et / ou les groupes (Schnecke et Boström, 2006).

Gagarin, Makarenkov et Zentilli (2006) analysent trois méthodes : le partitionnement *k-means* (voir MacQueen (1967) et Legendre et Legendre (1998)), la somme des distances inter-cluster moyennes (SASD : « Sum of the Average Squared inside-cluster Distances »), et la distance moyenne entre clusters (AICD : « Average Inter Cluster Distance »). Selon Gagarin, Makarenkov et Zentilli (2006), un des avantages des méthodes de groupement réside dans la sélection de valeurs proches du seuil de sélection de μ-3σ, qui seraient éliminées par la procédure classique mais qui pourraient être intéressantes. En conclusion, les auteurs conseillent d'utiliser les deux premières méthodes, mais nos recherches on également montré que toutes les méthodes de groupement proposées peuvent se montrer inferieures, dans le cas d'utilisation de plateaux de grande taille. Dans ce cas, la recherche d'une séparation significative entre les groupes de données est plus difficile à effectuer. Les méthodes de groupement étant innombrables et dépendant de plusieurs paramètres, il nous est impossible de conclure, de manière générale, à l'utilité ou non de procéder à une recherche des « hits » par ces méthodes.

Pour d'autres approches sur le groupement de données, non spécifiquement liées aux procédures HTS, voir : Arabie, Hubert et De Soete (1996); MacQueen (1967); Legendre et Legendre (1998) et Kaufman et Rousseeuw (1990).

CHAPITRE III

ERREURS SYSTÉMATIQUES

3.1 Types d'erreurs courants en HTS

La procédure de criblage à haut débit possède les inconvénients liés à tout processus de mesure : il existe toujours un écart entre une valeur mesurée et la valeur réelle. Ces écarts peuvent être plus ou moins importants selon la qualité des procédures et des instruments de mesure utilisés. On retrouve toujours des composantes aléatoires (qualifiées de bruit aléatoire) et des composantes systématiques. On peut approximer une mesure par la formule suivante :

$$m' = m + E_s + E_a , \tag{10}$$

où m' est la valeur mesurée, m la valeur réelle, et E_s et E_a les erreurs systématiques et aléatoires, respectivement.

S'il est impossible de corriger les erreurs aléatoires, leur influence est minimisée par leur faible ampleur. Typiquement les erreurs aléatoires ont une distribution normale de moyenne zéro et un écart type de 10 à 20 (Woodward *et al.*, 2006). De plus, leur effets ont tendance à être compensé par le grand nombre de mesures effectuées. Les seules approches pour minimiser leurs effets passent par des

procédures plus strictes, par des tests plus poussés, ainsi que par le double échantillonnage (Brideau *et al.*, 2003; Malo *et al.*, 2006).

Les erreurs systématiques posent plus de problèmes, car leur effet se fait ressentir tout au long de la procédure de criblage et leur ordre de grandeur peut être important. Les sources des erreurs systématiques peuvent être multiples :

- Interaction non prévue entre des composés et les substrats;
- Vieillissement, évaporation des réactifs ou précipitation des cellules;
- Erreurs dues à l'operateur, ou à la non-conformité des procédures;
- Erreurs dans les volumes des liquides, dans le fonctionnement des pipettes, dans les canaux de distribution des liquides, ou apparition de bulles d'air dans les conduits;
- Variations dans les temps d'incubation ou délai dans le traitement des différents plateaux;
- Erreurs dans les appareils de mesure ou positionnement erroné ou décalé;
- Contamination, conditions ambiantes changeantes;
- Etc.

Pour plus de détails sur les sources d'erreurs systématiques, voir : Heuer, Haenel et Prause (2003); Zhang *et al.* (2000); Heyse (2002); Brideau *et al.* (2003); Gunter *et al.* (2003); Harper et Picket (2006); Kevorkov et Makarenkov (2005a); Makarenkov *et al.* (2007).

Un cas relaté par Brideau *et al.* (2003) montre une erreur due au positionnement d'un capteur qui a produit un écart de 14% dans les mesures de la ligne A par rapport à ceux de la ligne P lors d'un criblage HTS (la numérotation des lignes et des colonnes suit le standard du domaine, voir section 1.4, figure 1.3 du présent mémoire pour plus de précisions). Cet article montre aussi des effets dus au délai entre les mesures lors d'une campagne HTS effectuée durant une période de 20 heures (figure 3.1). Il traite aussi des groupements et dérives s'étant produites lors d'un criblage sur plusieurs jours (figure 3.2).

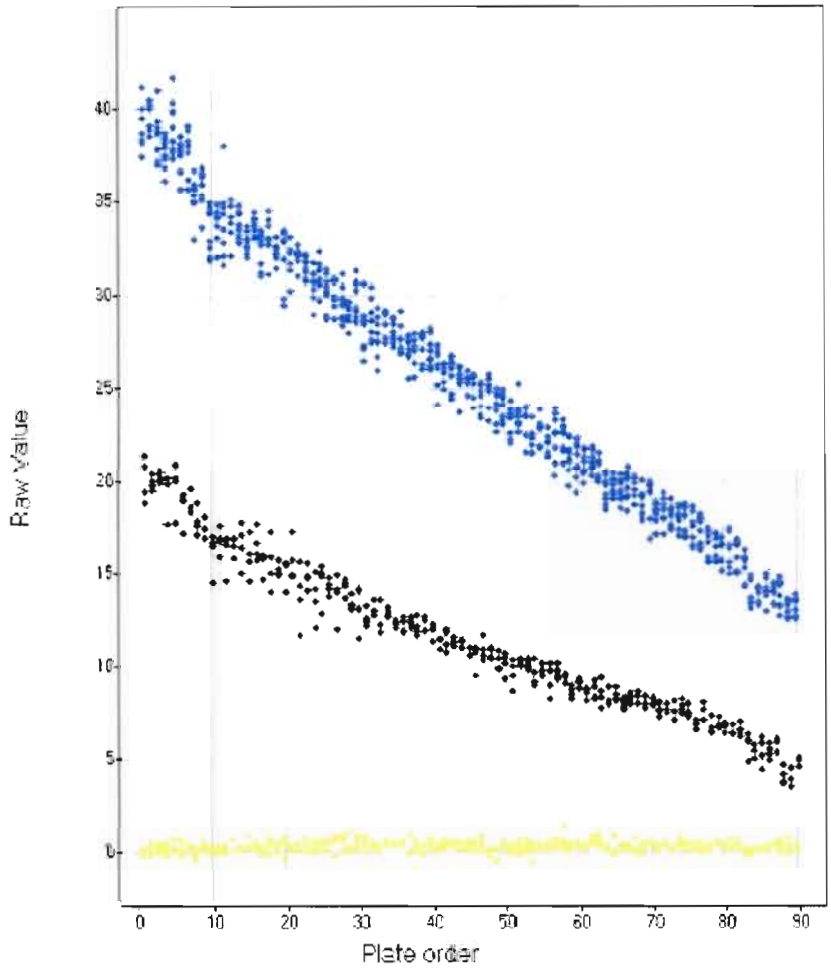


Figure 3.1 : Résultats des mesures sur un criblage de 20 heures. Le groupe supérieur est composé des valeurs de contrôle supérieur. Le groupe inferieur est composé des valeurs de contrôle de l'arrière-plan, et le groupe intermédiaire est composé des valeurs de référence à une concentration IC_{50}. On voit clairement la dérive des valeurs au cours du temps (tiré de Brideau *et al.*, 2003).

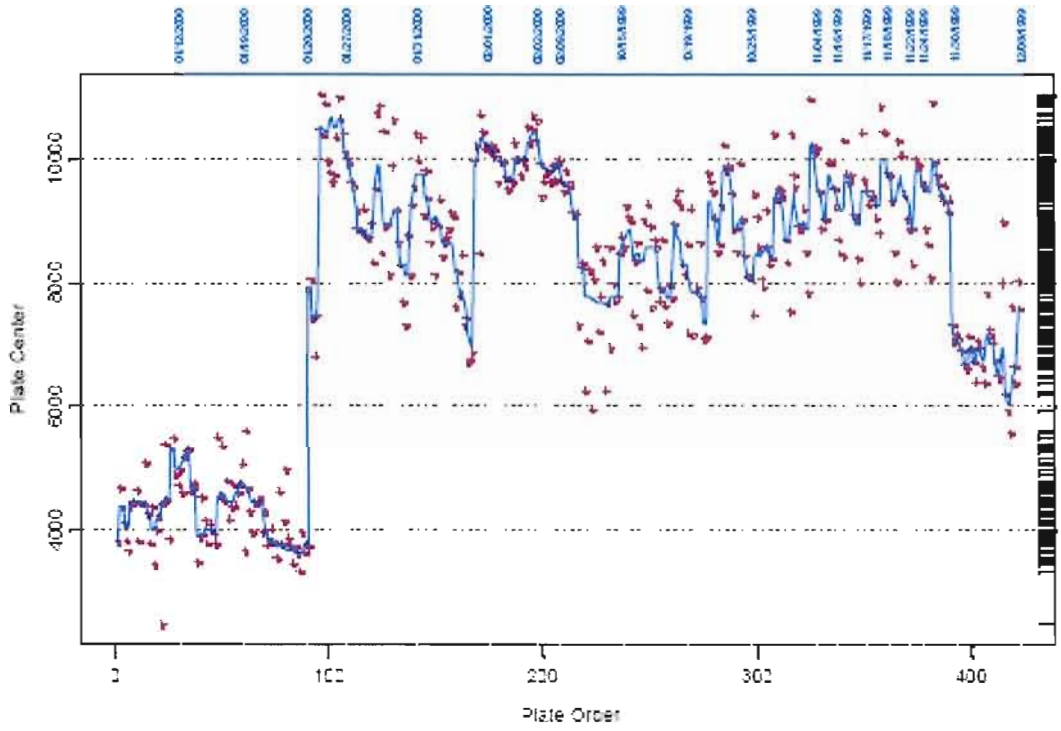


Figure 3.2 : Mesures de la tendance centrale d'une séquence de 423 plateaux lors d'un criblage sur plusieurs jours. On voit l'apparition de groupes de plateaux et de dérives sur certain de ces groupes (tiré de Brideau *et al.*, 2003). Attention : l'ordre indiqué des plateaux ne semble pas correspondre à l'ordre des dates des mesures (aucune indication n'est donnée par les auteurs).

La présence d'erreurs systématiques dans les mesures des composés d'un criblage à haut débit, mène à l'apparition de faux positifs (composés sélectionnés comme « hits » mais qui n'en sont pas) ou de faux négatifs (composés rejetés qui auraient dû être sélectionnés) lors de la recherche des « hits ». Si la présence des premiers n'induit en général qu'un coût supplémentaire (ces composés seront détectés et éliminés dans les étapes subséquentes), les seconds peuvent amener à écarter des composés ou des familles de composés prometteurs. Ceci peut se traduire par la perte d'un médicament potentiel avec les conséquences économiques et surtout « humaines » qui s'en suivent. De plus, les composés écartés lors des étapes primaires de recherche, tel que le criblage HTS, ne sont que rarement ré-testés par la suite, minimisant les chances de « rattrapage » d'un composé éliminé.

Selon Malo *et al.* (2006) et plusieurs autres auteurs, dont Brideau *et al.* (2003), Gunter *et al.* (2003) et Makarenkov *et al.* (2007), la meilleure approche pour réduire l'influence des erreurs systématiques passe par une analyse statistique des résultats. Nos travaux suivent cette voie. En contradiction apparente avec l'affirmation de Rutherford (1871-1937) : « Si votre expérience a besoin de statistiques, vous auriez dû faire une meilleure expérience » (traduction libre), nous pensons que c'est le moyen le plus réaliste d'obtenir une qualité accrue des résultats d'un criblage à haut débit. S'il reste possible de repousser les limites techniques des procédés de mesure, les études statistiques des mesures, aidées par la puissance de calcul des ordinateurs, permettent l'analyse et la correction des données pour un coût réduit et des délais raisonnables. La détection rapide d'erreurs systématiques lors du criblage, peut en outre aider à leur correction. On peut ainsi épargner des coûts et éviter des délais inutiles lors d'une campagne HTS.

3.2 Méthodes de correction

Plusieurs auteurs (Heuer, Haenel et Prause, 2003; Brideau *et al.*, 2003; Kevorkov et Makarenkov, 2005a, 2005b; Malo *et al.*, 2006; Makarenkov *et al.*, 2007; etc.) proposent des méthodes pour détecter et minimiser les effets des erreurs systématiques lors d'un criblage primaire. Les méthodes de « correction » de données présentent un fort intérêt économique, ce qui nous fait penser que plusieurs laboratoires utilisent leurs propres méthodes internes non publiées. De plus, les problèmes de détection et de correction des signaux et des mesures bruités, se retrouvent dans d'autres domaines techniques et scientifiques comme, par exemple, les télécommunications. Chaque domaine propose ses propres solutions, en les adaptant à ses problématiques et besoins particuliers. Il serait intéressant d'étudier les solutions mises en place, pour voir si leur application au criblage à haut débit pourrait être pertinente.

Les approches standard de correction des mesures, dans le domaine du criblage HTS, se basent sur l'utilisation de puits de contrôle, c.-à-d. des puits contenant des valeurs connues et / ou possédant des valeurs extrêmes (contrôles positifs et négatifs) placés sur chaque plateau. Ce sont typiquement des solutions neutres ou des puits vides de toute substance, ou des composés produisant une activité d'environ 67% par rapport aux composés inactifs des plateaux (Woodward *et al.*, 2006).

Une autre approche utilise des plateaux contenant seulement des puits de contrôle. Les mesures effectuées sur ces plateaux permettent d'estimer l'arrière-plan général de la campagne HTS (Fassina, 2006). Son inconvénient majeur est qu'elle ne tient pas compte des éventuelles erreurs qui pourraient apparaître sur les plateaux traités entre les phases de test.

Ces approches donnent des indications générales sur la qualité d'une campagne HTS (voir chapitre 2) et, dans l'ensemble, fonctionnent correctement. Mais les puits de contrôle peuvent aussi être la cible d'erreurs systématiques. Dans ce cas, les « corrections » effectuées par ces méthodes peuvent empirer la qualité des mesures. De plus, les puits de contrôle se situent typiquement dans la périphérie des plateaux, *i.e.*, dans les zones les plus exposées. Les risques d'être affectés par des erreurs systématique est donc plus fort (Brideau *et al.*, 2003; Kevorkov et Makarenkov, 2005a; Malo *et al.*, 2006; Makarenkov *et al.*, 2007; etc.).

Les nouvelles approches, proposées par les auteurs précédemment nommés, se basent plutôt sur des procédures statistiques et ne tiennent pas compte des puits de contrôles situés dans les plateaux. L'avantage de ces méthodes est que chaque donnée « devient » un contrôle, et donc leur nombre croit avec la taille du criblage. Le grand nombre de mesures prises en compte pour les études statistiques, minimise l'influence des possibles erreurs locales. La justification de cette approche se base sur la nature même d'un criblage à haut débit. Lors d'une campagne HTS, la plupart des composés testés ne produisent pas d'interaction avec la cible thérapeutique, c.-à-d., sont inactifs. Les mesures obtenues, lors du criblage, devraient être semblables. Si un écart systématique est détecté entre des plateaux ou à l'intérieur d'un même plateau, celui-ci est probablement dû à une dérive dans le processus de mesure. Sa détection, et sa mesure, doit permettre de minimiser, ou de corriger, son influence et d'améliorer la qualité globale des résultats du criblage.

Pour la pleine efficacité de cette approche, il est important de placer les composés de manière aléatoire, que ce soit par plateau ou par puits, pour l'ensemble des plateaux de la campagne de criblage. Les groupements spatiaux de composés peuvent être faussement détectés comme des biais s'ils ont des interactions semblables avec la cible thérapeutique ou s'ils appartiennent à une même famille. Les méthodes statistiques de correction pourraient être inadéquates dans ces cas.

Correction par soustraction de l'arrière-plan

Une des premières méthodes de correction, développées par notre équipe (voir Kevorkov et Makarenkov, 2005a, 2005b), se basait sur la mesure de l'arrière-plan, ou « background », d'un criblage HTS. L'analyse de plusieurs exemples de campagnes HTS a montré qu'un effet de zone pouvait se produire et se répéter pour l'ensemble des plateaux de la campagne. Le but était d'identifier le patron sous-jacent au criblage pour ensuite l'enlever aux données, lors de l'étape de correction.

Après la normalisation des données, l'arrière-plan moyen de la campagne est calculé comme la moyenne des valeurs par puits :

$$z_i = \frac{1}{N}\sum_{p=1}^{N} x'_{ip}\,, \tag{11}$$

où x'_{ip} est la valeur normalisée du puits i du plateau p, z_i la valeur de l'arrière-plan du puits i et N le nombre total de plateaux.

Pour faciliter la visualisation des tendances dues aux erreurs systématiques, les auteurs proposent le calcul d'une surface approchée de l'arrière-plan au moyen d'une fonction polynomiale (Figures 3.3b).

Pour établir si l'arrière-plan calculé correspond à des erreurs systématiques produisant un effet significatif sur les mesures de la campagne HTS, Kevorkov et Makarenkov (2005a) utilisent deux approches : (1) une analyse, par un test de contingence de χ^2, de la surface des « hits » trouvés avant correction et (2) une analyse des « hits » selon les lignes et les colonnes des plateaux.

Les études de Kevorkov et Makarenkov (2005a, 2005b) traitent un jeu de données d'inhibiteurs potentiels de la *glycosyltransferase MurG function* de *E. coli*.

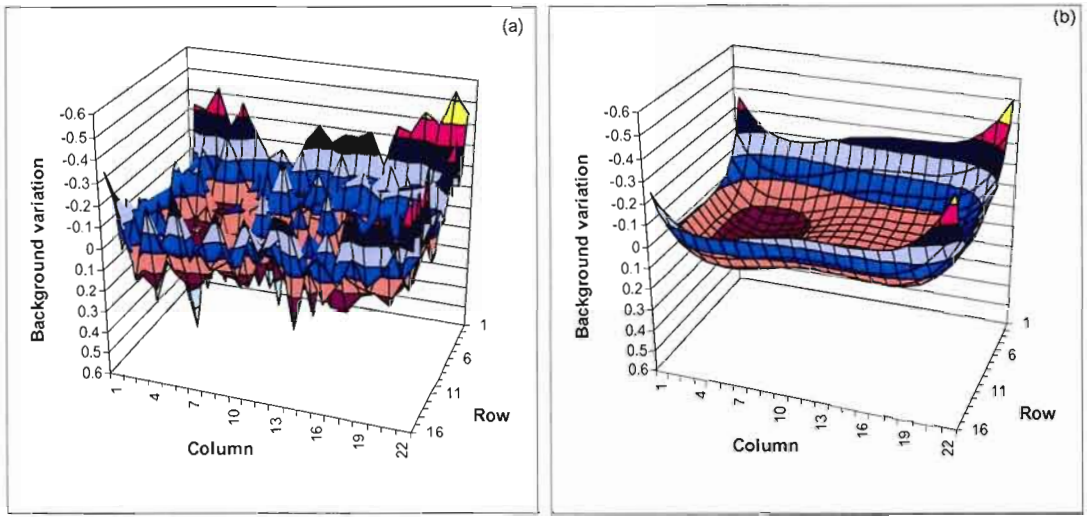


Figure 3.3 : Évaluation de l'arrière-plan pour un criblage d'inhibiteurs de *E. coli* comprenant 164 plateaux (a) et son approximation par une surface polynomiale du quatrième degré (b) (tiré de Kevorkov et Makarenkov, 2005a).

La première approche proposée calcule la surface formée par la moyenne des « hits » trouvés pour chaque puits (avant correction). Intuitivement, si la campagne HTS ne contient pas d'erreurs systématiques, ou si leur influence est faible, la surface définie par l'arrière-plan du criblage devrait s'approcher d'un plan constant. Ceci est dû au fait que la majorité des valeurs devraient être proches de zéro après l'étape de normalisation. La surface obtenue par le calcul des « hits » par puits devrait donc correspondre aussi à un plan constant. Cette surface est analysée par un test de contingence de χ^2, dont l'hypothèse nulle (H_0) stipule que la surface analysée est un plan constant. Si l'hypothèse nulle du test n'est pas rejetée, on en déduit que les erreurs ne sont pas systématiques ou qu'elles sont de faible amplitude et peuvent être ignorées. Dans le cas contraire, il faut appliquer une correction aux données avant la recherche des « hits ».

De même, pour une campagne HTS ne présentant pas d'erreurs systématiques, ou si leur amplitude est faible, le nombre de « hits » par ligne (respectivement par colonne) devrait être semblable pour toutes les lignes (*i.e.* : colonnes, voir figures 3.4 et 3.5). Les études de Kevorkov et Makarenkov (2005a, 2005b) mettent en évidence des problèmes, dans le jeu de données étudié, qui sont dus probablement à des erreurs systématiques lors du criblage. On peut noter que leur influence est plus importante sur les bords des plateaux. Les auteurs montrent également des résultats encourageants. La méthode de correction proposée influence favorablement l'étape de recherche des « hits » dans cette campagne, sans devoir écarter les résultats obtenus préalablement.

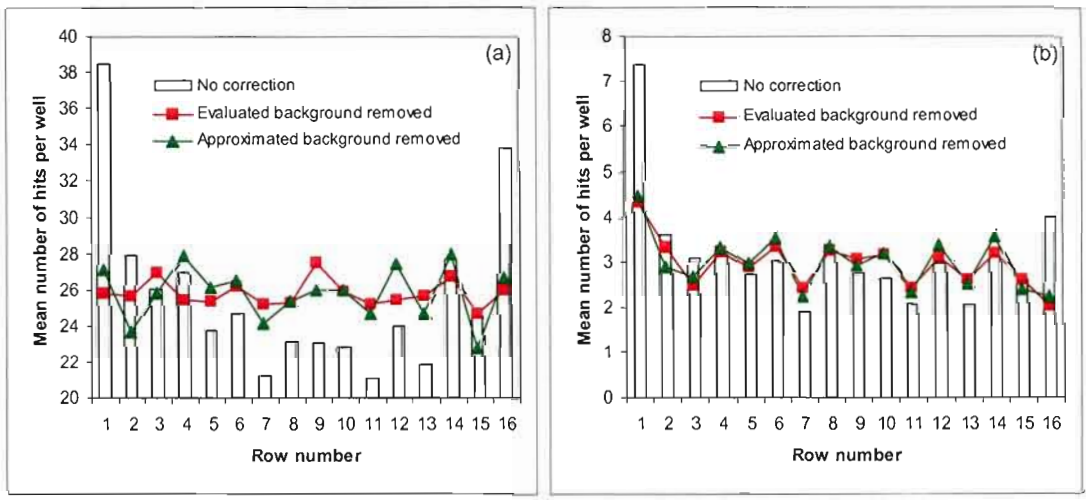


Figure 3.4 : Distribution des « hits » par lignes pour un criblage d'inhibiteurs de *E. coli* comprenant 164 plateaux : (a) sélection des « hits » par un seuil de μ-σ ; (b) sélection des « hits » par un seuil μ-2σ (tiré de Kevorkov et Makarenkov, 2005a).

Des tests similaires, effectués par Garneau et Zentilli (2006, non publié) sur des données provenant d'un criblage réalisé par Hamdan *et al.* (2005), ont aussi montré des biais importants au niveau des colonnes 3 et 4 (Note : la numérotation des colonnes dans ce criblage commençait par la colonne 2, la colonne 1 contenant des valeurs de contrôle). Les valeurs moyennes des mesures de ces colonnes étaient de l'ordre de 8% plus faibles que celles des autres colonnes. Garneau et Zentilli (2006, non publié) ont aussi remarqué que les valeurs dans les colonnes 7 à 11 étaient plus

élevées d'environ 4% par rapport aux autres. Les auteurs ont trouvé également des valeurs d'environ 8% plus élevées que la moyenne sur les lignes C, F et G pour ce criblage (figure 3.5).

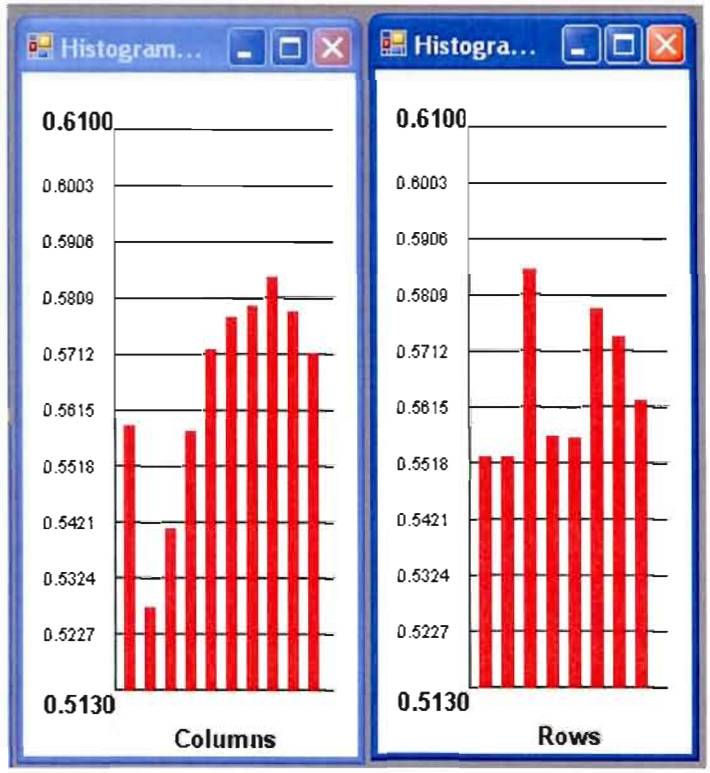


Figure 3.5 : Histogrammes des valeurs moyennes par ligne et par colonne de l'arrière-plan pour le jeu de données non corrigé de Hamdan *et al.* (2005) (réalisé à l'aide de HTS Corrector).

À la vue de ces études, il parait important d'utiliser des méthodes de détection et / ou correction des erreurs systématiques concentrées sur des lignes ou des colonnes avant de faire la sélection des « hits ».

Le calcul de la surface d'arrière-plan peut aussi être utilisé différemment. La surface est calculée pour quelques plateaux vides ou neutres, au début de la procédure HTS. Ceci permet de mettre en évidence des erreurs systématiques inhérentes aux appareils de mesure. La surface calculée sera, par la suite, soustraite aux valeurs

mesurées. Comme pour le cas de l'utilisation de plateaux de contrôle, cité plus haut, cette approche ne met pas en évidence des erreurs qui pourraient se produire plus tard lors de la campagne HTS et, selon Brideau *et al.* (2003), elle ne fait pas trop de différence dans la pratique.

Correction par « median polish »

Brideau *et al.* (2003), Gribbon et Sewing (2005) et Malo *et al.* (2006) proposent une méthode de détection et correction des erreurs systématiques basée sur l'algorithme « median polish » à deux critères de Tukey (voir aussi le chapitre 2). « Median polish », ou lissage par médianes mobiles, est utilisé en statistique pour trouver une approximation de données dans une matrice à deux critères. Cette méthode permet de corriger des tendances systématiques se produisant sur des lignes ou des colonnes de données présentées sous forme matricielle.

L'argumentation des auteurs proposant une correction par « median polish », se base sur la robustesse de cette méthode. L'utilisation des médianes, au lieu des moyennes, la rendant moins sensible aux valeurs extrêmes. Les médianes n'étant pas influencées par ces valeurs, la tendance centrale (ou « centre ») des données, définie par leur valeur médiane, est sensiblement plus « proche » des valeurs inactives que ne le serait la moyenne, maximisant ainsi la différence entre les valeurs inactives et les valeurs actives (les « hits » recherchés). La correction par « median polish » présente en outre l'avantage de gérer facilement les puits vides, qui sont fréquents dans la pratique du criblage à haut débit (Brideau *et al.*, 2003).

La division, par le facteur MAD, des résidus résultant de la méthode « median polish » (voir équations 7 et 8 du chapitre 2), nous permet de calculer le B score (ou « Better » scores) selon la nomenclature proposée par Brideau *et al.* (2003).

Un des défauts de cette méthode, signalé par les auteurs, est qu'elle est sensible à la « randomisation » des échantillons dans les plateaux. Si, par exemple, on regroupe des composés d'une même famille sur une ligne (ou colonne), et que ces composés possèdent des valeurs significativement inferieures au reste des composés du plateau (pour un criblage d'inhibition), la ligne (respectivement, la colonne) sera considérée comme étant affectée par une erreur systématique et sera « corrigée », alors que cette famille pourrait être une source prometteuse de « hits ».

Makarenkov *et al.* (2007) ont effectué des simulations en utilisant la méthode « median polish » comme méthode de correction, pour des données simulées auxquelles ils avaient ajouté des erreurs systématiques sur les lignes et les colonnes. Du fait que « median polish » enlève les effets de ce type d'erreurs, les auteurs s'attendaient à voir une amélioration des résultats lors de la recherche des « hits », tel que suggéré par les auteurs proposant la méthode (Brideau *et al.*, 2003). Comme indiqué dans le chapitre 2 et dans les conclusions de Makarenkov *et al.* (2007), les résultats ne se sont pas montrés concluants. Dès que les erreurs devenaient pour le moins complexes, la « correction » induisait plus d'erreurs dans la recherche des « hits » que les méthodes conventionnelles sans correction.

Correction par puits

Au vu de ces problèmes, notre équipe a développé une nouvelle méthode appelée *correction par puits* (ou « well correction »). Cette méthode se différencie des autres méthodes de correction rencontrées, par un traitement des puits le long du criblage HTS, et n'utilise pas un traitement des données plateau par plateau. Nous en faisons ici une brève présentation. Elle est détaillée dans l'article de Makarenkov *et al.* (2007) présenté au chapitre 4.

Pour mettre au point cette méthode, nous avons fait les hypothèses habituelles, soit :

- Les échantillons étudiés peuvent être divisés en deux groupes, les composés actifs et les composés inactifs;
- La plupart des échantillons sont inactifs;
- Les valeurs des échantillons actifs diffèrent significativement des valeurs des échantillons inactifs;
- Les échantillons sont manipulés dans des plateaux bidimensionnels et sont traités en séquence;
- Les erreurs systématiques produisent une influence répétée dans les mesures de tous les plateaux du criblage;
- Les échantillons sont disposés de façon aléatoire dans les plateaux;
- Les données et les erreurs sont distribuées selon une loi normale.

Les trois premières hypothèses permettent de séparer les échantillons en deux groupes, le plus important contenant les composés inactifs. Les valeurs mesurées pour ce groupe doivent être similaires et leur variabilité ne devrait être causée que par des erreurs systématiques ou aléatoires. Les erreurs aléatoires se compensent à cause de la disposition aléatoire des composés dans les plateaux et à travers le criblage HTS. Leur influence sur les résultats des mesures devrait être minime.

La méthode proposée comporte une étape de normalisation centrée réduite des valeurs obtenues lors du criblage. Cette étape est indispensable pour permettre leur comparaison. On peut précéder, au préalable, à une élimination des valeurs extrêmes (*i.e.* « outliers ») si nécessaire.

Nous partons de la proposition que, le long d'un puits, les données d'une campagne HTS devraient être également centrées réduites (après l'étape de normalisation faite précédemment). Nous pensons également que si des erreurs systématiques se produisent dans un puits donné, elles risquent de se retrouver dans le même puits à travers tout le criblage. Il reste à tester d'autres approches pour la simulation des erreurs systématiques, notamment en les faisant varier pour chaque plateau.

Nous avons étudié des données provenant d'un jeu de données proposé par l'Université McMaster dans le cadre d'un concours, et portant sur un criblage effectué au moyen de 1250 plateaux (disponibles sur le site de l'Université à l'adresse suivante : http://hts.mcmaster.ca/HTSDataMiningCompetition.htm). Ces données montraient des biais le long des puits, après normalisation. Par exemple, nous avons remarqué un décalage de la moyenne attendue (zéro) dans le puits défini par la première colonne et la 8ème ligne (figure 3.6). Des biais vers le bas et vers le haut, selon l'ordre de traitement des différents plateaux, ont aussi été trouvés (figure 3.7 a et b).

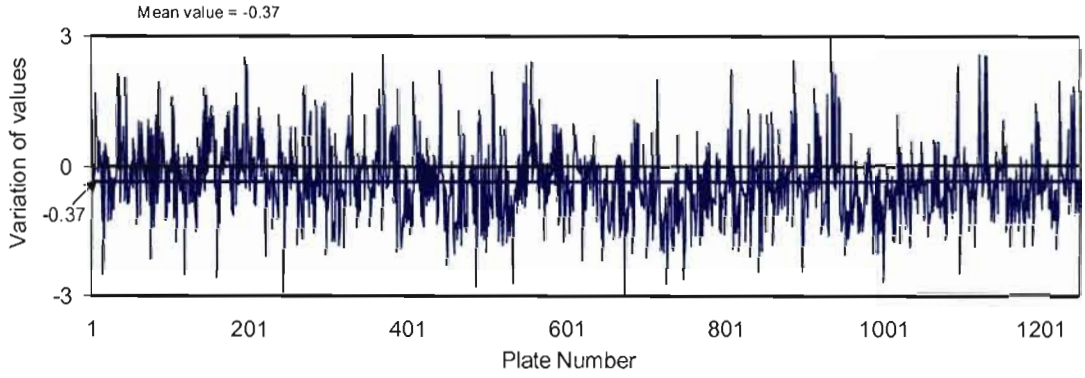


Figure 3.6 : Écart par rapport à la moyenne des valeurs normalisées, pour le puits formé par la colonne 1 et la ligne 8 dans le jeux de données de l'Université McMaster portant sur 1250 plateaux (tiré de Makarenkov *et al.*, 2007).

Pour corriger ces erreurs, nous avons procédé à une approximation linéaire des valeurs de chaque puits. Nous avons ensuite soustrait ces approximations aux valeurs normalisées des puits. La dernière étape de la méthode consiste en une

nouvelle normalisation centrée réduite pour chaque puits. Cette normalisation, s'effectuant sur un grand nombre de valeurs (selon le nombre de plateaux, soit 1250 pour le jeu de données considéré), est peu sensible aux valeurs extrêmes qui pourraient rester après la normalisation par plateaux (seulement 80 éléments par plateau).

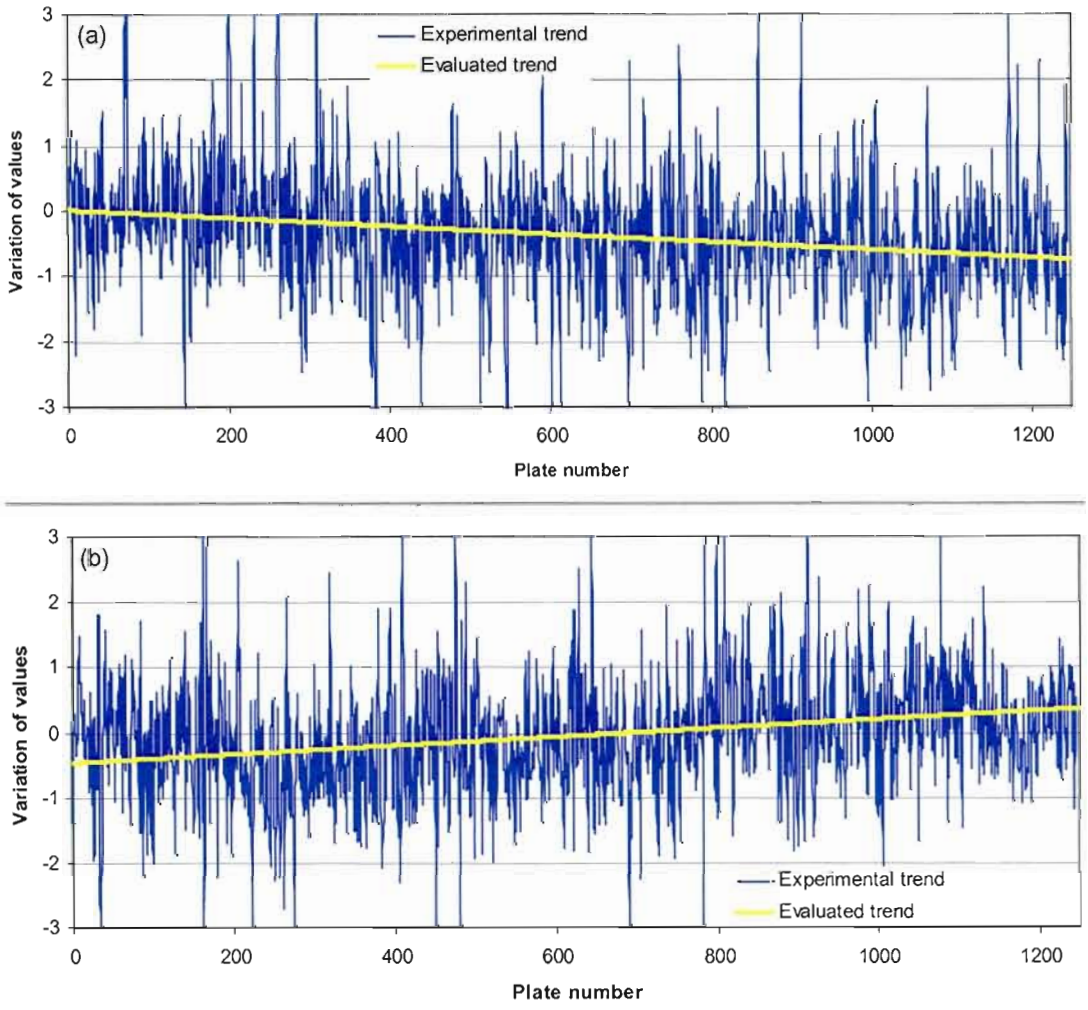


Figure 3.7 : Biais dans les valeurs normalisées des puits dans le jeux de données de l'Université McMaster : (a) biais descendant et (b) biais ascendant (tiré de Makarenkov *et al.*, 2007).

Nos travaux, tant sur les données fournies par l'Université McMaster que sur des données aléatoires simulées, ont montré une nette amélioration des résultats

obtenus lors de la phase de recherche des « hits ». Le nombre de faux positifs et de faux négatifs, trouvés après la correction par puits, était nettement inferieur à ceux trouvés sans correction ou par des corrections utilisant l'approche « median polish ». Le taux de détection des « hits » était supérieur aux méthodes concurrentes tout en étant constant pour différents scenarios d'ajouts d'erreurs (figure 3.8).

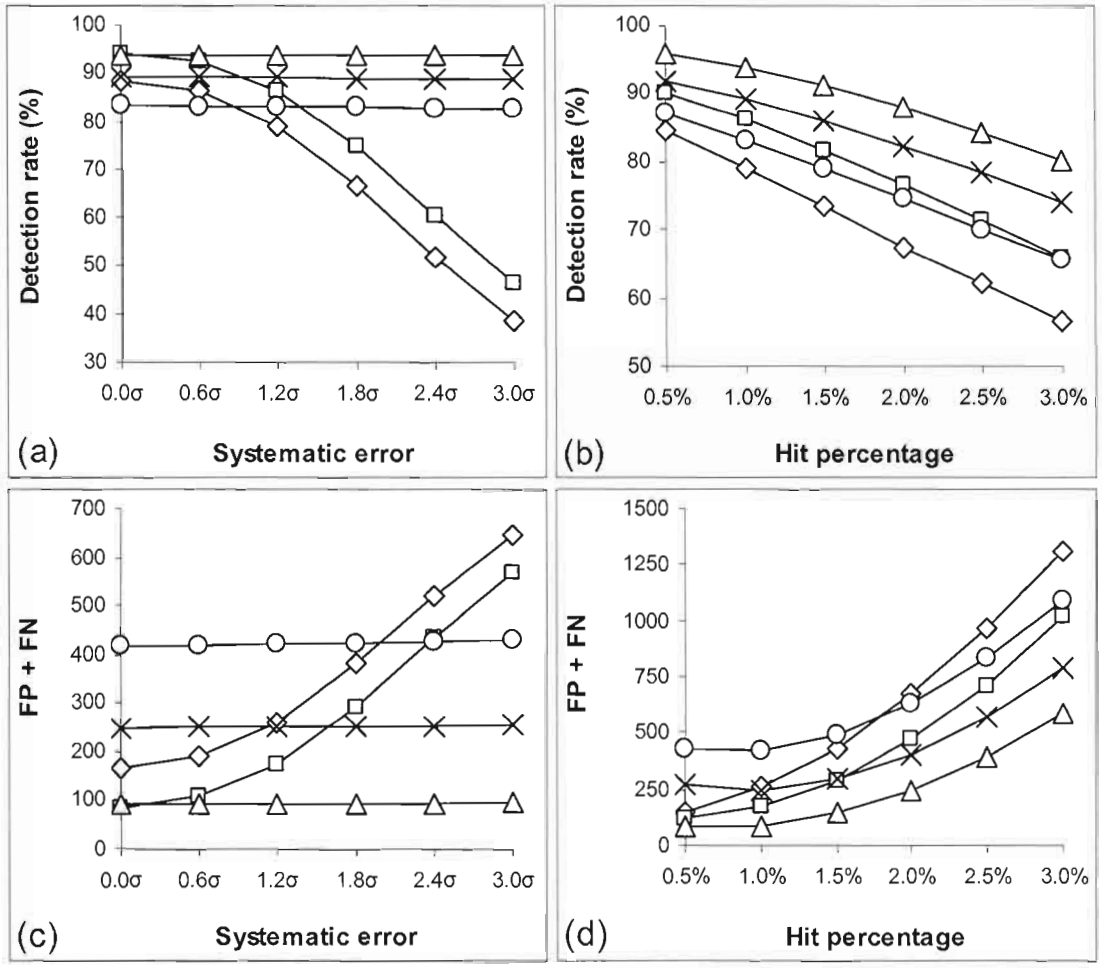


Figure 3.8 : Taux de détection des « hits » et nombre total de faux positifs et faux négatifs calculés pour des données simulées, selon une loi normale N(0, 1), en variant le nombre d'erreurs (a et c) ou de « hits » (b et d) ajoutés. Détection pour un seuil de μ-3σ sans correction : (◊) par plateau, (❑) pour le criblage complet. Détection pour un seuil de μ-3σ après correction : (X) par « median polish », (○) par B score, et (Δ) par correction par puits (tiré de Makarenkov *et al.*, 2007).

Les tests que nous avons effectués suggèrent que la méthode de correction par puits est préférable aux autres méthodes mentionnées dans ce chapitre. Ses performances sont bonnes même si le criblage ne présente pas d'erreurs systématiques, ce qui permet son utilisation par défaut. Néanmoins nous ne conseillons pas l'application de méthodes de correction sans une étude préalable pour savoir si les données sont entachées d'erreurs ou non.

Autres méthodes de correction

Comme signalé dans l'introduction de ce chapitre, d'autres méthodes de correction sont sûrement utilisées dans l'industrie, sans être publiées vu leur intérêt économique. Par exemple, Heuer, Haenel et Prause (2003) parlent d'une méthode utilisant une approche par intelligence artificielle mais n'en donnent que très peu de détails.

Dans la pratique, il semble que peu de laboratoires effectuent une véritable correction des résultats obtenus. L'usage semble privilégier le repérage visuel des possibles erreurs et la reprise des mesures, si les valeurs ne correspondent pas au patron attendu. Nous avons notamment été stupéfaits par des mesures faites dans le criblage réalisé par Hamdan *et al.* (2005) et étudié par Garneau et Zentilli (2006, non publié). La détection des biais et le choix des « hits » étaient effectués au moyen de calculs réalisés sur des feuilles Excel. Après un contrôle visuel des résultats numériques, les auteurs choisissaient de refaire les mesures pour certains plateaux présentant des données non-conformes à leurs attentes. Rien ne laissait croire qu'ils aient identifié les biais systématiques que nous avons trouvés. De plus, certaines feuilles Excel comportaient des erreurs grossières qui pouvaient altérer la procédure du choix des « hits ». Si cela ne semble pas avoir eu une importance cruciale dans leur étude, nous ne pouvons nous empêcher de penser qu'une approche plus rigoureuse, et partiellement automatisée, aurait eu au moins l'avantage de leur faire

gagner du temps. A quoi sert d'accélérer et d'automatiser la procédure de criblage si, par la suite, l'étude des données est faite visuellement?

Au vu de nos études, il nous semble important de mettre en place des procédures, automatisées et standardisées, de détection et / ou correction des mesures. Il est possible, notamment, d'utiliser, en parallèle, plusieurs de ces méthodes pour faire un choix plus judicieux des composées qui seront traités dans les phases subséquentes du cycle de recherche des médicaments. On pourrait tester en priorité les éléments ayant été signalés comme « hits » par deux (ou plus) méthodes différentes, pouvant utiliser ou non des algorithmes de correction.

Nous pensons qu'un effort devrait être fait dans la recherche des paramètres permettant des simulations plus réalistes des données d'un criblage HTS. Il nous semble primordial de tester, au moyen de données simulées, réalistes et standardisées, les méthodes, actuelles ou futures, d'analyse et de correction des données, comme celles proposées pour la recherche des « hits ». Cette approche permettrait de porter un jugement éclairé sur leur pertinence et leur efficacité.

CHAPITRE IV

ARTICLES

1. « Comparison of two methods for detecting and correcting systematic error in high-throughput screening data »

2. « HTS-Corrector: software for the statistical analysis and correction of experimental high-throughput screening data »

3. « Using Clustering Techniques to Improve Hit Selection in High-Throughput Screening »

4. « An efficient method for the detection and elimination of systematic error in high-throughput screening »

4.1 « Comparison of two methods for detecting and correcting systematic error in high-throughput screening data »

Cet article compare deux techniques de correction des erreurs systématiques pouvant être utilisées lors de campagnes de criblage HTS. Les méthodes étudiées sont : la *correction par soustraction de l'arrière-plan* (« background correction ») et la *correction par puits* (« well correction »). Elles ont été proposées et mises au point par les membres de notre équipe (voir aussi chapitre 3).

L'article présente brièvement les deux méthodes et les résultats obtenus par leur application à un jeu de données type, composé de 1250 plateaux, d'un test sur l'inhibition de *l'Escherichia coli dihydrofolate reductase* (soit les données fournies pour la compétition de l'Université McMaster). Des tests ont été aussi effectués sur des données simulées.

L'article conclut en une amélioration des résultats, lors de la sélection des « hits », après l'utilisation des méthodes de correction. La méthode de correction par puits obtenant de meilleurs résultats par rapport à celle par soustraction de l'arrière-plan.

Gagarin A., Kevorkov D., Makarenkov V. et Zentilli P. (2006) « Comparison of two methods for detecting and correcting systematic error in high-throughput screening data. » In *Data Science and Classification*, IFCS 2006, 241-249.

Comparison of two methods for detecting and correcting systematic error in high-throughput screening data

Andrei Gagarin[1], Dmytro Kevorkov[1], Vladimir Makarenkov[2], and Pablo Zentilli[2]

[1] Laboratoire LaCIM, Université du Québec à Montréal,
C.P. 8888, Succ. Centre-Ville, Montréal (Québec), Canada, H3C 3P8
[2] Département d'Informatique, Université du Québec à Montréal,
C.P. 8888, Succ. Centre-Ville, Montréal (Québec), Canada, H3C 3P8

Abstract. High-throughput screening (HTS) is an efficient technological tool for drug discovery in the modern pharmaceutical industry. It consists of testing thousands of chemical compounds per day to select active ones. This process has many drawbacks that may result in missing a potential drug candidate or in selecting inactive compounds. We describe and compare two statistical methods for correcting systematic errors that may occur during HTS experiments. Namely, the collected HTS measurements and the hit selection procedure are corrected.

1 Introduction

High-throughput screening (HTS) is an effective technology that allows for screening thousands of chemical compounds a day. HTS provides a huge amount of experimental data and requires effective automatic procedures to select active compounds. At this stage, active compounds are called hits; they are preliminary candidates for future drugs. Hits obtained during primary screening are initial elements for the determination of activity, specificity, physiological and toxicological properties (secondary screening), and for the verification of structure-activity hypotheses (tertiary screening) (Heyse (2002)).

However, the presence of random and systematic errors has been recognized as one of the major hurdles for successful implementing HTS technologies (Kaul (2005)). HTS needs reliable data classification and quality control procedures. Several methods for quality control and correction of HTS data have been recently proposed in the scientific literature. See for example the papers of Zhang et al. (1999), Heyse (2002), Heuer et al. (2003), and Brideau et al. (2003).

There are several well-known sources of systematic error (Heuer et al. (2003)). They include reagents evaporation or decay of cells which usually show up as smooth trends in the plate mean or median values. Another typical error can be caused by the liquid handling or malfunctioning of pipettes. Usually this generates a localized deviation of expected values. A variation

in the incubation time, a time drift in measuring different wells or different plates, and reader effects may appear as smooth attenuations of measurements over an assay. This kind of effects may have a significant influence on the selection process of active compounds. They can result in an underestimation (false negative hits) or overestimation (false positive hits) of the number of potential drug targets.

We have developed two methods to minimize the impact of systematic errors when analyzing HTS data. A systematic error can be defined as a systematic variability of the measured values along all plates of an assay. It can be detected, and its effect can be removed from raw data, by analyzing the background pattern of plates of the same assay (Kevorkov and Makarenkov (2005)). On the other hand, one can adjust the data variation at each well along the whole HTS assay to correct the traditional hit selection procedure (Makarenkov et al. (2006)). Methods described in Sections 3 and 4 originate from the two above-mentioned articles.

2 HTS procedure and classical hit selection

An HTS procedure consists of running samples (i.e. chemical compounds) arranged in 2-dimensional plates through an automated screening system that makes experimental measurements. Samples are located in wells. The plates are operated in sequence. Screened samples can be divided into active (i.e. hits) and inactive ones. Most of the samples are inactive, and the measured values for the active samples are significantly different from the inactive ones. In general, samples are assumed to be located in a random order, but it is not always the case in practice.

The mean values and standard deviations are calculated separately for each plate. To select hits in a particular plate, one usually takes the plate mean value μ and its standard deviation σ to identify samples whose values differ from the mean μ by at least $c\sigma$, where c is a preliminary chosen constant. For example, in the case of an inhibition assay, by choosing $c = 3$, we would select samples with the values lower than $\mu - 3\sigma$. This is the simplest and most widely-known method of hit selection. This method is applied on a plate-by-plate basis.

3 Correction by removing the evaluated background

This correction method is a short overview of the corresponding procedure of Kevorkov and Makarenkov (2005). To use it properly, we have to assume that all samples are randomly distributed over the plates and systematic error causes a repeatable influence on the measurements in all plates. Also, we have to assume that the majority of samples are inactive and that their average values measured for a large number of plates are similar. Therefore, the average variability of inactive samples is caused mainly by systematic

error, and we can use them to compute the assay background. In the ideal case, the measurements background surface is a plane, but systematic errors can introduce local fluctuations in it. The background surface and hit distribution surface of an assay represent a collection of scalar values which are defined per well and are plotted as a function of the well coordinates in a 3-dimentional diagram.

An appropriate statistical analysis of experimental HTS data requires a preprocessing. This will ensure the meaningfulness and correctness of the background evaluation and hit selection procedures. Therefore, we use normalization by plate and exclude outliers from the computations. Keeping in mind the assumptions and pre-procession requirements, the main steps of this method can be outlined as follows:

- Normalization of experimental HTS data by plate,
- Elimination of outliers from the computation (optional),
- Topological analysis of the evaluated background,
- Elimination of systematic errors by subtracting the evaluated background surface from normalized raw data,
- Selection of hits in the corrected data.

3.1 Normalization

Plate mean values and standard deviations may vary from plate to plate. To compare and analyze the experimental data from different plates, we need first to normalize all measurements within each plate.

To do this, we use classical *mean centering and unit variance standardization* of the data. Specifically, to normalize the input measurements, we apply the following formula:

$$x'_i = \frac{x_i - \mu}{\sigma}, \tag{1}$$

where x_i, $i = 1, 2, \ldots, n$, is the input element value, x'_i, $i = 1, 2, \ldots, n$, is the normalized output element value, μ is the plate mean value, σ is the plate standard deviation, and n is the total number of elements (i.e. number of wells) in each plate. The output data will have the plate mean value $\mu' = 0$ and the plate standard deviation $\sigma' = 1$.

Another possibility discussed by Kevorkov and Makarenkov (2005) is to normalize all the plate values to a given interval. This normalization generally produces results similar to the described one.

3.2 Evaluated background

Systematic error is assumed to appear as a mean fluctuation over all plates. Therefore, an assay background can be defined as the mean of normalized plate measurements, i.e.:

$$z_i = \frac{1}{N} \sum_{j=1}^{N} x'_{i,j}, \tag{2}$$

where $x'_{i,j}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, N$, is the normalized value at well i of plate j, z_i is the background value at well i, and N is the total number of plates in the assay.

Clearly, Formula 2 is more meaningful for a large number of plates: in this case the values of inactive samples will compensate the outstanding values of hits. To make Formula 2 useful and more accurate for an assay with a small number of plates, one can exclude hits and outliers from the computations. Thus, the evaluated background will not be influenced by the outstanding values and will better depict systematic errors.

3.3 Subtraction of evaluated background

Analyzing the distribution of selected hits, we can tell whether any systematic error is present or not in the assay: hits should be more or less evenly distributed over all wells. Otherwise, the hit amounts vary substantially from one well to another indicating the presence of systematic errors.

Deviations of the evaluated background surface from the zero plane indicate an influence of systematic errors on the measured values. Therefore, it is possible to correct raw HTS data by subtracting the evaluated background, defined by Formula 2, from the normalized values of each plate, given in Formula 1. After that, we can reassess the background surface and hit distribution again.

4 Well correction method

This section is a concise description of the well correction approach presented in detail in Makarenkov et al. (2006). We have to make the assumptions stated in the previous section about input HTS data and positions of samples in wells. The main steps of the well correction method are the following:

- Normalization of all sample values by plate,
- Analysis of hit distribution in the raw data,
- Hit and outlier elimination (optional),
- Correction and normalization of samples by well,
- Normalization of all samples by plate,
- Selection of hits in the corrected data.

Similarly to the evaluated background approach, the normalization of all samples by plate is done here using the mean centering and unit variance standardization procedure described above. The hit distribution surface can be computed as a sum of selected hits by well along the whole assay. If this surface is significantly different from a plane, it implies the presence of systematic errors in the assay measurements. Excluding hits and outliers from the computation, we obtain the non-biased estimates for the mean values and standard deviations of inactive samples in plates.

4.1 Well correction technique

Once the data are plate-normalized, we can analyze their values at each particular well along the entire assay. The distribution of inactive measurements (i.e. excluding hits and outliers) along wells should be zero-mean centered if systematic error is absent in the dataset.

However, a real distribution of values by well can be substantially different from the ideal one. Such an example is shown in the article by Makarenkov et al. (2006). A deviation of the well mean values from zero indicates the presence of systematic errors. Experimental values along each well can have ascending and descending trends (Makarenkov et al. (2006)). These trends can be discovered using the linear least-squares approximation (e.g. the trends can be approximated by a straight line).

In the case of approximation by a straight line ($y = ax + b$), the line-trend is subtracted from or added to the initial values bringing the well mean value to zero (x denotes the plate number, and y is the plate-normalized value of the corresponding sample). For the analysis of large industrial assays, one can also use some non-linear functions for the approximation. On the other hand, an assay can be divided into intervals and a particular trend function characterizing each interval can be determined via an approximation. After that, the well normalization using the mean centering and unit variance standardization procedure is carried out. Finally, we normalize the well-corrected measurements in plates and reexamine the hit distribution surface.

5 Results and Conclusion

To compare the performances of the two methods described above, we have chosen an experimental assay of the HTS laboratory of McMaster University (`http://hts.mcmaster.ca/Competition_1.html`). These data consist of a screen of compounds that inhibit the *Escherichia coli* dihydrofolate reductase. The assay comprises 1250 plates. Each plate contains measurements for 80 compounds arranged in 8 rows and 10 columns. A description of the hit follow-up procedure for this HTS assay can be found in Elowe et al. (2005).

Table 1 shows that the proposed correction methods have slightly increased the number of selected hits. However, the standard deviation of selected hits by well and the χ-square values (obtained using the χ-square contingency test with α-parameter equal to 0.01; the null hypothesis, H_0, here is that the hit distribution surface is a constant plane surface) become smaller after the application of the correction procedures. Moreover, the well correction method allowed the corresponding hit distribution surface to pass the χ-square contingency test in both cases (using 2.5σ and 3σ thresholds for hit selection). Figure 1 shows that the hit distribution surfaces have become closer to planes after the application of the correction methods.

To demonstrate the effectiveness of the proposed correction procedures, we have also conducted simulations with random data. Thus, we have con-

	Raw data	Rem. backgr.	Well correct.	Raw data	Rem. backgr.	Well correct.
Hit selection threshold	3σ	3σ	3σ	2.5σ	2.5σ	2.5σ
Mean value of hits per well	3.06	3.13	3.08	6.93	6.93	7.03
Standard deviation	2.17	2.16	2.06	3.93	3.55	2.61
Min number of hits per well	0	0	0	1	2	2
Max number of hits per well	10	10	10	19	22	15
χ-square value	121.7	118	109.1	175.8	143.8	76.6
χ-square critical value	111.14	111.14	111.14	111.14	111.14	111.14
χ-square contingency H_0	No	No	Yes	No	No	Yes

Table 1. Results and statistics of the hit selection carried out for the raw, background removed (Rem. backgr.) and well-corrected (Well correct.) McMaster data.

sidered random measurements generated according to the standard normal distribution. The randomly generated dataset also consisted of 1250 plates having wells arranged in 8 rows and 10 columns. The initial data did not contain any hit. However, the traditional hit selection procedure has found 119 false positive hits in the random raw data using the 3σ threshold. The correction methods detected 117 (removed background) and 104 (well correction) false positive hits.

Then, we have randomly added 1% of hits to the raw random data. The hit values were randomly chosen from the range $[\mu - 3.5\sigma; \mu - 4.5\sigma]$, where μ denotes the mean value and σ denotes the standard deviation of the observed plate. After that, the data with hits were modified by adding the values $4c, 3c, 2c, c, 0, 0, -c, -2c, -3c$, and $-4c$ to the 1st, 2nd, ..., and 10th columns, respectively, thus simulating a systematic error in the assay, where the variable c was consequently taking values $0, \sigma/10, 2\sigma/10, \ldots$, and $5\sigma/10$. The value $c = 0$ does not create any systematic error, but bigger values of c increase systematic error proportionally to the standard deviation σ.

For each value of the noise coefficient c, hits were selected in the raw, background removed and well-corrected datasets using the 3σ threshold. The hit detection rate as well as the false positive and false negative rates were assessed. The hit detection rate was generally higher for both corrected datasets. Figure 2(a) shows that the background and well correction procedures successfully eliminated systematic error from the random data. Both methods were robust and showed similar results in terms of the hit detection rate. However, the well correction method systematically outperformed the background method in terms of the false positive hit rate (see Figure 2(b)).

In conclusion, we developed two statistical methods that can be used to refine the analysis of experimental HTS data and correct the hit selection procedure. Both methods are designed to minimize the impact of systematic error in raw HTS data and have been successfully tested on real and artificial datasets. Both methods allow one to bring the hit distribution surface closer to a plane surface. When systematic error was not present in the data, both correcting strategies did not deteriorate the results shown by the tra-

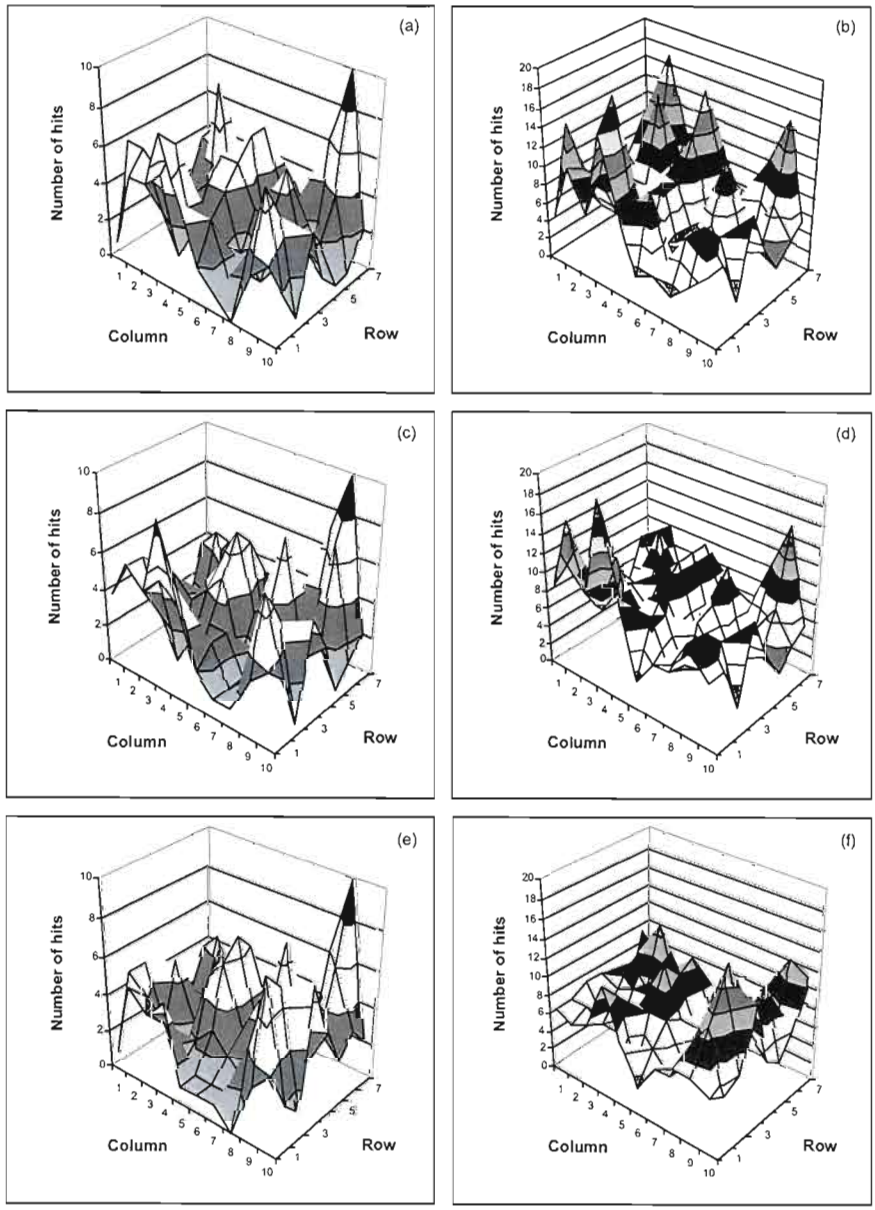


Fig. 1. Hit distribution surfaces computed for the 3σ and 2.5σ hit selection thresholds for the raw (a and b), background removed (c and d), and well-corrected (e and f) McMaster datasets.

ditional approach. Thus, their application does not introduce any bias into the observed data. During the simulations with random data, the well correction approach usually provided more accurate results than the algorithm proceeding by the removal of evaluated background.

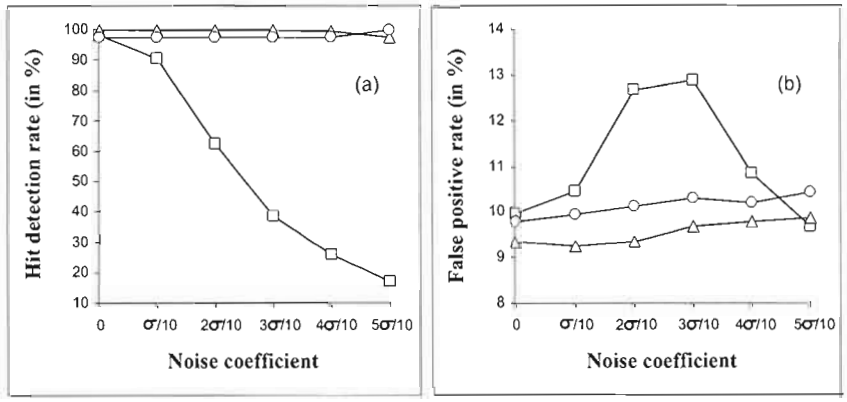


Fig. 2. Correct (a) and false positive (b) detection rates for the noisy random data obtained by the traditional hit selection procedure (denoted by □), the removed background (denoted by ○), and well correction (denoted by △) methods.

References

BRIDEAU, C., GUNTER, B., PIKOUNIS, W. and LIAW, A. (2003): Improved statistical methods for hit selection in high-throughput screening. *Journal of Biomolecular Screening, 8, 634-647.*

ELOWE, N.H., BLANCHARD, J.E., CECHETTO, J.D. and BROWN, E.D. (2005): Experimental screening of dihydrofolate reductase yields a "test set" of 50,000 small molecules for a computational data-mining and docking competition. *Journal of Biomolecular Screening, 10, 653-657.*

HEUER, C., HAENEL, T., PRAUSE, B. (2003): A novel approach for quality control and correction of HTS data based on artificial intelligence. *The Pharmaceutical Discovery & Development Report.* PharmaVentures Ltd. [Online].

HEYSE, S. (2002): Comprehensive analysis of high-throughput screening data. In: *Proceedings of SPIE 2002*, 4626, 535-547.

KAUL, A. (2005): The impact of sophisticated data analysis on the drug discovery process. *Business Briefing: Future Drug Discovery 2005.* [Online]

KEVORKOV, D. and MAKARENKOV, V. (2005): Statistical analysis of systematic errors in HTS. *Journal of Biomolecular Screening, 10, 557-567.*

MAKARENKOV, V., KEVORKOV, D., GAGARIN, A., ZENTILLI, P., MALO, N. and NADON, R. (2006): An efficient method for the detection and elimination of systematic error in high-throughput screening. Submitted.

ZHANG, J.H., CHUNG, T.D.Y. and OLDENBURG, K.R. (1999): A Simple Statistic Parameter for Use in Evaluation and Validation of High Throughput Screening Assays. *Journal of Biomolecular Screening, 4, 67-73.*

4.2 « HTS-Corrector: software for the statistical analysis and correction of experimental high-throughput screening data »

Cet article présente les fonctionnalités de la première version du logiciel « HTS Corrector » développé par notre équipe. Cette version du logiciel comprenait les premières méthodes de correction de criblages à haut débit (HTS) proposées par notre équipe : la méthode de *correction par soustraction de l'arrière-plan* et la méthode de *correction par puits*. Le logiciel proposait aussi une méthode en cours de validation, basée sur l'adaptation des seuils de recherche des « hits » par puits au lieu d'utiliser un seuil unique pour l'ensemble du criblage. La variabilité des mesures à travers des groupes de plateaux de composés, lors de la procédure de criblage, pouvant être importante, une procédure de groupement par *k-means* était aussi proposée en option par le logiciel. Des méthodes de visualisation des données faisaient aussi partie de cette version développée en C++.

Makarenkov V., Kevorkov D., Zentilli P., Gagarin A., Malo N. et Nadon R. (2006) « HTS-Corrector: software for the statistical analysis and correction of experimental high-throughput screening data. » *Bioinformatics*, **22**, 1408-1409.

APPLICATIONS NOTE Vol. 22 no. 11 2006, pages 1408–1409
doi:10.1093/bioinformatics/btl126

Systems biology

HTS-Corrector: software for the statistical analysis and correction of experimental high-throughput screening data

Vladimir Makarenkov[1,*], Dmytro Kevorkov[1], Pablo Zentilli[1], Andrei Gagarin[1], Nathalie Malo[2] and Robert Nadon[2]

[1]Departement d'informatique, Université du Québec à Montreal, C.P.8888, suc.Centre-Ville, Montreal, QC, H3C 3P8, Canada, [2]McGill University and Genome Quebec Innovation Centre, 740 Dr Penfield, Montreal, QC, H3A 1A4, Canada

Received on February 9, 2006; revised on March 20, 2006; accepted on March 28, 2006
Advance Access publication April 4, 2006
Associate Editor: Jonathan Wren

ABSTRACT

Motivation: High-throughput screening (HTS) plays a central role in modern drug discovery, allowing for testing of >100 000 compounds per screen. The aim of our work was to develop and implement methods for minimizing the impact of systematic error in the analysis of HTS data. To the best of our knowledge, two new data correction methods included in HTS-Corrector are not available in any existing commercial software or freeware.

Results: This paper describes HTS-Corrector, a software application for the analysis of HTS data, detection and visualization of systematic error, and corresponding correction of HTS signals. Three new methods for the statistical analysis and correction of raw HTS data are included in HTS-Corrector: background evaluation, well correction and hit-sigma distribution procedures intended to minimize the impact of systematic errors. We discuss the main features of HTS-Corrector and demonstrate the benefits of the algorithms.

Availability: The Microsoft Windows version and a detailed description of the software are freely available at the following URL: http://www.labunix.uqam.ca/~makarenv/hts.html

Contact: makarenkov.vladimir@uqam.ca

1 INTRODUCTION

High-throughput screening (HTS) technology provides rapid screening of large number of compound collections against putative drug targets. A typical HTS operation in the pharmaceutical industry consists of processing >100 000 compounds per screen, generating ~50 million data points per year (Heuer *et al.*, 2002). HTS operates with samples in microliter volumes that are arranged in two-dimensional plates. HTS plates may contain 96 (8 × 12), 384 (16 × 24) or 1536 (32 × 48) samples. Hits can be defined as positive signals corresponding to biologically or chemically active compounds.

Quality control is an essential part of correct hit selection in HTS. Random and systematic error can induce either underestimation (false negatives) or overestimation (false positives) of measured signals. Various methods for quality control, systematic error correction, random error estimation and statistical testing of HTS data have been proposed in the scientific literature (Zhang *et al.*, 2000; Brideau *et al.*, 2003; Gunter *et al.*, 2003; Malo *et al.*, 2006).

*To whom correspondence should be addressed.

We describe a new software (HTS-Corrector) designed for statistical analysis and visualization of HTS data. The methods included in HTS-Corrector are based on the statistical analysis of signal variation within plates of an assay. HTS-Corrector displays the results of the correction analysis in the form of plate maps, tables and bar charts (Fig. 1). Details of the algorithms are provided elsewhere (Kevorkov and Makarenkov, 2005; Gagarin *et al.*, 2006).

2 MAIN FEATURES

HTS-Corrector offers the following options for the statistical analysis and correction of HTS data.

2.1 Evaluation of the background surface

Ideally, across-plate means for inactive samples should be unrelated to the well location. In practice, however, systematic errors generate reproducible local artifacts and smooth global drifts on the background surface, creating biased measurements which depend on well location within the plates. One option in HTS-Corrector is to first normalize all plates using the *Z*-score transformation, which standardizes measurements within plates to have a mean of zero and a standard deviation of one. Mean *Z*-score values are then calculated for each well position across all plates within the screen.

A trend-surface analysis procedure uses a fourth degree polynomial least-squares function to discover general trends and local effects on the mean Z-score values. This global surface-fitting method generates a two-dimensional surface which depicts the main trends of the evaluated background. However, because of their statistical properties, we recommend not to use the high order polynomials for data correction, but only for a better visualization of general trends of the background surfaces (Hastie *et al.*, 2001). The 'Remove background' option enables the user to subtract evaluated background from the original dataset, minimizing the impact of systematic errors (Kevorkov and Makarenkov, 2005).

2.2 Well normalization procedure

The 'Well correction' option first normalizes the experimental values within plates (using 'Z-score standardization') and then analyzes arrays of values in each well along the whole HTS assay. The method proceeds by computing the approximation curve using either a straight line or a second-degree polynomial and subtracts it from plate-normalized experimental values. In

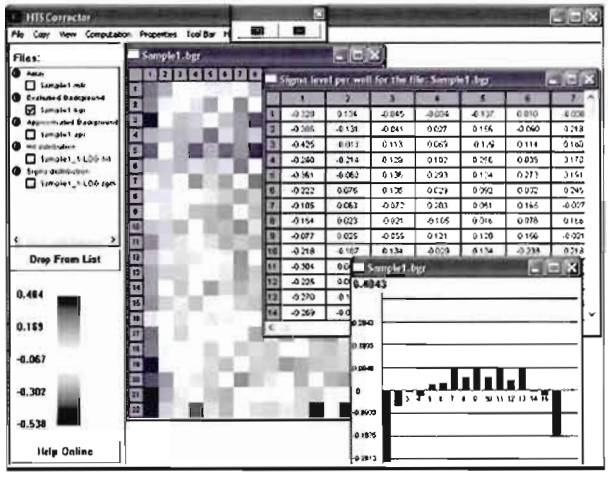


Fig. 1. Screenshot of HTS-Corrector showing a background map, data table and bar chart characterizing an evaluated background.

addition, normalization along each well is carried out (Gagarin *et al.*, 2006).

2.3 Hit-sigma distribution method

The 'Hit-sigma distribution' method allows the user to set well-specific standard deviation thresholds with respect to a user-defined mean hit rate. This approach offers advantages over the more traditional procedure of using a fixed threshold for all wells (e.g. 3 SD), which is not optimal in the presence of location-specific systematic errors.

2.4 *K*-means plate partitioning of large assays

Systematic errors generate repeatable local artifacts and smooth global drifts which become more noticeable when computing a mean assay background. For small HTS assays, the plate background patterns should not substantially vary from plate to plate. However, the plate patterns for large assays may change from batch to batch or shift over a time. HTS-Corrector includes a *k*-means partitioning procedure (MacQueen, 1967) to find the breaks between batches. This procedure allows the user to separate the initial dataset into several subsets with similar plate patterns.

3 BENEFITS OF THE PROPOSED APPROACHES

We use simulated data to illustrate the proposed correction procedures. The dataset contains 1250 'plates' with 'wells' arranged in 8×10 matrices (imitating the parameters of the McMaster assay proposed as a benchmark for the McMaster data mining and docking competition and examined in detail in Gagarin *et al.*, 2006) (http://hts.mcmaster.ca/HTSDataMiningCompetition.htm).

We first examined the output of various hit identification methods on null simulated data [$\sim N(0, 1)$]. Using a 3σ hit detection threshold, 119, 117 and 104 false positives were found in the uncorrected, and in the HTS-Corrector 'background removed' and 'well correction' data, respectively. Second, we randomly added 1% of hits to the null dataset. Hit values were randomly chosen from the range $[\mu - 3.5\sigma; \mu - 4.5\sigma]$, where μ and σ denote the within-plate mean and standard deviation, respectively. Systematic errors affected by random noise were added to this dataset according to a constant c, which ranged from 0σ through 0.5σ (in 0.1σ increments), to produce six 'noisy'datasets. The values $4c$, $3c$, $2c$, $1c$, 0, 0, $-1c$, $-2c$, $-3c$ and $-4c$ were added to the 1st, 2nd, ..., 10th columns, respectively. Finally, normally distributed random error, [$\sim N(0, c/4)$], was added independently to each well of each plate. For each value of c, hits were selected from the uncorrected, background-removed and well-corrected datasets using the 3σ threshold. The true positive rate was noticeably higher for both corrected datasets, while the background subtraction procedure slightly outperformed the well correction method (Fig. 2a). The false positive rates of the corrected data were always lower than that of the uncorrected data, although the well correction method generally outperformed the background subtraction procedure (Fig. 2b).

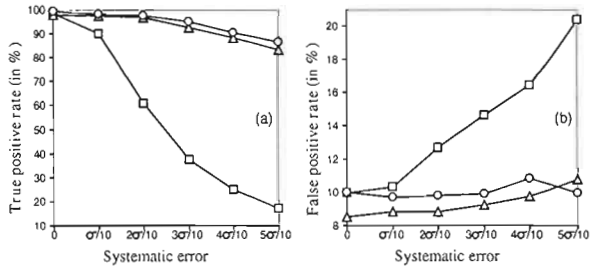


Fig. 2. True (a) and false positive (b) hit detection rates for random data with added systematic error obtained from uncorrected (open square), background subtracted (open circle) and well corrected (open triangle) methods.

ACKNOWLEDGEMENTS

The authors thank Genome Quebec for funding this project. The authors also thank two anonymous referees and Associate Editor Jonathan Wren for their helpful comments.

Conflict of Interest: none declared.

REFERENCES

Brideau,C. *et al.* (2003) Improved statistical methods for hit selection in high-throughput screening. *J. Biomol. Screen.*, **8**, 634–647.

Gagarin,A., Kevorkov,D., Makarenkov,V. and Zentilli,P. (2006) Comparison of two methods for detecting and correcting systematic error in HTS data. In *Proceedings of IFCS 2006*, Ljubljana, Springer-Verlag, (in press).

Gunter,B. *et al.* (2003) Statistical and graphical methods for quality control determination of HTS data. *J. Biomol. Screen.*, **8**, 624–633.

Hastie,T., Tibshirani,R. and Friedman,J. (2001) *The Elements of Statistical Learning*. Springer-Verlag.

Heuer,C., Haenel,T. and Prause,B. (2002) A novel approach for quality control and correction of HTS data based on artificial intelligence. *Pharmaceutical Discovery and Development Report*.

Kevorkov,D. and Makarenkov,V. (2005) Statistical analysis of systematic errors in high-throughput screening. *J. Biomol. Screen.*, **10**, 557–567.

MacQueen,J. (1967) Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press.

Malo,N. *et al.* (2006) Statistical practice in high-throughput screening data analysis. *Nat. Biotechnol.*, **24**, 167–175.

Zhang,J.H. *et al.* (2000) Confirmation of primary active substances from HTS of chemical and biological populations: a statistical approach and practical considerations. *J. Comb. Chem.*, **2**, 258–265.

4.3 « Using Clustering Techniques to Improve Hit Selection in High-Throughput Screening »

Cet article présente trois approches de recherche des « hits » par des méthodes de groupement : le groupement par *k-means*, la *somme des distances inter-cluster moyennes* (SASD : « Sum of the Average Squared inside-cluster Distances ») et la *distance moyenne entre clusters* (AICD : « Average Inter Cluster Distance »).

L'article présente la problématique, notamment les diverses situations pouvant se présenter lors d'un criblage à haut débit, vue dans l'optique du groupement. Il présente les différentes méthodes utilisées et les teste selon deux approches : le criblage étant considéré dans sa totalité ou plateau par plateau. Les tests ont été effectués pour des données simulées présentant une distribution normale et une distribution longue traine (ou « heavy tailed»). Ils ont été ensuite effectués aussi sur les données de la compétition de l'Université McMaster (1250 plateaux d'un test sur l'inhibition de *l'Escherichia coli dihydrofolate reductase*).

L'article conclut que les deux premières méthodes (*k-means* et SASD) présentent de meilleurs résultats par rapport à la troisième méthode (AICD) et à la méthode classique de seuil unique (généralement μ-3σ). Il indique aussi que l'utilisation de ces méthodes en considérant le criblage dans sa totalité, quand possible, mène à de meilleurs résultats.

Gagarin A., Makarenkov V. et Zentilli P. (2006) « Using Clustering Techniques to Improve Hit Selection in High-Throughput Screening. » *Journal of Biomolecular. Screening*, **11**, 903-914.

J Biomol Screen OnlineFirst, published on November 7, 2006 as doi:10.1177/1087057106293590

Using Clustering Techniques to Improve Hit Selection in High-Throughput Screening

ANDREI GAGARIN,[1] VLADIMIR MAKARENKOV,[2] and PABLO ZENTILLI[2]

A typical modern high-throughput screening (HTS) operation consists of testing thousands of chemical compounds to select active ones for future detailed examination. The authors describe 3 clustering techniques that can be used to improve the selection of active compounds (i.e., hits). They are designed to identify quality hits in the observed HTS measurements. The considered clustering techniques were first tested on simulated data and then applied to analyze the assay inhibiting *Escherichia coli* dihydrofolate reductase produced at the HTS laboratory of McMaster University. (*Journal of Biomolecular Screening* 2006:1-12)

Key words: high-throughput screening, hit selection, nonhierarchical clustering, k-means partitioning, inside-cluster distance, intercluster distance

INTRODUCTION

HIGH-THROUGHPUT SCREENING (HTS) is one of the initial stages of the drug discovery process. It allows for testing of hundreds of thousands of chemical compounds per day to select the most prominent candidates for future examination. The compounds are tested against therapeutic targets. A typical HTS procedure generates enormous data volume that requires appropriate processing tools and mechanisms. Recent developments in modern mass screening are highly influenced by the increasing number of targets identified by genomics and by the expansion of the libraries of compounds synthesized using methods of combinatorial chemistry.

Correct selection of active compounds (i.e., hits), is crucial: It is important to know that the expected drug candidate is present in the data and that the selected set of compounds does not lead to unnecessary clinical research. The mass screening process has several drawbacks including the absence of standardized data validation and the lack of reliable quality control. This makes the correct identification of active compounds quite difficult. The incoherencies are partially due to the presence of random and systematic errors in the data. Some statistical methods and software for correcting HTS data have been recently proposed in the literature. The reader is referred to the articles by Heuer et al.,[1] Gunter et al.,[2] Brideau et al.,[3] Heyse,[4] Zhang et al.,[5,6] Kevorkov and Makarenkov,[7] and Makarenkov et al.[8,9]

Another factor that influences the identification of active compounds is the hit selection procedure itself: Once the data are preprocessed and checked for quality, one has to decide which compounds should be tested in a secondary screen. However, from the statistical point of view, it is not well defined or reasonably grounded how to select the active compounds. Current practices usually apply informal rules that are based on particular laboratory constraints such as capacity limitations or financial costs of the follow-up procedures.[10]

One can identify hits by plotting raw or preprocessed measured values against compound label. First, plot the compound identity on the x axis and its activity measurement on the y axis for each plate separately and then identify compounds whose measured activity deviates from the majority of the measurements. This approach is well suited for identifying compounds with a high activity level. However, compounds of low or intermediate activity levels may be missed by such an "eyeball" procedure.

One can also select hits by computing a fixed percentage of the compounds with the highest measured activity (e.g., select 1% or 2% of the most active compounds on each plate or in the bulk of the data). This method is not statistically justified and can lead to some undesirable artifacts: The real number of hits per plate may vary a lot, and it is usually not reliable to compare measurements on different plates. Because the true number of active compounds is not known in advance, one cannot justify the selection of a fixed percentage of the primary screen compounds.

[1]Laboratoire LaCIM, [2]Département d'Informatique, Université du Québec à Montréal, C.P. 8888, succursale Centre-Ville, Montréal (Québec), Canada, H3C 3P8.

Received Apr 21, 2006, and in revised form Jul 3, 2006. Accepted for publication Jul 11, 2006.

Journal of Biomolecular Screening 11(X); 2006
DOI:10.1177/1087057106293590

Another current practice to select hits in a particular plate consists of calculating the plate mean value μ and its standard deviation σ to identify samples differing from the mean μ by at least $c\sigma$, where c is a preliminary chosen constant. For example, in the case of an inhibition assay and $c = 3$, we would select samples with measured values smaller than $\mu - 3\sigma$. This is a classical hit selection approach applied on a plate-by-plate basis. The main drawback of this hit selection procedure is the assumption that all hits have measured values smaller than a preestablished threshold depending only on the plate mean and standard deviation. To avoid this limitation, we propose using the strategy based on the following assumption: The measured values of active samples are significantly different from those of inactive ones. Such an assumption leads to a new hit selection approach that consists of finding statistically justified clusters of samples having some of the smallest or biggest measured values on each plate or in the single batch of the assay data; their values should be well distinguished from the values of all other samples. Because only a small percentage of compounds are active, the size of their cluster should be reasonably small.

Two clustering procedures will be discussed in the article. The 1st procedure considers each plate as an independent experiment, whereas the 2nd one treats all assay compounds as a single batch. The performances of the hit selection methods based on the cluster analysis with respect to the classical hit selection procedure will be illustrated first on the random data having standard normal and long-tails distributions. We will also show the differences between the 2 approaches while examining the assay inhibiting *Escherichia coli* dihydrofolate reductase generated at the HTS laboratory of McMaster University.[11]

MATERIALS AND METHODS

Random data generation and type I error

A typical HTS procedure consists of running samples arranged in 2-dimensional plates of the same format through automated screening machines that make experimental measurements. Samples are placed in wells. Most of the samples are inactive, and the measurements corresponding to active samples are assumed to be substantially different from those of inactive compounds.

We use random data sets following 2 different distributions to prove the efficiency of the cluster-based hit selection approach. The experiments were carried out on random data having the standard normal and long-tails distributions. Each random data set consists of a 1250-plate assay, with each plate having wells arranged in 8 rows and 10 columns, thus imitating the parameters of the McMaster *E. coli* screen.[11]

First, 2 random data sets with no hits were generated according to the standard normal (~N(0, 1)) and long-tails distributions. The classical hit selection procedure and 3 hit selection methods based on the cluster analysis were applied to these data. The 3 clustering strategies used to search for hits were the following: k-means partitioning, sum of the average squared inside-cluster distances (SASD), and average intercluster distance (AICD). The reader is referred to Arabie et al.[12] for an overview of clustering methods. The 3 clustering strategies considered in this study are described in more detail later in this section. Because the initial random data are not supposed to contain hits at all, the detected hits should be considered false positives. **Table 1** reports the number of false-positive hits found in the random data with no hits. Here, each plate was considered an independent experiment. In the case of classical hit selection, the sigma thresholds of $\mu - 3\sigma$ (standard normal data) and $\mu - 3.37\sigma$ (long-tails data) were applied. The sigma threshold for the long-tails data was chosen to have approximately the same number of hits that were found in the standard normal data (~120 hits in the no-hits data sets). Note that the cluster-based hit selection generally caused a small increase in the number of false positives (3.2% on average) compared to the classical hit selection.

Table 1. Type I Error: False-Positive Hits Found Using the Classical Hit Selection and the 3 Considered Clustering Methods in the Raw Random Data (With No Hits) Having Standard Normal and Long-Tails Distributions

Statistic\Distribution	*Standard Normal*	*Long-Tails*
Sigma threshold for hit selection	$\mu - 3\sigma$	$\mu - 3.37\sigma$
Number of false positives		
Classical selection	119	120
K-means partitioning	125	129
Sum of the average squared inside-cluster distances	122	125
Average intercluster distance	119	120
Interval for hit generation	$[\mu - 3.4\sigma;$ $\mu - 4.4\sigma]$	$[\mu - 4.07\sigma;$ $\mu - 5.07\sigma]$

Computations were carried out on a plate-by-plate basis.

Then, we added 1% to 5% of hits into 5 replicates of the random no-hits data. The hit locations were chosen randomly: The probability of each well to contain a hit was, respectively, 1%, 2%, 3%, 4%, and 5%. The hit values were randomly selected to be in the interval $[\mu - 3.4\sigma; \mu - 4.4\sigma]$ for the standard normal data and in the interval $[\mu - 4.07\sigma; \mu - 5.07\sigma]$ for the long-tails data, where μ denotes the mean value and σ denotes the standard deviation of the observed plate. Having data with randomly generated hits, we applied the classical hit selection procedure and the 3 considered clustering methods to detect hits in each simulated data set. The analyses were conducted for the clustering procedure working on the plate-by-plate basis and that treating all the assay data as a single batch.

Clustering methods and hit selection

One of the advantages of the clustering techniques is the possibility to find hits having values bigger than a fixed threshold (in

the case of an inhibition assay). Such hits are completely ignored by the classical hit selection procedure.

In this study, we consider 3 nonhierarchical clustering methods. First, we assume that the number of clusters k is known. The objective is to partition n-given elements into the required k nonempty clusters. Clustering techniques allow objects to change their group membership through the cluster formation process. A clustering method usually starts from an initial partition chosen according to a certain criterion. Then, the reallocation of elements takes place according to an optimality criterion. Here, we consider the 3 following optimality criteria:

1. K-means partitioning[13] that minimizes the total inside-cluster variance:

$$K\text{-}means(X_1, \ldots, X_k) = \sum_{i=1}^{k} \frac{1}{N_i} \sum_{x_j \in X_i} d^2(x_j, \mu_i), \quad \sum_{i=1}^{k} N_i = n, \tag{1}$$

where X_i is the i's cluster containing $N_i = |X_i|$ elements ($N_i > 0$), x_j is an element of X_i, μ_i is the mean point of the cluster X_i, and $d(x_j, \mu_i)$ is the distance between the element x_j and the mean point μ_i of X_i.

2. Sum of the average squared inside-cluster distances between all pairs of elements (both elements of the pair must belong to the same cluster) taken over all clusters. More precisely, the method minimizes the following function:

$$SASD(X_1, \ldots, X_k) = \sum_{i=1}^{k} \frac{2}{N_i(N_i - 1)} \sum_{(x_j, x_m) \in X_i} d^2(x_j, x_m), \quad \sum_{i=1}^{k} N_i = n, \tag{2}$$

where X_i is the i's cluster containing $N_i = |X_i|$ elements ($N_i > 0$), x_j and x_m are 2 elements from the cluster X_i, and $d(x_j, x_m)$ is the distance between x_j and x_m.

3. Average intercluster distance between all pairs of elements belonging to different clusters (the members of each pair must be from different clusters). This partitioning method is adapted from the average linkage method of hierarchical clustering. The method consists in maximizing the following function:

$$AICD(X_1, X_2) = \frac{1}{N_1 N_2} \sum_{x_j \in X_1} \sum_{x_m \in X_2} d(x_j, x_m), \quad N_1 + N_2 = n, \tag{3}$$

where X_1 and X_2 are 2 clusters containing $N_1 = |X_1|$ and $N_2 = |X_2|$ elements, respectively, ($N_1 > 0$ and $N_2 > 0$), x_j is an element of X_1 and x_m is an element of X_2, and $d(x_j, x_m)$ is the distance between x_j and x_m.

A detailed description of the nonhierarchical partitioning methods can be found in the books of Arabie et al.[12] and Legendre and Legendre.[14]

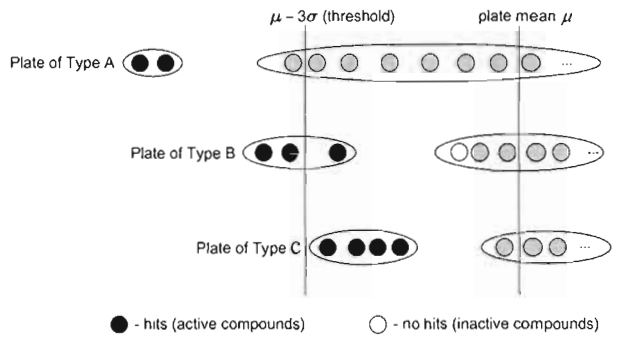


FIG. 1. Hit selection on the plate-by-plate basis. Three types of plates with respect to the classical hit selection threshold set to $\mu - 3\sigma$. Hits are denoted by black points, and nonactive compounds are denoted by gray points.

To be able to use a clustering method in the hit selection process, the following assumptions about HTS data in hand should be made: It is possible to divide clearly the screened samples into active and inactive, the majority of the screened samples are inactive, and measured values of the active samples differ substantially from the inactive ones. We will first show how the clustering methods can be applied on the plate-by-plate basis and then how they can be used when all the assay data are processed as a single batch.

Hit selection on the plate-by-plate basis

Following the above-mentioned assumptions, the measured values are divided into 2 groups. The bigger group contains inactive samples, and the smaller group contains active samples (i.e., hits). We assume that there is a relatively big gap that separates active samples from inactive ones. Also, we assume that it is possible to estimate the size of the gap separating the 2 groups.

We will distinguish 3 types of plates (A, B, and C) with respect to a preestablished hit selection threshold. They are depicted in **Figure 1**. All hits in the plates of type A can be found using the classical hit selection procedure, hits in the plates of type B are partially identified using the classical hit selection, and hits in the plates of type C are ignored by the classical approach. Moreover, the classical approach can also select some false positive elements. The latter elements are located outside the hit clusters in the plates of type A and close to the preestablished threshold.

An appropriate hit selection method should be able to avoid the pitfalls of the data distribution. Usually, it requires data preprocessing to ensure the accuracy of the hit selection. Ideally, active samples have similar values, form a minority group, and can be clearly distinguished from inactive ones. In the case of real data, measured values for active and inactive samples may overlap and not form 2 well-separated clusters. This can happen

naturally or can be due to random and systematic errors that are usually present in HTS screens. However, despite the noise present in the measured values, it should be possible to distinguish 1 or 2 clusters of samples that have some of the smallest (or biggest) values on a particular plate.

In general, to identify clusters of hits, the following steps should be carried out for each plate independently:

- computation of the plate mean value μ and standard deviation σ (hit and outlier elimination can be also carried out at this step),
- sorting measured plate values by increasing order,
- computation of the size of the hit cluster(s) according to a chosen clustering criterion, and
- selection of hits in the obtained clusters.

Cluster construction on the plate-by-plate basis

We assume that in the absence of active compounds on a plate, there should be no significant gaps (according to a selected clustering criterion) separating the measurements. Consequently, in this case, it should be impossible to identify any kind of hit clusters in such a plate. Otherwise, hits will form a cluster (possibly divided into 2 or 3 smaller subclusters) that corresponds to some of the smallest (inhibition assay) measured values of the plate.

Let N be the number of measurements on a plate (in our simulations, N was equal to 80), and assume that the measurements are sorted by increasing order. Each of the plates can fall into 1 of the 2 possible categories:

1. some of the plate values are smaller than a fixed threshold, for example, $\mu - 3\sigma$ (plates of types A or B) or
2. all plate values are equal to or bigger than a fixed threshold (plates of type C).

Depending on the plate category, we use 2 different strategies to find the hit clusters.

Plates of types A and B. When the smallest measured value of the plate is smaller than a preestablished threshold, the 1st local minimum of the clustering function will indicate the size of a preliminary hit cluster. Because we need to partition the data into 2 groups only (i.e., active and inactive samples), the 3 clustering functions described before will be of the following form:

1. The k-means partitioning function:

$$K\text{-}means(N_1) = \frac{1}{N_1}\sum_{i=1}^{N_1}(x_i - \mu_1)^2 + \frac{1}{N - N_1}\sum_{i=N_1+1}^{N}(x_i - \mu_2)^2, \quad (4)$$

where

$$\mu_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} x_i, \quad \text{and} \quad \mu_2 = \frac{1}{N - N_1}\sum_{i=N_1+1}^{N} x_i.$$

2. The sum of the average inside-cluster distances:

$$SASD(N_1) = \frac{2}{N_1(N_1 - 1)}\sum_{i=1}^{N_1 - 1}\sum_{j=i+1}^{N_1}(x_i - x_j)^2 + \frac{2}{(N - N_1)(N - N_1 - 1)}\sum_{i=N_1+1}^{N-1}\sum_{j=i+1}^{N}(x_i - x_j)^2. \quad (5)$$

3. The average intercluster distance:

$$AICD(N_1) = \frac{1}{N_1(N - N_1)}\sum_{i=1}^{N_1}\sum_{j=N_1+1}^{N}|x_i - x_j|, \quad (6)$$

where N is the total number of samples per plate, N_1 is the number of samples in the 1st cluster (i.e., hits), $N - N_1$ is the number of samples in the 2nd cluster (i.e., no hits), and x_i's are the plate-measured values sorted by increasing order.

Assume that there are at most 20% of active compounds on each plate. Then the algorithm searching for hit clusters consequently places 1, 2, 3, . . . , and $N/5$ elements ($N/5 = 16$ in our simulations) into the hit cluster and verifies whether the criterion k-means(N_1), SASD(N_1), or AICD(N_1) is increasing or decreasing. The complement to the hit cluster contains, respectively, $N - 1$, $N - 2$, $N - 3$, . . . , and $4N/5$ elements. The 1st local minimum of k-means(N_1) or SASD(N_1), or the 1st local maximum of AICD(N_1), indicates that there is a well-distinguishable gap between 2 clusters under consideration. If the 1st cluster found by 1 of the clustering methods is too big (e.g., contains more than 20% of the total number of samples of the plate), then we assume that this plate contains no hit cluster at all.

Thus, the coefficients k-means(N_1), SASD(N_1), and AICD(N_1) can be useful in finding the hit clusters containing elements whose measured values are smaller than the fixed threshold. However, if the 1st value outside the preliminary hit cluster is smaller than the fixed threshold, some hits close to the preliminary hit cluster may be ignored. In this case, we assume that the whole hit cluster is composed of 2 or more subclusters. Therefore, in this case, our program searches for the 2nd hit cluster that is indicated by the 1st local minimum of the same clustering coefficient calculated for the plate samples from which the 1st hit cluster has already been removed. The 2nd hit cluster, if found and distinguished from the remaining data, is added to the 1st hit cluster to form the final hit cluster. In some situations, the AICD function may be used to search for the 3rd hit cluster to be added to the 1st 2 to provide better selection results.

Plates of type C. Now consider plates whose values are all bigger than a preestablished threshold (**Fig. 1**). Here, the coefficients k-means(N_1), SASD(N_1), and AICD(N_1) may not be working properly: They may include too many false-positive elements into the hit clusters. Plates with a big number of hits often have all their values bigger than the preestablished threshold. This can be explained by the presence of several

small values that affect the plate's mean and standard deviation. Usually, in this case, there exists a relatively big gap between the hit cluster and the no-hit values. Therefore, we assume that it is still possible to find the hit cluster by estimating this sigma-depending gap for the screen in hand.

If the sigma-gap value is underestimated, it would lead to the selection of small clusters with too many false positives and not enough true hits. An overestimation of the gap value would not allow finding the hit clusters at all. Thus, the cluster search can be very sensitive to the sigma-gap value. During the simulations, we calculated experimentally the optimal values of the sigma gaps, assuming that they depended on the average value of the maximum sigma gaps on all plates of type C. The maximum sigma-gap values between 2 consecutive elements were calculated on the 1st 20% of the plate elements. The computations were done separately for the screens with 0%, 1%, 2%, 3%, 4%, and 5% of generated hits.

Hit selection algorithm based on the cluster analysis using the plate-by-plate approach

The type of the data distribution should be determined in advance. For example, to verify the assumptions of normally distributed data, one can carry out the Kolmogorov-Smirnov test. Given a known statistical distribution, one can use the following clustering algorithm to identify hits:

- Normalize data using the zero-mean centering and unit variance standardization (also known as z-score method).
- Sort the plate measurements by increasing order.
- Compute the maximum sigma-gap value between 2 consecutive elements of the smallest 20% of the plate values for the plates of type C (**Fig. 1**), compute the average of the maximum sigma gaps for all plates of type C, and estimate the optimal sigma-gap values for the data sets with different hit percentages (see **Tables 2** and **3**).
- Compute the size of the hit cluster: For the plates of types A and B, find the 1st local minimum of the clustering coefficient k-means(N_1), SASD(N_1), or AICD(N_1) (if necessary, repeat the computation twice or thrice, removing previously found clusters); for the plates of type C, search for the cluster defined by the optimal sigma-gap value. If the cluster includes elements outside the smallest 20% of the plate measurements, disregard it (it is not a hit cluster). Also, for the plates of type C, if the cluster size is too small (e.g., less than 4% of the plate elements), disregard the cluster (it is not a hit cluster).
- Identify elements in the clusters as hits.

The optimal values of the sigma gaps calculated experimentally in the simulations based on a dichotomy search are presented in **Tables 2** and **3**. Note that in the case of no-hit data (0% of generated hits), the optimal sigma gaps are the smallest values (1.12 for the standard normal data and 1.49 for the long-tails data) that do not lead to the identification of any hit on the plates of type C. The optimal sigma-gap values are assumed to depend on the average of the maximum sigma-gap values of all plates of type C. The best-fit cubic polynomials approximating the values in **Tables 2** and **3** (see formulas 7 and 8) were calculated using the least-squares method from Maple X.[15] The main trend that may be observed in **Figure 2** is as follows: the larger the average of the maximum sigma gaps calculated on the smallest 20% of the plates' measurements, the smaller is the value of the corresponding optimal sigma-gap constant that will be used for the cluster identification.

Table 2. Correspondence between the Average of the Maximum Sigma Gaps Measured on the Smallest 20% of the Plates' Elements and the Optimal Sigma-Gap Values Used to Find Hit Clusters on the Plates of Type C for the Standard Normal Data (Calculated Experimentally)

	Generated Hits (%)					
	0	*1*	*2*	*3*	*4*	*5*
Average max sigma gap	0.447	0.453	0.53	0.71	0.82	0.85
Optimal sigma gap to be used to find clusters	1.12	0.93	0.62	0.49	0.43	0.41

Table 3. Correspondence between the Average of the Maximum Sigma Gaps Measured on the Smallest 20% of the Plates' Elements and the Optimal Sigma-Gap Values Used to Find Hit Clusters on the Plates of Type C for the Long-Tails Data (Calculated Experimentally)

	Generated Hits (%)					
	0	*1*	*2*	*3*	*4*	*5*
Average max sigma gap	0.632	0.645	0.78	0.98	1.055	1.063
Optimal sigma gap to be used to find clusters	1.49	1.04	0.76	0.56	0.5	0.48

The cubic polynomial approximating the data in **Table 2** (standard normal distribution) is as follows:

$$P(x) = 14.37 - 60.29x + 86.68x^2 - 41.28x^3. \quad (7)$$

The graph showing the correspondence of this approximation to the experimentally calculated values is represented in **Figure 2a**.

The cubic polynomial approximating the data in **Table 3** (long-tails distribution) is as follows:

$$P(x) = 22.55 - 71.54x + 78.06x^2 - 28.5x^3. \quad (8)$$

The graph showing the correspondence of this approximation to the experimentally calculated values is represented in **Figure 2b**.

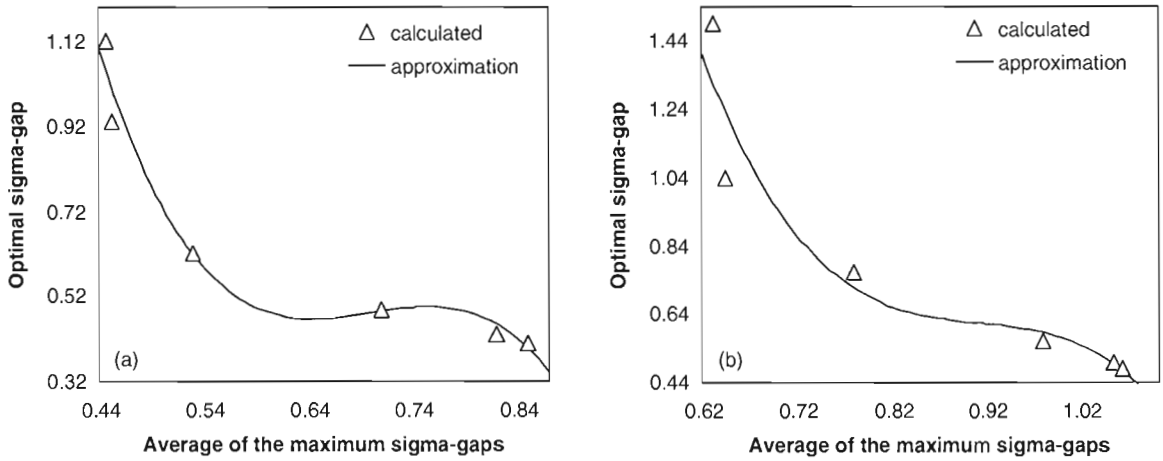


FIG. 2. Hit selection on the plate-by-plate basis. Approximation of the experimentally calculated optimal sigma-gap values by the best-fit cubic polynomial for (a) the standard normal and (b) the long-tails data.

Hit selection from a single batch

In the previous section, we considered the hit selection process that treats each plate as an independent experiment. Here, we focus on the analysis considering the whole assay data as a single batch. To conduct this kind of experiment, we first have to make sure that all plates of our assay were processed under the same testing conditions. This analysis can be recommended when a well-optimized assay protocol shows little plate-to-plate variability.

Thus, we conducted the simulations treating altogether the data of the whole assay. Two algorithmic strategies are possible in this case. First, we can still process the data on the plate-by-plate basis but use the parameters (here, the mean value and standard deviation) computed for the data from the whole assay instead of those computed for each particular plate. Such a strategy will not be sensitive at all to the data distribution (i.e., the data can be distributed randomly or not). Thus, the compounds that are not randomly distributed can be processed in this way. No important changes in the above-presented algorithm should be done to implement this strategy.

The 2nd strategy, the strategy that we actually tested, assumes that all the assay data are coming from a large single plate. Thus, the 3 considered clustering procedures can be tested in turn to find the best, according to the selected criterion, partition of the entire data set into the clusters of active compounds (i.e., hits) and inactive compounds. We assume that this separation should occur not far from the traditional hit selection threshold $\mu - 3\sigma$, where μ is the mean value of the whole assay and σ is the assay standard deviation. This strategy proceeds by testing a fixed number of cluster partitions and selects the partition that optimizes the value of the given clustering coefficient (see **Fig. 3**). Up to 500 cluster partitions (i.e., the search interval in **Fig. 3**) were tested in our simulation study for each considered random data set.

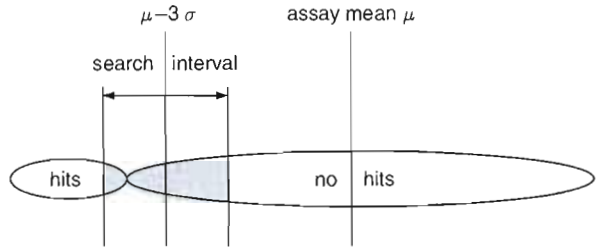


FIG. 3. Hit selection from a single batch. Classical hit selection procedure selects as hits the compounds whose values are lower than $\mu - 3\sigma$. The clustering procedures look for the best partitioning of the given assay into the clusters of hits and no hits. The gray area indicates the search interval for partitioning.

RESULTS AND DISCUSSION

Experimental results for the hit selection carried out on the plate-by-plate basis

In this study, we examined 2 random HTS assays. Both simulated assays consisted of 1250 plates having wells arranged in 8 rows and 10 columns. The measurements of the 1st assay followed a standard normal distribution (~N(0, 1)), and the measurements of the 2nd assay followed a long-tails distribution. First, we conducted the analysis on the plate-by-plate basis. Numerical results for the classical hit selection and the 3 clustering methods considered in this article are presented in **Tables 4** and **5**. The statistics reported in **Tables 4** and **5** were obtained by running the simulation program 100

Table 4. Hit Selection on the Plate-by-Plate Basis: Average Hit Selection Results for the Standard Normal Data Obtained Using the Classical Hit Selection Procedure and the 3 Clustering Methods

	Generated Hits (%)				
	1	*2*	*3*	*4*	*5*
Generated hits (number)	998.5	1997.1	2997.6	3992.5	4994.6
Classical hit selection					
Sigma threshold	3.0σ	2.88σ	2.82σ	2.76σ	2.67σ
Total hits found	1069.1	1930.4	2521.3	2872.4	3180.7
False positives	106.5	125.6	126.3	115.4	107.3
False negatives	37.1	192.2	602.5	1235.5	1919.6
Correct hit rate (%)	96.3	90.4	79.9	69.1	61.6
K-means partitioning					
Total hits found	1119.2	2096.1	3108.2	4079.2	5041.2
False positives	127.9	136.4	135.7	125.8	116.4
False negatives	12.3	26.2	27.6	34.8	57.9
Correct hit rate (%)	98.8	98.7	99.1	99.1	98.8
Sum of the average squared inside-cluster distances					
Total hits found	1098.4	2083.8	3061.9	4046.1	4978.8
False positives	117.9	123.3	123.7	113.9	106.9
False negatives	13.8	38.4	54	72.9	104.8
Correct hit rate (%)	98.6	98.1	98.2	98.2	97.9
Average intercluster distance					
Total hits found	1077.8	1987.4	2845.9	3708.3	4629
False positives	106.9	102.5	96.2	84.2	78.7
False negatives	27.2	118.7	247.3	366.7	446
Correct hit rate (%)	97.3	94.1	91.8	90.8	91.1

Table 5. Hit Selection on the Plate-by-Plate Basis: Average Hit Selection Results for the Long-Tails Data Obtained Using the Classical Hit Selection Procedure and the 3 Clustering Methods

	Generated Hits (%)				
	1	*2*	*3*	*4*	*5*
Generated hits (number)	999.8	2005.3	2997.9	3996.5	4994.7
Classical hit selection					
Sigma threshold	3.37σ	3.27σ	3.19σ	3.16σ	3.09σ
Total hits found	1047.9	1764.8	2134.8	2132.1	2090.3
False positives	103.5	123.2	118.3	89.2	73.5
False negatives	56.3	361.8	984.7	1953.2	2980.2
Correct hit rate (%)	94.4	82	67.2	51.2	40.4
K-means partitioning					
Total hits found	1105.1	2099.7	3108.7	4079.1	5056.1
False positives	125.2	129.5	118.7	93.9	77.1
False negatives	18.2	31.8	7.5	9.2	15.5
Correct hit rate (%)	98.2	98.4	99.8	99.8	99.7
Sum of the average squared inside-cluster distances					
Total hits found	1099.3	2080.4	3082.2	4061.3	5016.1
False positives	117.9	122.3	114.5	86.8	73.1
False negatives	20.5	35.6	16	20.4	28.8
Correct hit rate (%)	98	98.2	99.5	99.5	99.4
Average intercluster distance					
Total hits found	1084.6	2041.5	2999.1	3931.7	4903.9
False positives	112.8	113.8	105.2	78.4	64.6
False negatives	27.7	82	106.6	145.4	161.4
Correct hit rate (%)	97.2	95.9	96.5	96.4	96.8

times for each random data set and calculating the average values of the runs.

In the case of 1% hit data, the hits were classically selected by choosing values lower than the thresholds $\mu - 3\sigma$ and $\mu - 3.37\sigma$ for the standard normal and long-tails data, respectively. In the case of other hit percentages, the classical sigma threshold was adjusted to keep the number of false positives close to that found by the clustering methods. Note that a lower threshold increases the number of false negatives. This is due to a trade-off between the number of true hits and the number of false negative hits (see **Fig. 4a**).

Thus, we computed the true hit detection rate, the false-positive (i.e., inactive samples that were identified as hits) rate, and the false-negative (i.e., real hits that were not detected) rate during the simulations. The true hit detection rate was much higher for all 3 clustering methods compared to the classical hit selection procedure (see **Fig. 5**). For both data distributions, k-means partitioning and SASD clustering outperformed AICD clustering and the classical hit selection procedure.

The classical hit selection implies a trade-off between the true hit rate and the false-positive rate. The influence of a fixed threshold on the false-positive and false-negative rates is illustrated in **Figure 4**. This figure shows a schematic distribution and an overlap between hit and no-hit measurements with respect to a fixed classical threshold (**Fig. 4a**). **Figure 4b** shows the distribution and the overlap between the false-positive and false-negative values in the threshold area for the generated standard normal random data with 5% of added hits. An increase in the number of correctly found hits obtained by adjusting the classical threshold usually implies a sensitive increase in the number of false positives. The cluster approach can attenuate this artifact: Hit clusters are not sensitive to any preestablished threshold and can grab more correct hits without an important increase in the false positive and false negative rates. For both standard normal and long-tails data including 1% to 3% of generated hits (see **Tables 4** and **5**), the number of false positives remained at the same level as the number of false positives in the raw data with no hits (in our case, approximately 120 false positives).

Note that the k-means partitioning method detected slightly more true hits than the SASD method did (see **Fig. 5**). However, the SASD method performed slightly better than k-means partitioning in terms of false positives (see **Tables 4** and **5**). The AICD method was certainly the best in terms of

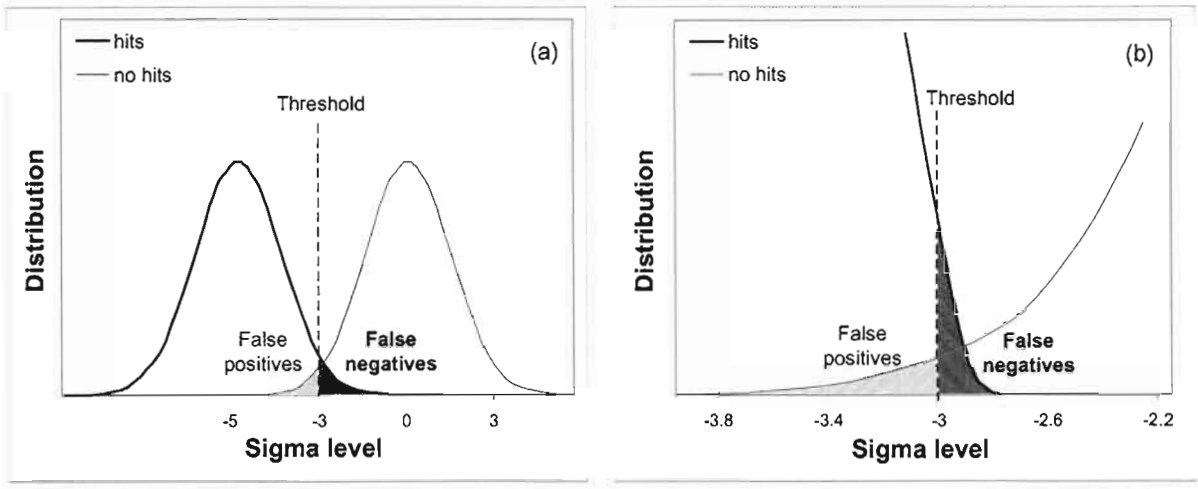


FIG. 4. Influence of a fixed threshold on the false positive (gray area) and false negative (black area) rates in the case of standard normal data. (a) Typical distribution of hits and no hits. (b) Distribution of hits and no hits in the threshold area for the generated standard normal random data with 5% of added hits.

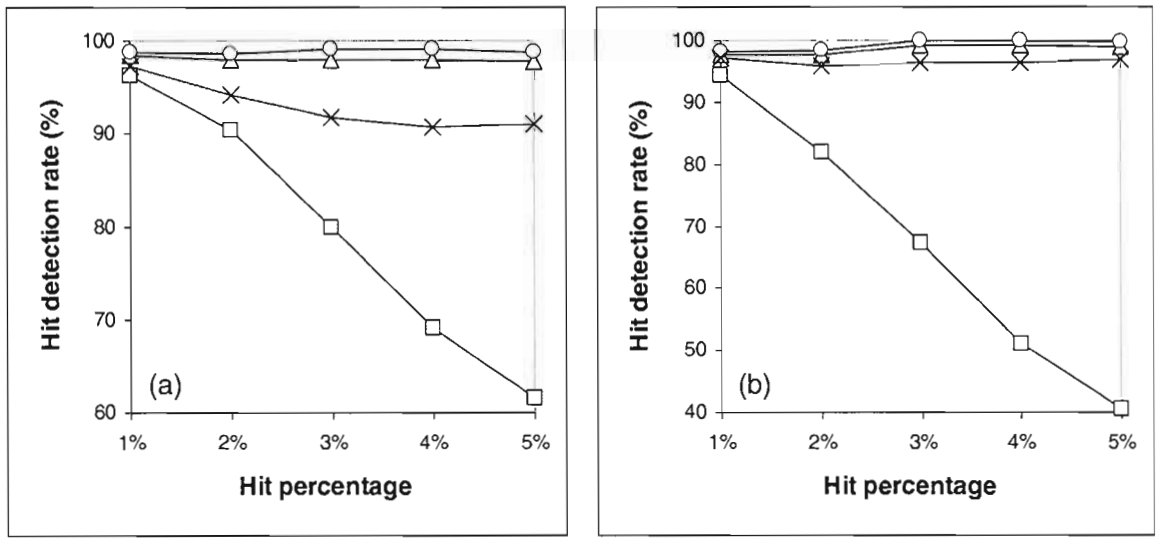


FIG. 5. Hit selection on the plate-by-plate basis. Variation of the true hit detection rate depending on the hit percentage. (a) Standard normal data. (b) Long-tails data. Hit selection methods: □ = classical; ○ = k-means partitioning; △ = sum of the average squared inside-cluster distances clustering; × = average intercluster distance clustering.

false positives but very inconsistent in terms of false negatives (see **Tables 4** and **5**).

Experimental results for the hit selection carried out for a single batch of data

We also carried out simulations to test the procedure processing all the assay data at the same time. Similarly to the plate-by-plate approach, we conducted our experiments on the two 1250-plate assays having standard normal (~N(0, 1)) and long-tails distributions of data. The obtained results for the 4 competing strategies, including the traditional hit selection method, are given in **Tables 6** and **7**. The reported statistics were obtained as average results obtained after 100 runs. Similar to the simulation described in the previous paragraph, the hit percentage varied from 1% to 5%.

The hits were randomly generated with respect to the above-described procedure. Note that the sigma thresholds reported in **Tables 6** and **7** were different from those reported in **Tables 4** and **5**. Here, they were adjusted in a way that the classical hit

Table 6. Hit Selection from a Single Batch: Average Hit Selection Results for the Standard Normal Data Obtained Using the Classical Hit Selection Procedure and the 3 Clustering Methods

	Generated Hits (%)				
	1	*2*	*3*	*4*	*5*
Generated hits (number)	1004.5	2002.9	2994.4	3992.2	5004.9
Classical hit selection					
Sigma threshold	3.0σ	2.58σ	2.30σ	2.10σ	1.94σ
Total hits found	1043.0	1998.1	2957.0	3941.5	5007.7
False positives	60.9	203.9	399.7	643.0	965.0
False negatives	22.5	208.7	437.1	693.8	962.2
Correct hit rate (%)	97.8	89.6	85.4	82.6	80.8
K-means partitioning					
Total hits found	1038.0	2097.1	3156.0	4140.5	5306.7
False positives	60.3	234.9	447.9	710.7	1078.6
False negatives	26.8	140.8	286.3	562.5	776.8
Correct hit rate (%)	97.3	92.3	90.4	85.9	84.5
Sum of the average squared inside-cluster distances					
Total hits found	1091.1	2097.1	3156.0	4140.5	5003.7
False positives	86.5	234.9	447.9	710.7	963.7
False negatives	0.0	140.8	286.3	562.5	964.9
Correct hit rate (%)	100.0	93.0	90.4	85.9	80.7
Average intercluster distance					
Total hits found	1039.0	1994.1	2953.0	3937.5	5003.7
False positives	64.5	202.5	398.9	641.7	963.7
False negatives	26.0	211.3	440.3	696.4	964.9
Correct hit rate (%)	97.0	89.5	85.3	82.6	80.7

Table 7. Hit Selection from a Single Batch: Average Hit Selection Results for the Long-Tails Data Obtained Using the Classical Hit Selection Procedure and the 3 Clustering Methods

	Generated Hits (%)				
	1	*2*	*3*	*4*	*5*
Generated hits (number)	1001.6	2004.2	3001.4	3991.6	4999.1
Classical hit selection					
Sigma threshold	3.12σ	2.70σ	2.38σ	2.10σ	1.95σ
Total hits found	985.6	1940.6	2896.5	4069.5	4912.9
False positives	203.1	540.6	883.7	1378.6	1636.7
False negatives	219.1	604.2	988.5	1300.7	1722.9
Correct hit rate (%)	78.1	69.9	67.1	67.4	65.5
K-means partitioning					
Total hits found	1084.6	2039.6	3095.5	4368.5	5511.9
False positives	251.1	570.6	960.8	1470.8	1845.9
False negatives	168.0	535.1	866.6	1093.9	1333.1
Correct hit rate (%)	83.2	73.3	71.1	72.6	73.3
Sum of the average squared inside-cluster distances					
Total hits found	1084.6	2039.6	3095.5	4368.5	4908.9
False positives	251.1	570.6	960.8	1470.8	1635.2
False negatives	168.0	535.1	866.6	1093.9	1725.4
Correct hit rate (%)	83.2	73.3	71.1	72.6	65.5
Average intercluster distance					
Total hits found	976.6	1936.6	2892.5	4065.5	4908.9
False positives	198.9	539.4	882.2	1377.3	1635.2
False negatives	223.9	607.0	991.0	1303.4	1725.4
Correct hit rate (%)	77.6	69.7	67.0	67.3	65.5

selection procedure selects the number of hits close to the real number of generated hits. Because of this difference, the hit detection rate for the classical hit selection procedure reported in **Tables 4** and **6** and **Tables 5** and **7**, and illustrated in **Figures 5** and **6**, respectively, cannot be actually compared between them. The 3 clustering algorithms were then carried out within the search interval (see **Fig. 3**) in the area of the classical hit selection threshold. The search intervals with 10, 100, 250, and 500 elements were tested in our study, and the best results were reported. In general (see **Fig. 6**), the k-means partitioning and SASD clustering procedures outperformed AICD clustering and the classical hit selection method.

One can notice that the results of the clustering methods obtained using the plate-by-plate approach (**Tables 4** and **5**) are better, in almost all cases, than those obtained using the single-batch approach (**Tables 6** and **7**). This can be explained by the fact that the latter approach is dependable on the classical hit selection threshold and does not offer the possibility of studying independently the plate distributions of measured values as the plate-by-plate approach does.

Searching for hits in the experimental data

We applied the classical hit selection procedure and the 3 considered clustering methods to the experimental data set generated at the McMaster University HTS laboratory and compared the obtained results. This HTS assay is publicly available at the following Web site: http://hts.mcmaster.ca/HTSDataMiningCompetition.htm (see also the work of Zolli-Juran et al[11]). It consists of a screen of compounds inhibiting *E. coli* dihydrofolate reductase. Each compound was screened twice: 2 copies of 625 plates were run through the screening machines. This gives 1250 plates in total, each having wells arranged in 8 rows and 12 columns (columns 1 and 12 containing controls were not considered in this study). The assay conditions reported in Zolli-Juran et al[11] were the following: Assays were carried out at 25° C and performed in duplicate. Each 200-μL reaction mixture contained 40 μM NADPH, 30 μM DHF, 5 nM DHFR, 50 mM Tris (pH 7.5), 0.01% (w/v) Triton, and 10 mM β-mercaptoethanol. Test compounds from the screening library were added to the reaction before initiation by enzyme and at a final concentration

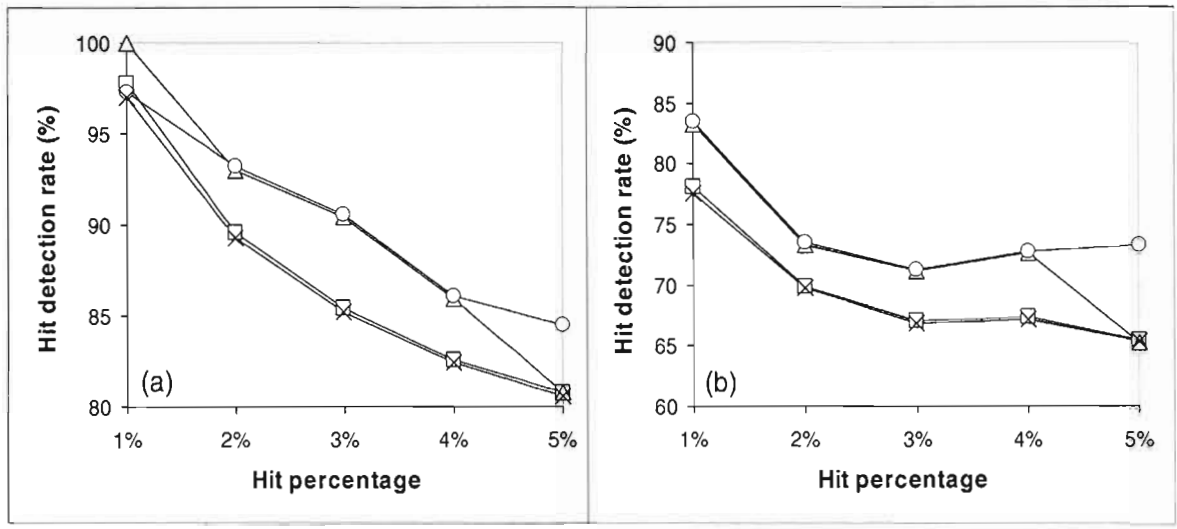


FIG. 6. Hit selection from a single batch. Variation of the true hit detection rate depending on the hit percentage. (a) Standard normal data. (b) Long-tails data. Hit selection methods: □ = classical; ○ = k-means partitioning; △ = sum of the average squared inside-cluster distances clustering; × = average intercluster distance clustering.

of 10 μM. All data are reported as the percentage residual activity relative to the average of the high controls.

The fact that the original study has identified only 32 hits in both copies (for more detail see, http://hts.mcmaster.ca/Competition_FAQ.html) shows that either some kind of random noise was added to the data during the analysis or that the testing conditions were slightly different for the 2 assay replicates. This could also happen because the plate-to-plate variability was high or the testing conditions were inconsistent from one replicate to another. Thus, we decided to carry out the plate-to-plate clustering analysis that is more appropriate than the hit selection from 2 batches, 1 per replicate, in such a situation.

The distribution function for this data set and its approximation by a Gaussian distribution are shown in **Figure 7**. It is worth noting that for this representation, the experimental data were plate normalized using the zero-mean centering and unit variance standardization. The Gaussian distribution was modeled using the parameters of the experimental data. The classical hit selection threshold was set to $\mu - 3\sigma$. The upper right corner of **Figure 7** shows the data distribution in the hit selection area.

To apply properly a clustering method, it is first necessary to calculate the average of the maximum sigma gaps for the plates of type C (**Fig. 1**). There were precisely 886 plates of type C in the McMaster data set. The average of the maximum sigma gaps on the smallest 20% of elements of these plates was equal to 0.485. Using the approximation by the polynomial (formula 7), we obtained the value of the optimal sigma-gap constant to be used for identifying hit clusters on the plates of type C. The sigma-gap constant was equal to 0.81 in this case.

The classical hit selection procedure found 429 hits in the McMaster data, the k-means partitioning method found 467

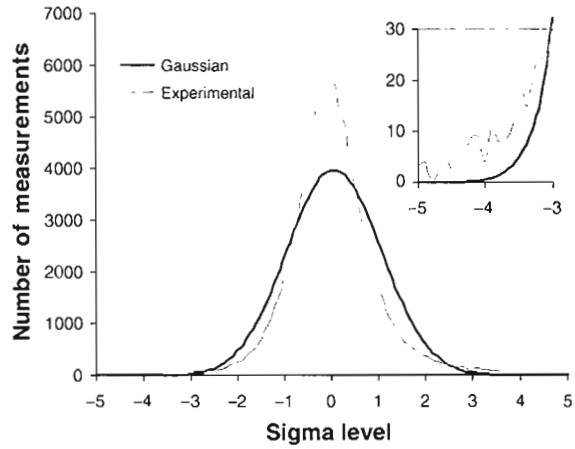


FIG. 7. Distribution of measurements at the McMaster *Escherichia coli* assay (1250 plates) and its comparison to a Gaussian distribution.

hits, the SASD method found 465 hits, and the AICD method found 757 hits. The bar chart representing the hit selection results is shown in **Figure 8a**. The k-means partitioning method detected 39 hits that were not identified as hits by the classical hit selection procedure (see **Fig. 8b**), the SASD method found 38 extra hits, and the AICD method found 328 extra hits. On the other hand, almost all hits detected by the classical procedure were confirmed by the clustering methods: The k-means method missed only 1 classical hit, the SASD method missed only 2 hits, and the AICD method did not miss any of the classical hits. The intersections between the sets of hits provided by the 3 considered clustering strategies are

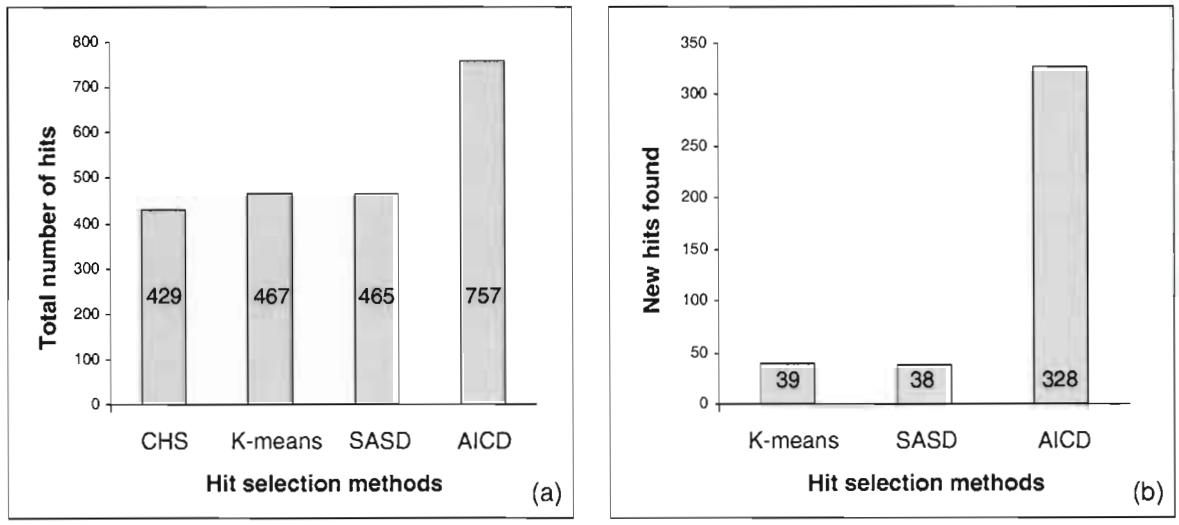


FIG. 8. Comparison of the results provided by 4 hit selection methods for the considered McMaster University experimental HTS screen. (a) Total number of selected hits. (b) Number of hits found by the 3 clustering methods and not found by the classical hit selection procedure.

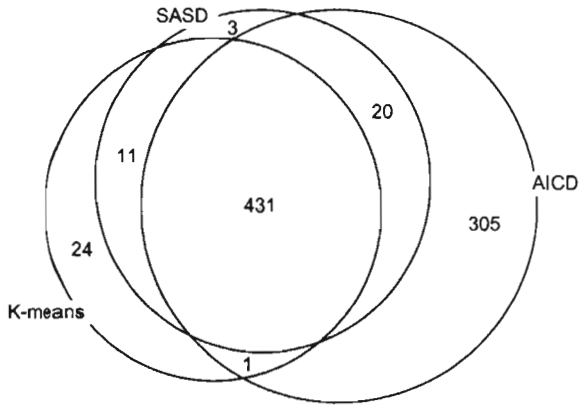


FIG. 9. Intersections between the 3 hit sets found by the 3 considered clustering methods for the McMaster University experimental screen.

shown in **Figure 9**. Note that the results obtained by k-means partitioning and SASD were very similar, whereas the AICD procedure found 305 hits that were not detected by any other clustering method.

Let us compare in more detail the results provided by the classical hit selection procedure and those obtained by the SASD method. The classical procedure identified 429 hits in the McMaster assay, whereas the SASD method found 465 hits in the same data set (total increase of 36 hits). Note that 26 of these 36 hits were found on the plates of type C (**Fig. 1**); that is, their values as well as all values in their clusters were bigger than the classical threshold of $\mu - 3\sigma$. These 26 hits were found in the clusters containing 3 and 4 elements located on 8 plates. Only 2 hits detected by the classical approach were not identified as hits by SASD. These 2 elements were located on the plates of type A (**Fig. 1**) that had nonempty hit clusters. Both values of the nondetected classical hits were close to the threshold of $\mu - 3\sigma$. Thus, it is likely that these 2 elements identified as hits by the classical method were false positives. The SASD method also found 12 hit elements with measured values bigger than $\mu - 3\sigma$ located in the clusters on the plates of type B (**Fig. 1**).

Note that each compound of the considered McMaster University assay had 2 copies and thus was tested twice during this screen (for more detail, see the screen description on the McMaster University Web site). Both the classical hit selection procedure and SASD selected 36 compounds were confirmed as hits for both copies of the compound. However, the 36 compounds confirmed twice by the classical procedure were different in 1 compound from the 36 compounds confirmed twice via SASD.

CONCLUSION

We described a new approach to select hits in HTS data. The presented approach is based on the cluster analysis of assay measurements. We considered 3 clustering techniques that enabled us to improve classical hit selection results in the simulations with random data. Two clustering schemes were examined, with the 1st one considering each plate as an independent experiment and the 2nd one processing all the data as a single batch. We agree with Malo et al[10] that improving hit specificity and sensitivity cannot be met by technological and organizational improvements alone and that improvements in data analysis methods are needed to fulfill the promise of HTS.

The results of the considered clustering techniques depend on the data distribution as well as on the plate size and the number of plates. Given an experimental HTS data set, we recommend trying clustering methods on the random data that have the identical distribution and are arranged in the same number of plates of the same size. The random data should be generated and modeled using the mean values and standard deviations of the experimental data. This will allow one to choose plausible clustering methods and parameters for hit selection in the experimental data. The simulations with random data can be done by analogy with the computational experiments described in this article.

Based on the simulations with random data, we recommend using k-means partitioning or the SASD clustering method. In general, the 2 methods have shown better performances than the AICD method and classical hit selection. However, if it is more important to have a low number of false positives in a particular HTS assay, the AICD method can be considered as well. It is worth noting that it is possible to combine the clustering hit selection methods with data correction methods and the classical hit selection. Moreover, one can also combine the hit selection methods, for example, searching first for the initial hit cluster using the k-means partitioning method and then applying the AICD method to the remaining plate elements (in this study, the k-means partitioning was the best method in terms of true hits, and AICD showed the best performance in terms of false positives). A more conservative option would consist of the selection of compounds that were identified as hits by all considered clustering methods. It also would be interesting to test the popular k-medoids method[16] in the hit selection context and to incorporate the available chemical information about the compounds into the clustering methods (e.g., molecular weight, reactivity level, etc.). This would lead to multivariable data sets that provide the possibility of using weighting variables.

The experiments described in this article showed that the application of different clustering techniques leads to different hit selection results. Therefore, it would be interesting to design and carry out significance tests for the hit selection methods. One can also simulate and analyze some other types of data to confirm the advantages of the cluster-based hit selection.

Finally, we also suggest trying methods of machine learning that would combine the obtained information on experimental HTS data with chemical description parameters of the tested compounds. Such a combination of quantitative and qualitative descriptors seems to be very promising for an efficient selection of high-quality drug candidates.

ACKNOWLEDGMENTS

We thank Genome Quebec for funding this project. We are also thankful to Dr. Robert Nadon (McGill University, Montreal, Canada) for his help with the data simulations. We are also thankful to an anonymous referee for her or his helpful comments.

REFERENCES

1. Heuer C, Haenel T, Prause B: A novel approach for quality control and correction of HTS data based on artificial intelligence. In *Pharmaceutical Discovery & Development Report 2003/03*. Oxford, UK: PharmaVentures Ltd, 2002.
2. Gunter B, Brideau C, Pikounis B, Liaw A: Statistical and graphical methods for quality control determination of high throughput screening data. *J Biomol Screen* 2003;8:624-633.
3. Brideau C, Gunter B, Pikounis W, Liaw A: Improved statistical methods for hit selection in high-throughput screening. *J Biomol Screen* 2003;8: 634-647.
4. Heyse S: Comprehensive analysis of high-throughput screening data. *Proceedings of SPIE* 2002;4626:535-547.
5. Zhang JH, Chung TDY, Oldenburg KR: A simple statistic parameter for use in evaluation and validation of high throughput screening assays. *J Biomol Screen* 1999;4:67-73.
6. Zhang JH, Chung TDY, Oldenburg KR: Confirmation of primary active substances from high throughput screening of chemical and biological populations: a statistical approach and practical considerations. *J Comb Chem* 2000;2:258-265.
7. Kevorkov D, Makarenkov V: Statistical analysis of systematic errors in high-throughput screening. *J Biomol Screen* 2005;10:557-567.
8. Makarenkov V, Kevorkov D, Gagarin A, Zentilli P, Malo N, Nadon R: *An Efficient Method for the Detection and Elimination of Systematic Error in High-Throughput Screening*. Unpublished manuscript, 2006.
9. Makarenkov V, Kevorkov D, Zentilli P, Gagarin A, Malo N, Nadon R: HTS-Corrector: software for the statistical analysis and correction of experimental high-throughput screening data, *Bionformatics* 2006;22:1408-1409.
10. Malo N, Hanley JA, Cerquozzi S, Pelletier J, Nadon R: Statistical practice in high-throughput screening data analysis. *Nat Biotechnol* 2006;24: 167-175.
11. Zolli-Juran M, Cechetto JD, Hartlen R, Daigle DM, Brown ED: High throughput screening identifies novel inhibitors of *Escherichia coli* dihydrofolate reductase that are competitive with dihydrofolate. *Bioorg Med Chem Lett* 2003;13:2493-2496.
12. Arabie P, Hubert LJ, De Soete G: *Clustering and Classification*. Hackensack<N> NJ: World Scientific, 1996.
13. MacQueen J. Some methods for classification and analysis of multivariate observations. In Le Cam LM, Neyman J (eds): *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1, Statistics. Berkeley: University of California Press, 1967.
14. Legendre P, Legendre L: *Numerical Ecology*. 2nd ed. Amsterdam: Elsevier Science BV, 1998.
15. Char BW, Geddes KO, Gonnet GH, Monagan MB, Watt SM: *Maple Reference Manual*. New York: Springer-Verlag, 1988.
16. Kaufman L, Rousseeuw PJ: *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley & Sons, 1990.

Address reprint requests to:
Andrei Gagarin
Laboratoire LaCIM
Université du Québec à Montréal
C.P. 8888, succursale Centre-Ville
Montréal (Québec), Canada, H3C 3P8

E-mail: gagarin@lacim.uqam.ca

4.4 « An efficient method for the detection and elimination of systematic errors in high-throughput screening »

Cet article présente une discussion détaillée sur les problèmes des erreurs systématiques, rencontrées dans les procédures de criblage à haut débit, et sur des méthodes de correction pouvant être utilisées pour minimiser leurs effets avant l'étape de recherche des « hits ».

Après la présentation des problèmes pouvant être rencontrés, trois méthodes proposées pour la correction des erreurs systématiques sont testées : « *median polish* », « *B Score* » et « *well correction* ». Ces tests sont effectués sur plusieurs jeux de données simulées, en variant des paramètres, tels que : le pourcentage des « hits » ajoutés, l'amplitude des erreurs systématiques ajoutées et la distribution théorique des données. Finalement, on compare ces résultats aux méthodes classiques de recherche des « hits » par seuil prédéfini sans correction préalable, que ce soit plateau par plateau, ou en considérant toute la campagne HTS au complet. Les mêmes tests sont effectués sur les données de la compétition de l'université McMaster (1250 plateaux d'un test sur l'inhibition de *l'Escherichia coli dihydrofolate reductase*).

L'article conclut sur l'avantage de la méthode de correction par puits, par rapport à ses rivales. Cette conclusion est valable en présence ou non d'erreurs systématiques dans les données étudiées.

Makarenkov V., Zentilli P., Kevorkov D., Gagarin A., Malo N. et Nadon R. (2007) « An efficient method for the detection and elimination of systematic errors in high-throughput screening. » *Bioinformatics*, **23**, 1648-1657.

ORIGINAL PAPER

Vol. 23 no. 13 2007, pages 1648–1657
doi:10.1093/bioinformatics/btm145

Data and text mining

An efficient method for the detection and elimination of systematic error in high-throughput screening

Vladimir Makarenkov[1,*], Pablo Zentilli[1], Dmytro Kevorkov[1], Andrei Gagarin[1], Nathalie Malo[2,3] and Robert Nadon[2,4]

[1]Department d'informatique, Université du Québec à Montreal, C.P.8888, s. Centre Ville, Montreal, QC, Canada, H3C 3P8, [2]McGill University and Genome Quebec Innovation Centre, 740 Dr. Penfield Ave., Montreal, QC, Canada, H3A 1A4, [3]Department of Epidemiology, Biostatistics, and Occupational Health, McGill University, 1020 Pine Av. West, Montreal, QC, Canada, H3A 1A4 and [4]Department of Human Genetics, McGill University, 1205 Dr. Penfield Ave., N5/13, Montreal, QC, Canada, H3A 1B1

Received on December 7, 2006; revised on February 22, 2007; accepted on April 10, 2007

Advance Access publication April 26, 2007

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: High-throughput screening (HTS) is an early-stage process in drug discovery which allows thousands of chemical compounds to be tested in a single study. We report a method for correcting HTS data prior to the hit selection process (i.e. selection of active compounds). The proposed correction minimizes the impact of systematic errors which may affect the hit selection in HTS. The introduced method, called a *well correction*, proceeds by correcting the distribution of measurements within wells of a given HTS assay. We use simulated and experimental data to illustrate the advantages of the new method compared to other widely-used methods of data correction and hit selection in HTS.

Contact: makarenkov.vladimir@uqam.ca

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

High-throughput screening (HTS) is a modern technology used for the identification of pharmacologically active compounds (i.e. hits). In screening laboratories, testing more than 100 000 compounds a day has become routine. Automated mass screening for pharmacologically active compounds is now widely distributed. It serves for the identification of chemical compounds as starting points for optimization (primary screening), for the determination of activity, specificity, physiological and toxicological properties of large libraries (secondary screening) and for the verification of structure-activity hypotheses in focused libraries (tertiary screening) (Heyse, 2002). The lack of standardized data validation and quality assurance processes has been recognized as one of the major hurdles for successful implementing high-throughput experimental technologies (Kaul, 2005). Therefore, automated quality assessment and data correction systems need to be applied to biochemical data in order to recognize and eliminate experimental artefacts that might confound with important biological or chemical effects. The description of several methods for quality control and correction of HTS data can be found in Brideau *et al.* (2003), Gagarin *et al.* (2006a), Gunter *et al.* (2003), Heuer *et al.* (2003), Heyse (2002), Kevorkov and Makarenkov (2005), Makarenkov *et al.* (2006), Malo *et al.* (2006) and Zhang *et al.* (1999, 2000).

Various sources of systematic errors can affect experimental HTS data, and thus introduce a bias into the hit selection process (Heuer *et al.*, 2003), including:

- Systematic errors caused by aging, reagent evaporation or cell decay which can be recognized as smooth trends in the plate means/medians.
- Errors in liquid handling and malfunction of pipettes which can generate localized deviations from expected values.
- Variation in incubation time, time drift in measuring different wells or different plates and reader effects which can be recognized as smooth attenuations of measurements over an assay.

Random errors produce noise that cause minor variation of the hit distribution surface. Systematic errors generate repeatable local artifacts and smooth global drifts, which become more noticeable when computing a hit distribution surface (Kevorkov and Makarenkov, 2005). Often systematic errors create border, row or columns effects, resulting in the measurements in certain rows or columns that are systematically over or underestimated (Brideau *et al.*, 2003). This article introduces a new method allowing one to minimize the impact of systematic error on the hit selection process. We propose to examine the hit distribution of raw data and fit the data variation within each well to correct the data at hand. The comparison of the new method to other data correction techniques used in HTS is described in the Simulations section. The latter section is followed by an application example, where we carried out the identification of active compounds in the raw and well-corrected HTS assay generated at McMaster University.

*To whom correspondence should be addressed.

2 MATERIALS AND METHODS

2.1 Experimental data

In this article, we examine an experimental data set generated at the HTS Laboratory of McMaster University. This test assay was proposed as a benchmark for the McMaster Data mining and docking competition (http://hts.mcmaster.ca/Downloads/82BFBEB4-F2A4-4934-B6A8-804CAD8E25A0.html; Elowe *et al.*, 2005). It consists of a screen of compounds that inhibits the *Escherichia coli* dihydrofolate reductase. Each compound was screened twice: two copies of 625 plates were run through the screening machines. This gives 1250 plates in total, each having wells arranged in eight rows and 12 columns (the columns 1 and 12 containing controls were not considered in this study). The assay conditions reported in Elowe *et al.* (2005) were the following: assays were carried out at 25°C and performed in duplicate. Each 200 µl reaction mixture contained 40 µM NADPH, 30 µM DHF, 5 nM DHFR, 50 mM Tris (pH 7.5), 0.01% (w/v) Triton and 10 mM β-mercaptoethanol. Test compounds from the screening library were added to the reaction before initiation by enzyme and at a final concentration of 10 µM. The Supplementary Materials section contains more detail on the screening method and plate layout (Fig. 1S) for this assay.

2.2 Data preprocessing and correction in HTS

The analysis of experimental HTS data requires preprocessing to ensure the statistical meaningfulness and accuracy of the data analysis and the hit selection. Ideally, inactive samples should have similar mean values and generate a constant surface. In a real case, however, random errors produce random noise. For a large number of plates considered, the noise residuals should compensate each other in the computation of mean values. Systematic repeatable artifacts become more visible as the number of plates increases (Kevorkov and Makarenkov, 2005). The following steps can be carried out to pre-process experimental HTS data:

(1) Hit and outlier elimination (optional). This elimination can be carried out in order to reduce the influence of hits and outliers on the plates' means and SDs (standard deviations). It can be particularly important when analyzing screens with few (<100) plates.

(2) Within-plate normalization of all samples, which can be done including or excluding hits and outliers, using the *Z-score transformation* (i.e. zero mean and unit variance standardization, Equation 1) or the *Control normalization* (Equation 2) can be carried out. Such transformations should be applied to analyze together experimental HTS data generated under different testing conditions. In case of *Z-score*, the following formula is used:

$$x_i' = \frac{x_i - \bar{x}}{\mathrm{SD}}, \tag{1}$$

where x_i—measured value at well i, x_i'—normalized output value at well i, $\bar{x}$—mean value.

The *control normalization* (i.e. normalized percent inhibition) is based on the following formula:

$$x_i' = \frac{H - x_i}{H - L} \times 100\%, \tag{2}$$

where x_i—measured value at well i, H—mean of high controls, L—mean of low controls and x_i'—evaluated percentage at well i.

The additivity of experimental data is a necessary property that should hold prior to the application of some statistical procedures. Plate means and SDs vary substantially from plate to plate. In order to compare and analyze together experimental data from various plates and data tested under different conditions, all measurements should be normalized.

(3) Data correction of all samples. This step can be conducted using the median polish procedure (Tukey, 1977), the background correction procedure (Kevorkov and Makarenkov, 2005), or the well correction method discussed in this article. The B-score transformation procedure (Brideau *et al.*, 2003; Malo *et al.*, 2006), additionally correcting for row and column biases, can also be carried out. The comparison of the data correction techniques is presented in the Simulation study section.

Also, background plates (i.e. control plates) can be inserted throughout a screen. Background plates are separate plates containing only control wells and no screening compounds. They are particularly useful for calculating the background levels of an assay and help determine whether an assay has sufficient signal which can be reliably detected (Fassina, 2006). Such additional plates enables one to create background signatures that lead to plate-based correction factors on, per well, per row or per column, basis for all other assay plates. The use of background plates gets around the main assumption made for the well correction procedure: when examined across plates, wells should not systematically contain compounds from the same family (see the description of the well correction method subsequently). The main inconvenience of this method is that it does not take into account possible errors that might occur in the plates processed between two background plates.

Note that for certain methods, the corrected data can be easily denormalized in order to obtain a data set scaled as the original one. Analysis of the hit distribution surface of the corrected data can then be carried out using the χ^2-contingency test (see the results in the section 3.2).

2.3 Hit selection process

In the HTS workflow, the bias correction process is followed by hit selection (i.e. inference of active compounds). The selection of hits in the corrected data is often done using a preselected threshold (e.g. $\bar{x} - 3\mathrm{SD}$, in case of an inhibition assay).

Hit selection is a process that should not only consider statistical treatment of HTS data. It should also be used in conjunction with the structure-activity-relationships (SAR) observed using the corrected HTS data (Gedeck and Willett, 2001). In SAR, the basic assumption for all molecule based hypotheses is that similar molecules have similar activities. The quality of hits is improved if SAR is taken into consideration. For instance, the likelihood that an identified hit is an artifact grows if a large number of highly related compounds was confirmed as inactive in the corrected data.

2.4 Analysis of hit distribution

The presence of systematic errors in an assay can be detected through the analysis of its hit distribution surface (Kevorkov and Makarenkov, 2005). This surface can be computed by estimating the number of selected hits within each well location. In the case of randomly distributed compounds, hits should be distributed evenly over the well locations. In the example presented in Figure 1, we considered the normalized McMaster assay (Elowe *et al.*, 2005; Zolli-Juran *et al.*, 2003) comprising 1250 plates arranged in eight rows and 10 columns (the control columns were not considered). For each well, we estimated the number of experimental values that deviated from the plate means by more than $\bar{x} - \mathrm{SD}$ (i.e. number of values that are lower than $\bar{x} - \mathrm{SD}$ at each well across plates).

As the common strategy utilizes the $\bar{x} - 3\mathrm{SD}$ threshold for the hit selection, the considered data comprise all hits as well as all values close to them. The substantial variation of the measurements shown in

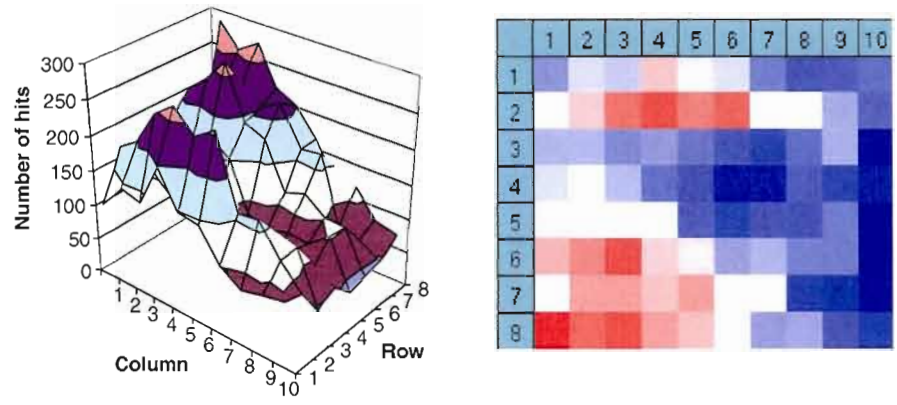


Fig. 1. Hit distribution surface for the McMaster data (1250 plates). Values deviating from the plate means for more than one SD were taken into account during the computation. (a) Well positional effects in 3D are shown; (b) Well, row and column positional effects are shown.

Figure 1 illustrates the presence of systematic errors in this assay (see the first two columns in Table 5S for the results of the χ^2-contingency test conducted on this surface). The detailed analysis of the McMaster hit distributions obtained for different thresholds is presented in the section 3.2.

2.5 Compared methods

The five following methods were compared in this study. Methods 1 and 2 do not involve any data correction, whereas Methods 3–5 proceed by the correction of systematic error before hit selection.

2.5.1 Method 1 Classical hit selection using the value $\bar{x} - c \times \mathrm{SD}$ as a hit selection threshold, where the mean value $\bar{x}$ and SD are *computed separately for each plate* and c is a preliminary chosen constant. All values lower than or equal to the threshold value are considered as hits.

This method should be applied cautiously when samples are not randomly assigned to plates. Specifically, Method 1 can lead to increased rates of false positives and false negatives when analyzing plates with compounds belonging to the same family (e.g. in case of an inhibition assay, a high concentration of small hit values in a plate can lead to their transformation into false negative hits, whereas a high concentration of large values can transform some of the lowest non hit measurements into false-positive hits).

2.5.2 Method 2 Classical hit selection using the value $\bar{x} - c \times \mathrm{SD}$ as a hit selection threshold, where the mean value $\bar{x}$ and SD are computed *over all assay values*, and c is a preliminary chosen constant. This method can be chosen when we are certain that all plates of the given assay were processed under the 'same testing conditions'.

Method 2 should be applied cautiously when the plates have been tested either over numerous days and/or by different machines (robots, readers, etc). Experience suggests that the 'same conditions' requirement may be violated under these circumstances. Moreover, in at least some circumstances, the variability of the various compounds does not appear to be constant but rather follows an inverse gamma distribution (Malo *et al.*, 2006). In the latter case, the pooled variance of Method 2 provides only one component of an individual compound's variance—the other component would be provided by the compound specific variance as estimated by replicates. However, if the differences among the compound variances are very small, then Method 2 is expected to do well.

2.5.3 Method 3 *Median polish* procedure (Tukey, 1977) can be used to remove the impact of systematic error. Median polish (Equation 3) works by alternately removing the row and column medians, and continues until the proportional reduction in the sum of absolute residuals is less than a fixed value ε or until a fixed number of iterations has been carried out. The residual (r_{ijp}) of the measurement for row i and column j on the p-th plate is obtained by fitting a two-way median polish, and is defined as follows:

$$r_{ijp} = x_{ijp} - \hat{x}_{ijp} = x_{ijp} - \left(\hat{\mu} + \hat{R}_{ip} + \hat{C}_{jp}\right). \tag{3}$$

The residual is defined as the difference between the observed result (x_{ijp}) and the fitted value ($\hat{x}_{ijp}$), which is defined as the estimated average of the plate ($\hat{\mu}_p$) + estimated systematic measurement offset for row i on plate $p\left(\hat{R}_{ip}\right)$ + estimated systematic measurement column offset for column j on plate $p\left(\hat{C}_{jp}\right)$. Thus, the matrix of residuals **R** replaces the original matrix in the further computations. In our simulations, Method 2 was applied to the values of the matrix **R** in order to select hits.

2.5.4 Method 4 *B-score* (Equation 4) normalization procedure (Brideau *et al.*, 2003) is designed to remove plate row/column biases in HTS (Malo *et al.*, 2006). The residual (r_{ijp}) of the measurement for row i and column j on the p-th plate is obtained by fitting a two-way median polish. In addition, for each plate p, the adjusted median absolute deviation (MAD_p) is obtained from the r_{ijp}'s. The B score is calculated as follows:

$$B\text{-}score = \frac{r_{ijp}}{(1.4826 \times MAD_p)}, \tag{4}$$

where $MAD_p = \mathrm{median}\{|r_{ijp} - \mathrm{median}(r_{ijp})|\}$. The raw MAD used in the B-score calculation is rescaled by the multiplicative constant of 1.4826. To select hits, this computation was followed by Method 2, applied to the **B-score** matrix. Here we considered the version of B-score presented in Malo *et al.* (2006); the latter version of the method does not include the smoothness parameter used by Brideau *et al.* (2003). The main assumption that must be met to apply Methods 3 and 4 is that the compounds should be randomly distributed within each plate.

Any systematic row or column placement of compounds within a plate will bias the results given by these two methods.

2.5.5 Method 5 *Well correction* procedure described below followed by Method 2.

The first two methods are the classical hit selection strategies not involving any correction of systematic bias, whereas the last three methods combine a data preprocessing procedure with the hit selection by Method 2. The results of the median polish and B-score methods were generated using the S-PLUS package (S-PLUS manual, 2006).

2.6 Well correction procedure

To be able to apply the new correction procedure to experimental data sets, the following assumptions about HTS data should be made: screened samples can be divided into active and inactive; the majority of the screened samples are inactive; values of the active samples differ substantially from the inactive ones; and systematic error causes a repeatable influence on the measurements within wells across plates. Also, wells, across plates, should not systematically contain compound samples belonging to the same family. However, it does not require the randomization of samples within plates, which seems to be a much more frequent situation in the real HTS campaigns. We studied the chemical structure of compounds within each well of the McMaster data set and have not found any systematic pattern in the compound distribution. Usually, each well location contains a large number of samples across plates (e.g. 1250 samples for the McMaster assay), and small systematic compound placements are very unlikely to bias the data.

The Z-score normalization produces a modified data set in which the values within each plate are zero-mean centered, whereas SD and variance are equal to unity. Once the data are plate-normalized, we propose to analyze the values within each well measured across all assay plates. If no systematic error is present in the data set, the *distribution of measurements within wells* should be also close to a zero-mean centered one with a SD close to unity. The *well correction method* consists of two main steps:

(1) *Least-squares approximation of the data carried out separately for each well of the assay.*

(2) *Z-score normalization of the data within each well location of the assay.*

The real distribution of values can differ substantially from the ideal one. The example presented in Figure 2S features the measurements obtained for the well located in column 1 and row 8 of the McMaster data (Elowe *et al.*, 2005). The mean of the observed values is -0.37. Such a deviation suggests the presence of systematic error in this well location. Experimental values for a specific well location can also have ascending or descending trends. The well correction procedure first discovers these trends using the linear least-squares approximation; note that the fitting by a polynomial of a higher degree can be also carried out instead of the linear approximation. Thus, the obtained trend (e.g. a straight line $y = ax + b$ in case of the linear fitting, where x denotes the plate number and y denotes the plate-normalized measurement) is subtracted from or added to the original measurements bringing the mean value of this well to zero. Because the optimal parameters are sought for each well location of the assay independently, well correction has more fitting parameters than B-score and median polish. For the analysis of large industrial assays, more sophisticated functions (e.g. higher degree polynomials or spline functions) can be also used. Alternatively, an assay can be divided into intervals and a particular trend function characterizing each interval can be determined through approximation.

Second, the well normalization using the Z-score normalization (Equation 1) of the well measurements is carried out *independently for each well location*. Then, we can select hits in the corrected data and reexamine the hit distribution surface.

3 RESULTS AND DISCUSSION

3.1 Simulation study

To demonstrate the effectiveness of the well correction procedure, we first carried out simulations with random data. Specifically, we considered three types of random symmetrically distributed data: standard normal, heavy tailed (positive kurtosis) and light tailed (negative kurtosis) distributions. As with the McMaster data, our data sets consisted of 1250-plate assays, each plate comprising wells arranged in eight rows and 10 columns. First, three random null data sets (i.e. without hits) were generated. The hit selection procedure was carried out on these data and the false positive hit rates reported in Table 1 were found for the five different hit selection methods presented earlier.

Hit selection thresholds equal to $\bar{x} - 3\text{SD}$ for the standard normal, to $\bar{x} - 2.042\text{SD}$ for the light tailed, and to $\bar{x} - 3.420\text{SD}$ for the heavy tailed data, were considered. The hit selection thresholds for the heavy and light tailed data were chosen to have approximately the same hit percentage found by the hit selection method based on the assay parameters (Method 2) for the three raw data sets ($\sim 0.14\%$ of hits; i.e. 140 hits). Since the simulated data did not contain any hits, the hits identified by the methods were false-positives by definition. Note that Method 1 was very sensitive to the data distribution; it found the lowest number of false-positives in the case of the heavy tailed (83 hits) and standard normal distributions (104 hits), but 642 false-positive hits in case of the light tailed data. The median polish and B-score methods were unstable, yielding the most false positives (2685 and 2676 for the light tailed data, and 288 and 361 for the heavy tailed data, respectively). The most stable results were obtained by Method 2 and the well correction procedure. As expected, the largest percentage of the false-positive hits was found in the light tailed data. For each type of random data we then generated and added to plates k percent of hits, where k was consequently taking values 0.5, 1, 1.5, 2, 2.5 and 3%, whose locations and values were chosen arbitrarily; the probability of each well in each plate to contain a hit was k percent. One thousand replicates of data of

Table 1. False positive hit rate for the five preprocessing methods. Random data without hits having standard normal, heavy and light tailed distributions were considered

Distributions\Methods	Standard normal	Light tailed	Heavy tailed
Threshold	$\bar{x} - 3\text{SD}$	$\bar{x} - 2.042\text{SD}$	$\bar{x} - 3.420\text{SD}$
Method 1	0.104%	0.642%	0.083%
Method 2	0.140%	0.138%	0.137%
Method 3	0.385%	2.685%	0.288%
Method 4	0.538%	2.676%	0.361%
Method 5	0.138%	0.292%	0.121%

each distribution and for each hit percentage were generated. The values of hits were assumed to have a standard normal distribution with the parameters $N(\bar{x} - 5SD, SD)$ for the standard normal, $N(\bar{x} - 4.9SD, SD)$ for the light tailed and $N(\bar{x} - 5.9SD, SD)$ for the heavy tailed distributions, respectively, where $\bar{x}$ is the mean value and SD is the standard deviation of the observed plate.

Second, row and column biases were generated as follows. For each row and each column of a given random assay we generated a systematic error that was identical for all assay plates. We also added a small random error to all assay measurements. Therefore, the error-perturbed value, x'_{ijp}, of the measurement in row i and column j on the p-th plate was obtained as follows:

$$x'_{ijp} = x_{ijp} + s_i + s_j + rand_{ijp}, \quad (5)$$

where x_{ijp} is the observed result in well ij of plate p, s_i is the systematic error present in row i, s_j is the systematic error present in column j and $rand_{ijp}$ is the random error in well ij of plate p (Table 1S, case a). The variables s_i and s_j in Equation (5) had a standard normal distribution with parameters $N(0, c)$, where the variable c was consequently taking the values 0, 0.6SD, 1.2SD, 1.8SD, 2.4SD and 3SD. For each value of the variable c, a different set of assays was generated and tested. For all values of the parameter c, the random error *rand* was always distributed according to a standard normal low with parameters $N(0, 0.6SD)$.

We carried out the five preprocessing methods described earlier, choosing as hits the measurements with the values lower than $\bar{x} - 3SD$, $\bar{x} - 2.04SD$ and $\bar{x} - 3.42SD$, for the standard normal, light tailed, and heavy tailed data, respectively. Statistics describing the impact of systematic error on the hit selection process are reported in Tables 2S–4S. Specifically, the hit detection rate as well as the false positive and the false negative rates were computed during the simulations. The results in Tables 2 and 3S,4S are indicated for the data with 1% of added hits whereas the systematic error varied from 0 to 3.0SD. The hit detection rates for the five competing methods corresponding to the systematic error of 1.2SD are depicted in Figures 2 and 3S, 4S. As the tables and graphics suggest, the well correction procedure showed the most stable behavior regardless the data distribution, level of systematic bias, and hit percentage. In all situations, well correction, median polish and B-score methods removed systematic error regardless of amount of bias (Figures 2, 3S, 4S, a and c). However, the well correction procedure generally outperformed the median polish and B-score methods. The two latter methods were able to remove systematic trend and return the correct residuals in the easiest cases, but they often converged to a local instead of a global minimum when the data structure was rather fuzzy. We can also observe that Method 2 based on the assay parameters was more precise than Method 1 using the plates' parameters. Consequently, when the testing conditions are similar for all plates of the given assay, treating all assay measurements as a whole batch should be preferred to the plate-by-plate analysis. On the other hand, the usage of the well correction procedure can be advocated for any type of data regardless of hit percentage. Compared to the four other methods, the well correction procedure was particularly accurate as to the false-negative rate (Tables 2S–4S). At the same time, when the well correction was applied to the data free of noise, it did not have any negative influence to the false-positive rate (Table 1).

The B score method with this type of normally distributed constant variance data did not perform well, although the median for the hits was separated from the median for the non-hits to a greater extent than in the well correction method which, in turn, was less separated from the non-hits than raw data (Fig. 8S). This effect was offset, however, by the increased variance for both the non-hits and the hits. The B-score method improved accuracy somewhat but at the high cost of a large decrease in precision. Accordingly, B-score method should not be used unless there is evidence of row or column effects.

We also constructed the Receiver Operating Characteristic (ROC) curves for the five methods under study. ROC curves provide a graphical representation of the relationship between the true-positive and false-positive prediction rate of a model. There are many advantages to this approach, including that thresholds do not need to be determined in advance.

The y-axis corresponds to the *sensitivity* of the model, i.e. how well the model is able to predict true positives (real cleavages); the y-coordinates are calculated as follows:

$$Y = \frac{TP}{(TP + FN)}, \quad (6)$$

where TP is the number of true positives and FN is the number of false negatives. The x-axis corresponds to 1-*specificity*, i.e. the ability of the model to identify true negatives. An increase in specificity (i.e. a decrease along the x-axis) results in an increase in sensitivity. The x-coordinates are calculated as follows:

$$X = 1 - \left[\frac{TN}{(TN + FP)}\right], \quad (7)$$

where TN is the number of true negatives and FP is the number of false positives. The greater the sensitivity at high specificity values (i.e. high y-axis values at low x-axis values) the better the model. A numerical measure of the accuracy of the model can be obtained from the area under the curve, where an area of 1.0 signifies near perfect accuracy, while an area of less than 0.5 indicates that the model is worse than just random. Figure 3 illustrates the ROC curves associated with the five methods compared in this article. The curves were obtained for the standard normal data with 1% of added hits without (a) and with (b) systematic error. The ROC curves confirm the conclusions that can be made while observing the methods' performances reported in Table 2S and depicted in Figure 2. The well correction procedure and Method 2 provide the best results for data free of systematic bias (Fig. 3a), whereas median polish and B-score methods fail to recover correct hits in this situation. After the addition of systematic noise (Fig. 3b) well correction procedure outperformed the four other methods, whereas the performances of Methods 1 and 2, not assuming any correction of systematic bias, decreased compared to the case of the error free data.

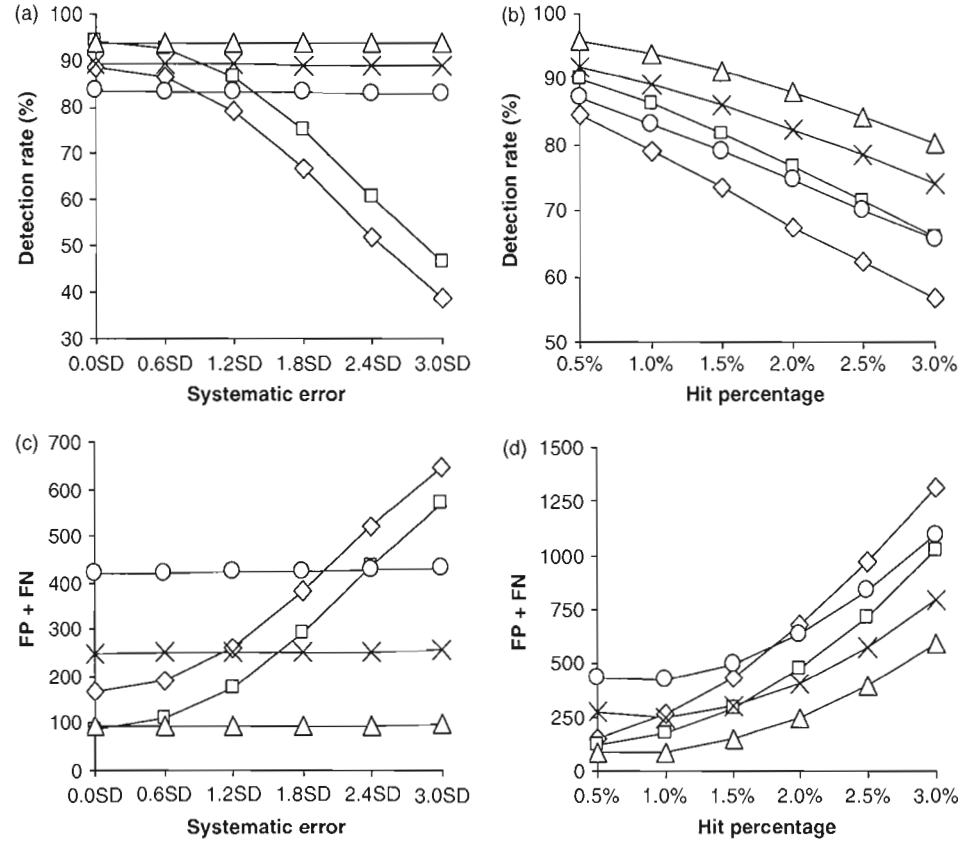


Fig. 2. (see also Table 2S). True hit rate and total of the number of false-positive and false-negative hits for the noisy standard normal data *with systematic error stemming from row × column interactions which are constant across plates*. The results were obtained with the methods using plates' parameters (i.e. Method 1, open diamond), assay parameters (i.e. Method 2, open square), median polish (cross), B-score (open circle) and well correction (open triangle). The abscissa axis indicates the noise factor (a and c—with fixed hit percentage of 1%) and the percentage of added hits (b and d—with fixed error rate of 1.2SD).

Finally, for the noisy standard normal data only, we carried out simulations with four additional types of error. The data generation diagram for these simulations is presented in Table 1S (see cases b–e). The following additional error conditions were considered:

(b) Systematic error with column effects across plates (Fig. 5S).

(c) Systematic error varying from well to well (no row × column interactions involved) across plates (Fig. 6S).

(d) Systematic error stemming from row × column interactions and changing from plate to plate (Fig. 4).

(e) Random error only varying from well to well and from plate to plate (Fig. 7S).

These situations account for the most realistic scenarios, although it is of course not possible to represent all contingencies. For these additional data sets, the well correction procedure was generally more accurate than the four other methods. The only case when the B-score method outperformed the well correction procedure was the case where systematic error stemmed from row × column interactions, which were changing from plate to plate (i.e. systematic error was not constant across plates, Fig. 4) and this error was sufficiently large (1.2SD and more for the true hit rate, and 2.3SD and more for the sum of false positives and false negatives).

3.2 Well correction of the McMaster data

We fitted the McMaster assay data to a Gaussian distribution (Fig. 9S). The raw values were first plate normalized using Z-scores. The experimental distribution was evaluated by counting the number of measurements in 0.1SD intervals in the range $(\bar{x} - 5\text{SD}; \bar{x} + 5\text{SD})$. The Gaussian distribution was modeled using the parameters of the experimental one. The hit selection area $(\bar{x} - 5\text{SD}; \bar{x} - 3\text{SD}$ is shown in the upper right corner of Fig. 9S).

One popular hit selection method in high-throughput screening proceeds by fixing a constant threshold (usually, $\bar{x} - 3\text{SD}$) for all considered wells. For an inhibition assay, all measurements that are lower than this threshold are

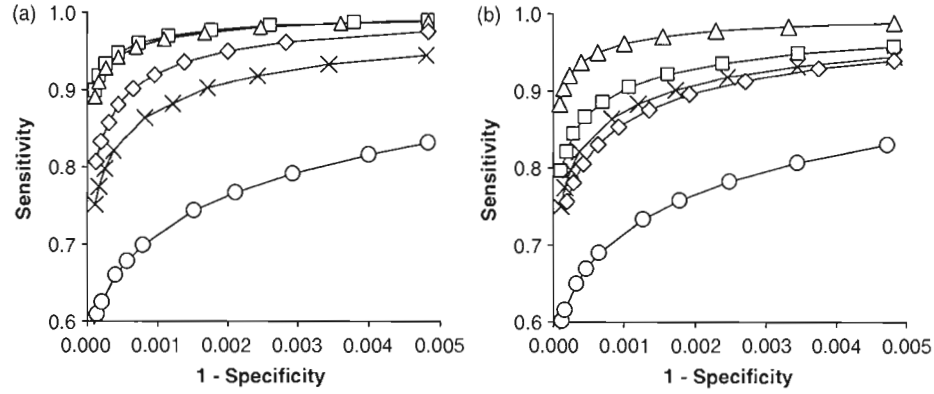


Fig. 3. ROC curves for the noisy standard normal data *with systematic error stemming from row × column interactions which are constant across plates.* The results were obtained using: Z-score (i.e. Method 1, open diamond), raw data (i.e. Method 2, open square), median polish (cross), B-score method (open circle), and the well correction procedure (open triangle). The graph (a) corresponds to the case: 1% of added hits and no systematic error; the graph (b) corresponds to the case: 1% of added hits and systematic error of 1.2SD.

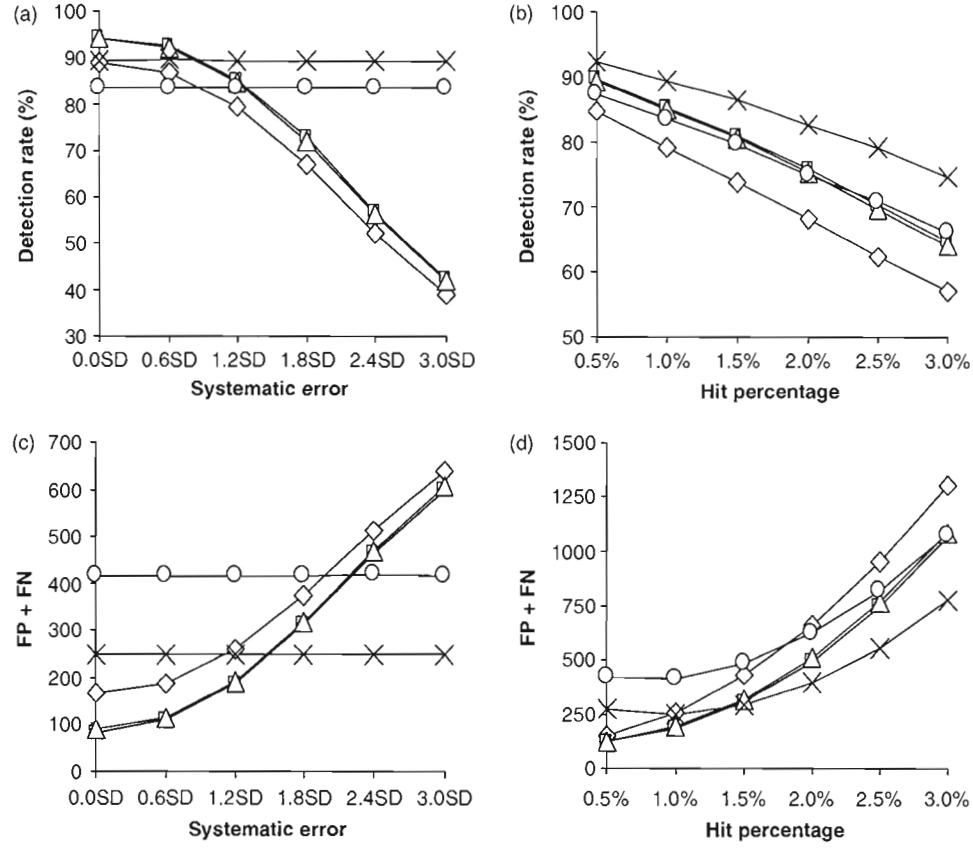


Fig. 4. True hit rate and total of the number of false positive and false negative hits for the noisy standard normal data *with systematic error stemming from row × column interactions which are varying across plates.* The same five methods as in Figure 2 are presented above.

identified as hits. The procedure assumes that the measurement distributions in each well have the same shapes and properties. We verified this assumption while examining the cumulative distribution functions at wells of the McMaster assay (80 functions for 8 × 10 plates). These functions have a broad band of shapes. These differences can be due to systematic biases and can have a substantial impact on the hit selection procedure.

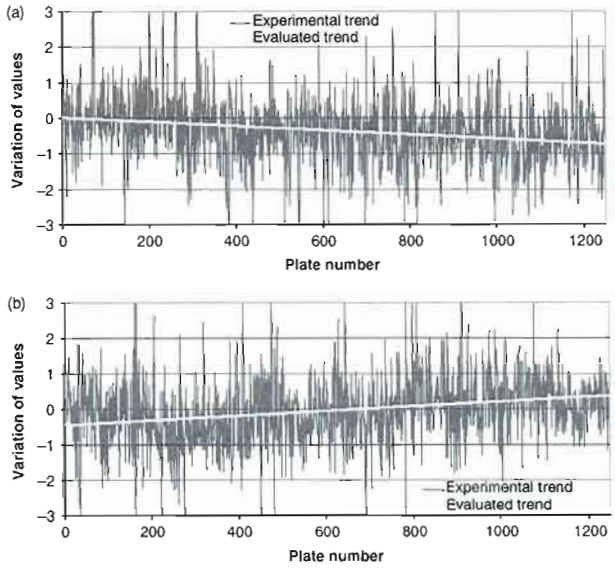


Fig. 5. Variation of the plate-normalized measurements in two different wells along 1250 plates of the McMaster assay; descending (a) and ascending (b) trends are highlighted.

After the plate normalization by Z-scores, the values in each plate are zero-mean centered, and their SD and variance are equal to unity. However, the values in wells, measured across all plates, can have different SDs and variances.

The example in Figure 2S shows that the mean value of the normalized measurements from the well located in column 1 and row 8 (McMaster assay) is −0.37. This deviation is likely to be caused by systematic error. Furthermore, Figure 5 shows that the variation of values within a well can have descending (Fig. 5a) and ascending (Fig. 5b) trends. Thus, to identify hits in the McMaster assay we applied the classical hit selection algorithm based on the assay mean and SD (Method 2) and the well correction procedure (Method 5). The well correction algorithm first evaluates trends using a least-squares approximation. These trends are removed from the experimental data. Then, the algorithm normalizes the modified assay values within each well separately using Z-scores. We examined and compared the hit distribution surfaces obtained using Methods 2 and 5 for different hit selection thresholds. The hit distribution surfaces for the McMaster raw and well-corrected data sets are shown in Figures 6 and 10S. Figure 6 presents the hit distributions for the following hit selection thresholds ($\bar{x}$ – SD $\bar{x} - 1.5$SD and $\bar{x} - 2$SD) and Figure 10S presents the hit distribution surfaces for the thresholds ($\bar{x} - 2.5$SD, $\bar{x} - 3$SD and $\bar{x} - 3.5$SD). Each value on the graphic depicts the number of hits found at the associated well. Figures 6 and 10S suggest that the well correction procedure allows one to attenuate the impact of systematic bias. The improvements in the hit distribution surfaces are more evident for the bigger hit selection thresholds (Fig. 6). For all obtained hit distributions, we also carried out a χ^2-contingency test (with the parameter α of 0.01). In our case, the null hypothesis (H_0) assumes that the observed hit distribution is a constant surface. The results of this test are reported in Tables 5S and 6S.

Figure 6 a (raw data) and b (well-corrected data) depicts the hit distributions for the $\bar{x} - $ SD threshold. The null hypothesis was rejected in both cases (Table 5S). However, the well-corrected data set demonstrated a substantial improvement of the χ^2-statistic compared to the raw data. The χ^2-value decreased from 2377.4 to 204.6 (with critical value equal to 111.1), i.e. this value for the corrected data is about 11.6 times lower compared to the raw ones. Figure 6 c (raw data) and d (well-corrected data) depict the hit distributions for the $\bar{x} - 1.5$SD threshold. After the well correction, the χ^2-coefficient decreased from 1258.4 to 173.6; i.e. it is about seven times lower for the well-corrected data compared to the raw ones, but it was still larger than the χ^2-critical value (111.1). Figure 6 e (raw data) and f (well-corrected data) shows the hit distribution surfaces for the $\bar{x} - 2$SD threshold. The χ^2-contingency test failed to reject the null hypothesis (H_0) for the corrected data (χ^2-value of 74.8) and rejected it for the raw data (χ^2-value of 438.6). Figure 10S illustrates the hit distribution surfaces obtained for the raw and well-corrected data for commonly used hit selection thresholds. The thresholds ($\bar{x} - 2.5$SD, $\bar{x} - 3$SD and $\bar{x} - 3.5$SD) were employed to identify hits in the raw and corrected McMaster data. The null hypothesis, postulating that the hit distribution corresponds to a constant surface, was not rejected for the well-corrected data in case of all three considered hit selection thresholds (Table 6S). For the raw data, the null hypothesis was rejected for all considered thresholds, except $\bar{x} - 3.5$SD, for which the values of the χ^2-coefficient for the raw and corrected data were close (106.2 and 105.3, respectively). This is certainly due to the decrease in the number of hits when lowering the hit selection threshold. The mean numbers of hits per well for the well-corrected data set were usually slightly lower than for the raw data (Tables 5S and 6S). Tables 7S and 8S report the hit numbers per well in the raw and well-corrected McMaster data computed for the $\bar{x} - 3$SD threshold. Even though for the corrected data set the null hypothesis was rejected for the thresholds $\bar{x} -$ SD and $\bar{x} - 1.5$SD, the well correction procedure led to an important reduction of the χ^2-statistic.

We also analyzed the list of active compounds from the Test Library of the McMaster data set. The samples in the original data set were identified as Consensus hits by the organizers of the McMaster HTS competition if both of their replicate measurements were lower or equal to 75% with respect to the reference controls. Only 42 of 50 000 different tested compounds indicated by their MAC-IDs in Table 9S satisfied this property. Our experiments showed that the selection of these 42 replicate compounds can be reached by carrying out Method 1 on the non-normalized data (hit selection by plates, where the values lower than $\bar{x} - c \times$ SD are identified as hits) with the SD coefficient $c = 2.29$. Among the 42 consensus hits identified in such a manner, the competition organizers also identified 14 compounds having well behaved dose–response curves (they are highlighted in Table 9S). We also performed the analysis of the original data set using the well correction procedure which was carried out prior to the selection of hits (Method 5 with hit selection carried out by plate). All other parameters were identical to those of Method 1. With these parameters the application of Method 5 led to the identification of 40 hits. Among these hits, 31 were those found by Method 1 (Table 9S),

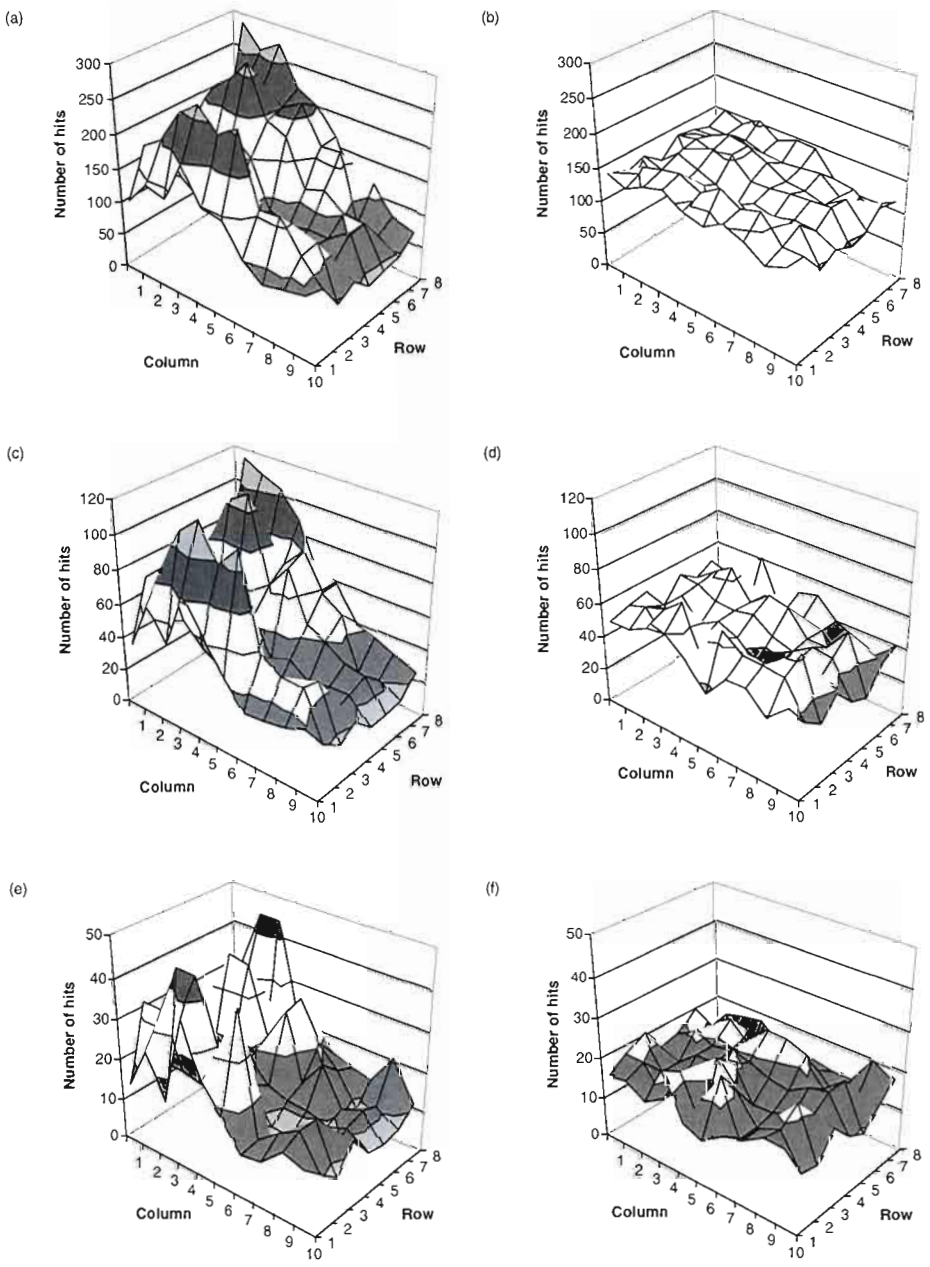


Fig. 6. Hit distributions for the raw (a, c and e) and well-corrected (b, d and f) McMaster data sets obtained for the thresholds $\bar{x}$ – SD (a and b), $\bar{x}$ – 1.5SD (c and d) and $\bar{x}$ – 2SD (e and f).

and nine hits were new. Note that the proportion of compounds with well behaved dose-response curves was better for the well-corrected data (~42%; this percentage does not include the nine new compounds for which the follow up tests were not conducted) than for the raw data (~33%). However, a more detailed study comparing the dose-response behavior of the hits identified by all the five competing methods was not possible because the dose-response follow up information is not available for the nine hits found by Method 5. To conduct a comprehensive study of the five methods compared in this manuscript an experimental dose-response follow-up of the hits obtained using each of these methods is certainly necessary.

It would be quite practical to have the benchmark data sets for which all the results, including the confirmed hits, and testing conditions are known. We think that the scientists from the HTS Laboratory of McMaster University who organized the HTS Data Mining and Docking competition are on the way of doing it: after the announcement of the competition results

a special issue of Journal of Bimolecular Screening was dedicated to the analysis of the test data set (see Elowe *et al.*, 2005 and the articles in the same issue).

4 CONCLUSION

We described a method that can be used to refine the analysis of experimental HTS data by eliminating systematic biases from them prior to the hit selection procedure. The proposed method, called a *well correction*, rectifies the distribution of assay measurements by normalizing data within each considered well across all assay plates. Simulations were carried out with standard normal, heavy and light tailed random data sets. They suggest that the well correction procedure is a robust method that should be used prior to the hit selection process. Well correction generally outperformed the median polish and B-score methods as well as the classical hit selection procedure. In the situations when neither hits nor systematic errors were present in the data, the well correction method showed the performance similar to the traditional method of hit selection. The well correction method also compares advantageously (Gagarin *et al.*, 2006b) to the background correction procedure (Kevokov and Makarenkov, 2005). In the future, it would be interesting to examine how robust the methods are to violations of normality of HTS data.

We also examined an experimental assay generated at the HTS Laboratory of McMaster University. The analysis of the hit distribution of the raw McMaster data set showed the presence of systematic errors. The McMaster data were processed using different hit selection thresholds varying from $\bar{x}$ – SD to $\bar{x} - 3.5$SD. Note that for all considered thresholds, except $\bar{x} - 3.5$SD, for the raw McMaster data, the χ^2-contingency test rejected the null hypothesis, postulating that the hit distribution surface is a constant. The analysis of the well-corrected data sets showed that the new method considerably smoothes the hit distribution surfaces for the $\bar{x}$ – SD and $\bar{x} - 1.5$SD thresholds. When applied to the well-corrected data set, the χ^2-contingency test failed to reject the null hypothesis for the thresholds $\bar{x} - 2$SD to $\bar{x} - 3.5$SD. Furthermore, the simulation study also confirmed that Method 2 based on the assay parameters was more accurate than Method 1 based on the plates' parameters. Therefore, in case of identical testing conditions for all plates of the given assay, all assay measurements should be treated as a single batch.

The HTS Corrector software (Makarenkov *et al.*, 2006, http://www.labunix.uqam.ca/~makarenv/hts.html), including the methods for data preprocessing and correction of systematic error, was developed. HTS Corrector includes all data correction methods discussed is this article (well correction, B-score, and median polish) as well as different methods of hit selection (e.g. Methods 1 and 2 compared in this study). Note that for large industrial assays, a procedure allowing one to divide assays into homogeneous sub-assays has been included in the program. HTS Corrector first establishes a user-defined distance measure between plates and carries out a k-means partitioning algorithm (Legendre and Legendre, 1998; MacQueen, 1967) to form k homogeneous subassays.

ACKNOWLEDGEMENTS

We thank Genome Quebec for funding this project. We also thank two anonymous referees for their helpful comments.

Conflict of Interest: none declared.

REFERENCES

Brideau,C. *et al.* (2003) Improved statistical methods for hit selection in HTS. *J. Biomol. Screen.*, **8**, 634–647.

Elowe,N.H. *et al.* (2005) Experimental screening of dihydrofolate reductase yields a "Test Set" of 50 000 small molecules for a computational data-mining and docking competition. *J. Biomol. Screen.*, **10**, 653–657.

Fassina,G. (2006) HTS of combinatorial libraries. Survey of applications of combinatorial technologies. *Training Course Presentation*: http://www.ics.trieste.it/Documents/Downloads/df3983.pdf.

Gagarin,A. *et al.* (2006a) Clustering techniques to improve the hit selection in HTS. *J. Biomol. Screen.*, **11**, 903–914.

Gagarin,A. *et al.* (2006b) Comparison of two methods for detecting and correcting systematic error in HTS data. In Batagelj,V., Bock,H.H., Ferligoj,A. and Ziberna,A. (eds.). *Data Science and Classification*. IFCS 2006, Studies in Classification, Data Analysis, and Knowledge Organization, Springer Verlag, pp. 241–249.

Gedeck,P. and Willett,P. (2001) Visual and computational analysis of structure-activity relationships in high-throughput screening data. *Curr. Opin. Chem. Biol.*, **5**, 389–395.

Gunter,B. *et al.* (2003) Statistical and graphical methods for quality control determination of HTS data. *J. Biomol. Screen.*, **8**, 624–633.

Heuer,C. *et al.* (2003) A novel approach for quality control and correction of HTS data based on artificial intelligence. *Pharmaceutical Discovery & Development Report*. PharmaVentures.

Heyse,S. (2002) Comprehensive analysis of high-throughput screening data. In *Proceedings of SPIE* 2002, Bellingham, WA **4626**, 535–547.

Kaul,A (2005) The impact of sophisticated data analysis on the drug discovery process. *Business Briefing: Future Drug Discovery 2005*.

Legendre,P. and Legendre,L. (1998) *Numerical Ecology*. 2nd English edn. Elsevier Science BV, Amsterdam, pp. 739–746.

Kevorkov,D. and Makarenkov,V. (2005) Statistical analysis of systematic errors in HTS. *J. Biomol. Screen.*, **10**, 557–567.

MacQueen,J. (1967) Some methods for classification and analysis of multivariate observations. In Le Cam,L.M. and Neyman,J. (eds.) *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. 1, Statistics*. Berkeley: University of California Press.

Makarenkov,V. *et al.* (2006) HTS-Corrector: new application for statistical analysis and correction of experimental data. *Bioinformatics*, **22**, 1408–1409.

Malo,N. *et al.* (2006) Statistical practice in high-throughput screening data analysis. *Nat. Biotechnol.*, **24**, 167–175.

S-PLUS 6. (2006) S-plus programmer's guide. Insightful, URL: http://www.insightful.com/support/doc_splus_win.asp.

Tukey,J.W. (1977) *Exploratory Data Analysis*. Cambridge, MA, Addison-Wesley.

Zhang,J.H. *et al.* (1999) A simple statistic parameter for use in evaluation and validation of HTS assays. *J. Biomol. Screen.*, **4**, 67–73.

Zhang,J.H. *et al.* (2000) Confirmation of primary active substances from HTS of chemical and biological populations: a statistical approach and practical considerations. *J. Comb. Chem.*, **2**, 258–265.

Zolli-Juran,M. *et al.* (2003) HTS identifies novel inhibitors of *Escherichia coli* dihydrofolate reductase that are competitive with dihydrofolate. *Bioorg. Med. Chem. Lett.*, **13**, 2493–2496.

Supplementary Materials

Screening Method

The high throughput screen of *E. coli* dihydrofolate reductase (DHFR) against 50,000 small molecules from ChemBridge Corporation was considered. The screen was performed at the McMaster High Throughput Screening Laboratory in duplicate in 96-well plates using the Beckman-Coulter Integrated Robotic System.

The statistical parameters Z and Z' (Zhang *et al.* 1999) for the screen were 0.57 and 0.72 respectively. These values were comparable to those calculated for the previously reported screen of DHFR against a 50,000 small molecule library from Maybridge plc (Zolli-Juran *et al.* 2003 and Elowe *et al.* 2005).

The following information about the screening procedure can be found in the McMaster HTS laboratory report available on the competition web site: http://hts.mcmaster.ca/Downloads/82BFBEB4-F2A4-4934-B6A8-804CAD8E25A0.html.

Once an assay plate was transferred to the SpectraMax Plus, it was shaken for 5 seconds and each well was read at 340 nm every 15 seconds for 5 minutes (without shaking between reads). All raw data were transferred directly to Activity Base for analysis. Three different controls, High, Low, and Reference, were used in the screen as outlined in Figure 1sm below. For each of these controls, library compounds were excluded from the assay reaction and replaced with: (i) High controls: 2 μL DMSO; (ii) Low controls: 2 μL of 150 μM trimethoprim in DMSO; Reference controls: 2 μL of 1.2 μM trimethoprim in DMSO.

- Enzymatic activity was calculated in Activity Base by the slope of the 7 data points between 20-130 seconds (inclusive) of the 5 minute kinetic read.
- Percent residual activity was calculated using a variant of Formula (2):

$$x_i' = \frac{x_i - L}{H - L} * 100\%,$$

where x_i - measured value at well i, H – mean of high controls, L - mean of low controls, and x_i' - evaluated percentage at well i.

The detailed description of the hit selection procedure can be found in the McMaster procedure report.

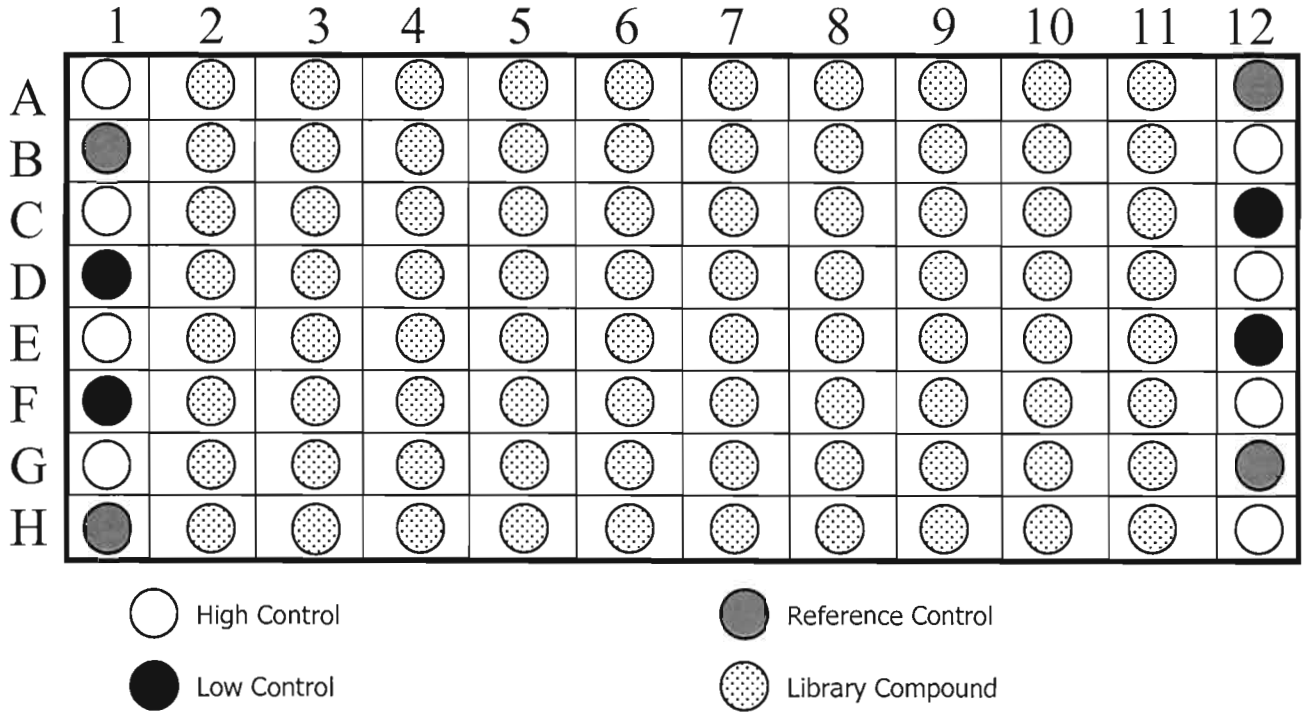


Figure 1sm. Plate layout of the McMaster test assay.

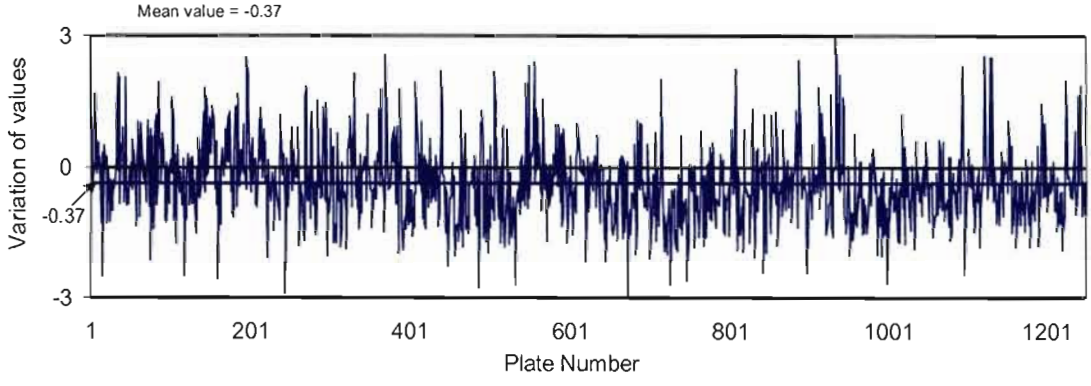


Figure 2sm. Variation of plate normalized values across different plates for the well located in column 1 and row 8 of the McMaster dataset (measured over 1250 plates).

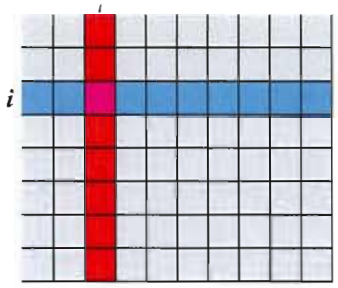
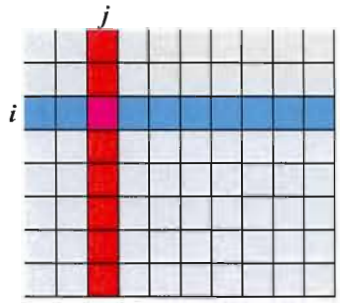
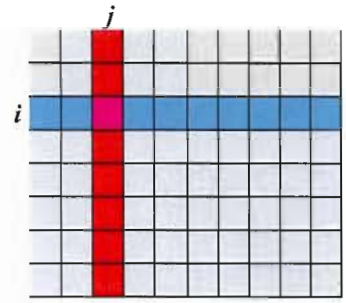
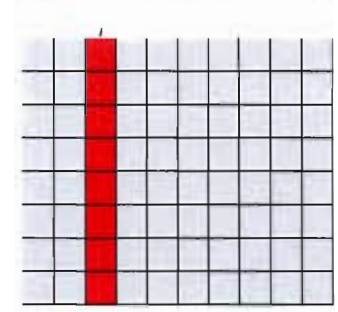
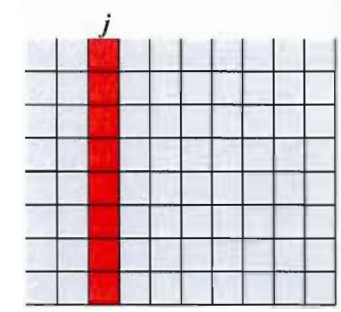
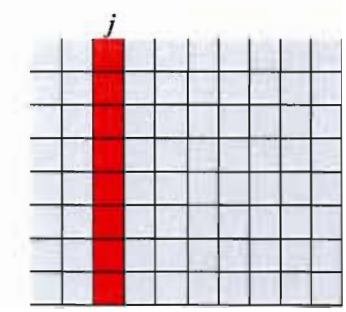
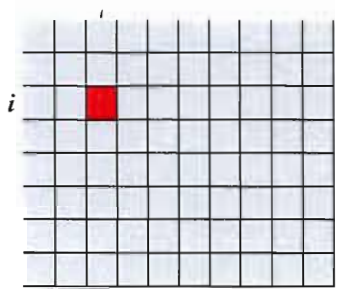
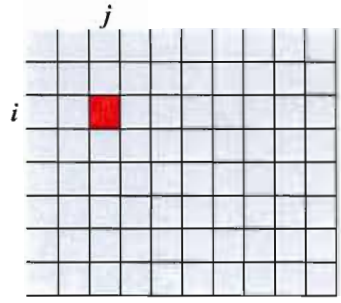
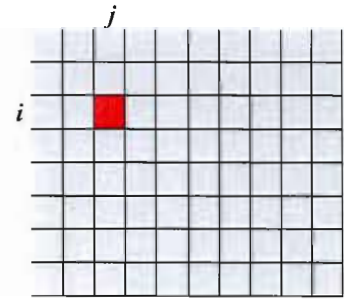
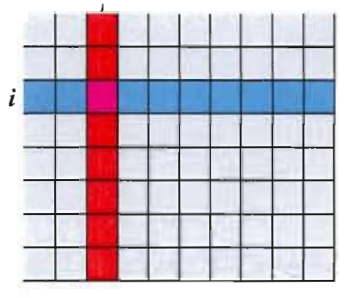
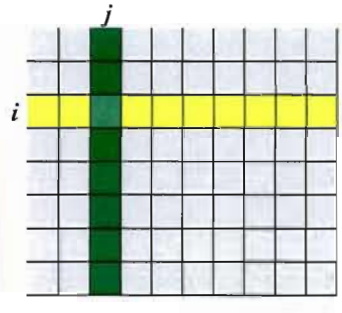
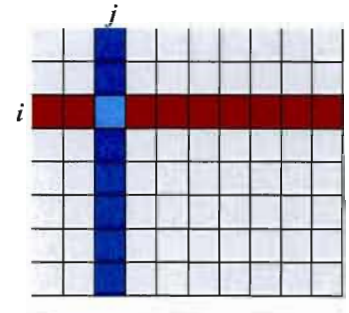
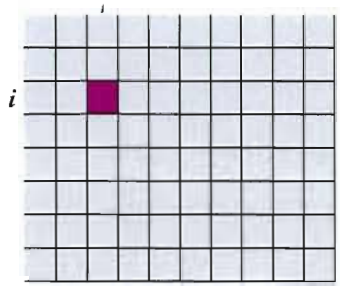
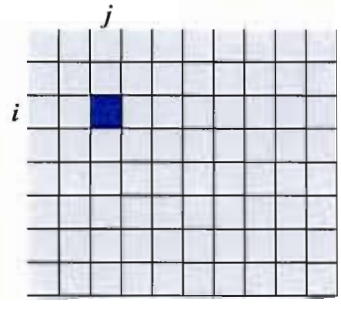
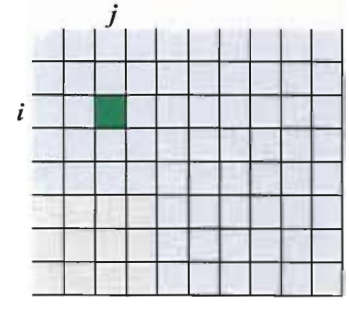
	Plate p	Plate $p+1$	Plate $p+2$
(a - Figure 2) $x'_{ijp} = x_{ijp} + s_i + s_j + rand_{ijp}$, $1 \le i \le 8; 1 \le j \le 10; 1 \le p \le 1250$			
(b - Figure 5sm) $x'_{ijp} = x_{ijp} + s_j + rand_{ijp}$, $1 \le i \le 8; 1 \le j \le 10; 1 \le p \le 1250$			
(c - Figure 6sm) $x'_{ijp} = x_{ijp} + s_{ij} + rand_{ijp}$, $1 \le i \le 8; 1 \le j \le 10; 1 \le p \le 1250$			
(d - Figure 4) $x'_{ijp} = x_{ijp} + s_{ip} + s_{jp} + rand_{ijp}$, $1 \le i \le 8; 1 \le j \le 10; 1 \le p \le 1250$			
(e - Figure 7sm) $x'_{ijp} = x_{ijp} + Rand_{ijp}$, $1 \le i \le 8; 1 \le j \le 10; 1 \le p \le 1250$			

Table 1sm: Schematic diagram of a set of "plates" for each of the 5 manipulations used to simulate systematic (or random) error. (a-d) Colored locations represent typical bias + random error effects; (e) Colored locations represent typical random error effects. Each row and each column (cases a, b and d) as well as each well (cases c and e) of each plate (gray colored locations) were also affected by this kind of systematic and random errors.

Diagram caption (Table 1sm)

(a) Systematic errors stemming from *row x column interactions*: Different constant values were applied to each row and each column of the first plate. The same constants were added to the corresponding rows and columns of all other plates. (b) Systematic error stemming from *column effects*: Different constant values were applied to each column of the first plate. The same constants were added to the corresponding columns of all other plates. (c) Systematic error stemming from *well effects*: Different values were added to each well of the first plate. These values were constant for each well across all plates. (d) Systematic error stemming from *changing row x column interactions*: As in (a) but with the values of the row and column constants varying across plates. (e) Random error only (present in each well of each plate and affecting all wells differently).

The error-perturbed value, x'_{ijp}, of the measurement in row i and column j on the p-th plate was obtained using the formulas indicated in the left, where x_{ijp} is the correct measurement in well ij of plate p, s_i is the systematic error affecting row i, s_j is the systematic error affecting column j, s_{ij} is the systematic error affecting well located on the intersection of line i and column j, s_{ip} is and the systematic error affecting line i of plate p, s_{jp} is the systematic error affecting column j of plate p, $rand_{ijp}$ and $Rand_{ijp}$ are the random error affecting well ij of plate p.

The variables s_i, s_j, s_{ij}, s_{ip} and s_{jp} (see also Equation 5 in the manuscript) had a standard normal distribution with parameters $N(0, c)$, where c equals 0, $0.6SD$, $1.2SD$, $1.8SD$, $2.4SD$, or $3SD$ for the various simulation conditions and SD is the standard deviation of the variable x_{ijp} distributed according to a standard normal distribution. For all values of the parameter c, the random error *rand* (cases a to d) was always distributed according to a standard normal distribution with parameters $N(0, 0.3SD)$. The random error *Rand* (case e) was distributed according to a standard normal distribution with parameters $N(0, 1.2SD)$.

Table 2sm (see also Figure 2 in the manuscript). True, false positive and false negative hit detection rates for the 5 methods used to process the random *standard normal data* with 1% of added hits. The hit detection rates were obtained by dividing the number of the true (or false positive, or false negative) hits by the total number of generated hits.

Methods \ Systematic error		0	0.6*SD*	1.2*SD*	1.8*SD*	2.4*SD*	3.0*SD*
Hit detection rate (in %)	1. $\overline{x} - 3SD$ per plate	88.65	86.37	79.12	66.52	51.90	38.74
	2. $\overline{x} - 3SD$ per assay	94.21	92.47	86.41	74.78	60.37	46.61
	3. Median polish	89.24	89.16	89.21	88.98	88.94	88.75
	4. B score	83.34	83.26	83.25	83.03	82.90	82.69
	5. Well correction	93.91	93.91	93.95	93.83	93.78	93.66
False positive rate (in %)	1. $\overline{x} - 3SD$ per plate	5.41	5.34	5.21	4.71	4.02	3.30
	2. $\overline{x} - 3SD$ per assay	2.79	3.39	3.96	4.04	3.83	3.36
	3. Median polish	14.30	14.31	14.33	14.30	14.29	14.27
	4. B score	25.38	25.39	25.49	25.57	25.53	25.57
	5. Well correction	3.13	3.12	3.16	3.19	3.20	3.25
False negative rate (in %)	1. $\overline{x} - 3SD$ per plate	11.35	13.63	20.88	33.48	48.09	61.26
	2. $\overline{x} - 3SD$ per assay	5.79	7.53	13.59	25.22	39.63	53.39
	3. Median polish	10.75	10.84	10.79	11.02	11.06	11.25
	4. B score	16.66	16.74	16.75	16.97	17.10	17.31
	5. Well correction	6.09	6.09	6.04	6.17	6.22	6.34

Table 3sm (see also Figure 3sm). True, false positive and false negative hit detection rates for the 5 methods used to process the random *heavy tailed data* with 1% of added hits. The hit detection rates were obtained by dividing the number of true (or false positive, or false negative) hits by the total number of generated hits.

Methods \ Systematic error		0	0.6*SD*	1.2*SD*	1.8*SD*	2.4*SD*	3.0*SD*
Hit detection rate (in %)	1. $\bar{x} - 3SD$ per plate	93.24	91.84	87.33	79.49	69.15	58.82
	2. $\bar{x} - 3SD$ per assay	94.67	93.45	89.37	82.02	72.00	61.89
	3. Median polish	87.93	87.91	87.80	87.77	87.64	87.50
	4. B score	84.66	84.61	84.57	84.47	84.37	84.18
	5. Well correction	94.57	94.56	94.50	94.52	94.44	94.30
False positive rate (in %)	1. $\bar{x} - 3SD$ per plate	35.84	60.93	95.94	120.02	131.36	136.64
	2. $\bar{x} - 3SD$ per assay	0.00	36.72	86.55	117.76	132.83	140.02
	3. Median polish	176.20	176.07	176.36	177.12	178.10	178.62
	4. B score	189.31	189.19	189.12	189.81	190.79	191.07
	5. Well correction	3.25	3.47	4.11	5.34	7.23	9.71
False negative rate (in %)	1. $\bar{x} - 3SD$ per plate	6.76	8.16	12.67	20.51	30.85	41.18
	2. $\bar{x} - 3SD$ per assay	5.33	6.55	10.63	17.98	28.00	38.11
	3. Median polish	12.07	12.09	12.20	12.23	12.36	12.49
	4. B score	15.34	15.39	15.43	15.53	15.63	15.82
	5. Well correction	5.43	5.44	5.50	5.48	5.56	5.70

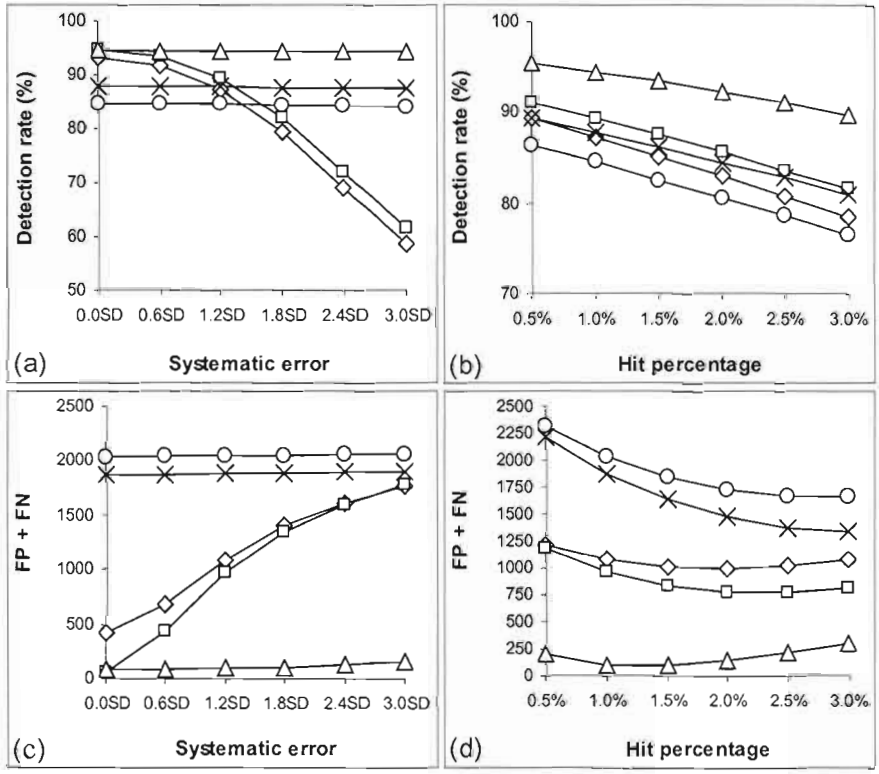


Figure 3sm (see also Table 3sm). True hit rate and total of the number of false positive and false negative hits for the noisy *heavy tailed data* obtained with the methods using plates' parameters (i.e., Method 1, ◊), assay parameters (i.e., Method 2, ❑), median polish (x), B score method (○), and the well correction procedure (Δ). The abscissa axis indicates the noise factor (a and c - with fixed hit percentage of 1%) and the percentage of added hits (b and d - with fixed error rate of 1.2*SD*).

Table 4sm (see also Figure 4sm). True, false positive, and false negative hit detection rates for the 5 methods used to process the random *light tailed data* with 1% of added hits. The hit detection rates were obtained by dividing the number of true (or false positive, or false negative) hits by the total number of generated hits.

Methods \ Systematic error		0	0.6*SD*	1.2*SD*	1.8*SD*	2.4*SD*	3.0*SD*
Hit detection rate (in %)	1. $\bar{x} - 3SD$ per plate	85.24	82.63	74.01	59.91	43.77	30.45
	2. $\bar{x} - 3SD$ per assay	94.50	92.57	85.50	72.18	55.65	40.86
	3. Median polish	90.78	90.77	90.69	90.59	90.43	90.25
	4. B score	82.88	82.90	82.81	82.67	82.50	82.33
	5. Well correction	94.04	94.01	93.96	93.91	93.78	93.62
False positive rate (in %)	1. $\bar{x} - 3SD$ per plate	3.89	3.53	2.83	2.01	1.29	0.85
	2. $\bar{x} - 3SD$ per assay	0.00	0.24	1.13	1.34	1.17	0.89
	3. Median polish	7.11	7.08	7.18	7.17	7.18	7.19
	4. B score	14.46	14.46	14.49	14.37	14.26	14.24
	5. Well correction	0.12	0.13	0.14	0.14	0.17	0.20
False negative rate (in %)	1. $\bar{x} - 3SD$ per plate	14.76	17.38	25.99	40.09	56.23	69.55
	2. $\bar{x} - 3SD$ per assay	5.50	7.43	14.50	27.82	44.35	59.14
	3. Median polish	9.22	9.23	9.31	9.40	9.57	9.75
	4. B score	17.12	17.10	17.19	17.33	17.50	17.67
	5. Well correction	5.96	5.99	6.04	6.09	6.22	6.38

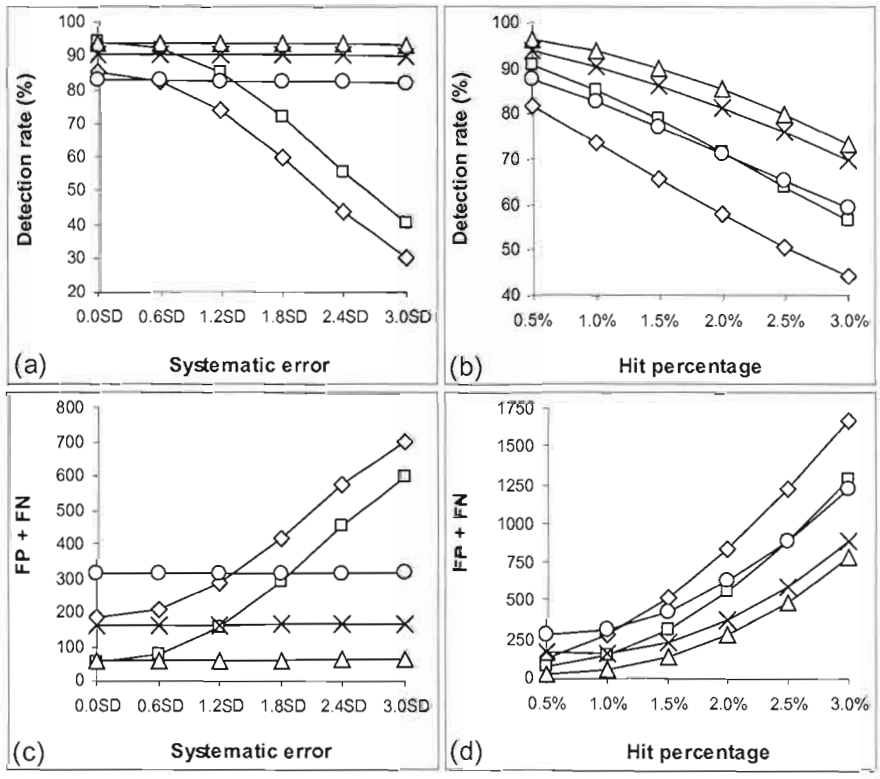


Figure 4sm (see also Table 4sm). True hit rate and total of the number of false positive and false negative hits for the noisy *light tailed data* obtained with the methods using plates' parameters (i.e., Method 1, ◊), assay parameters (i.e., Method 2, ❑), median polish (x), B score method (○), and the well correction procedure (Δ). The abscissa axis indicates the noise factor (a and c - with fixed hit percentage of 1%) and the percentage of added hits (b and d - with fixed error rate of 1.2*SD*).

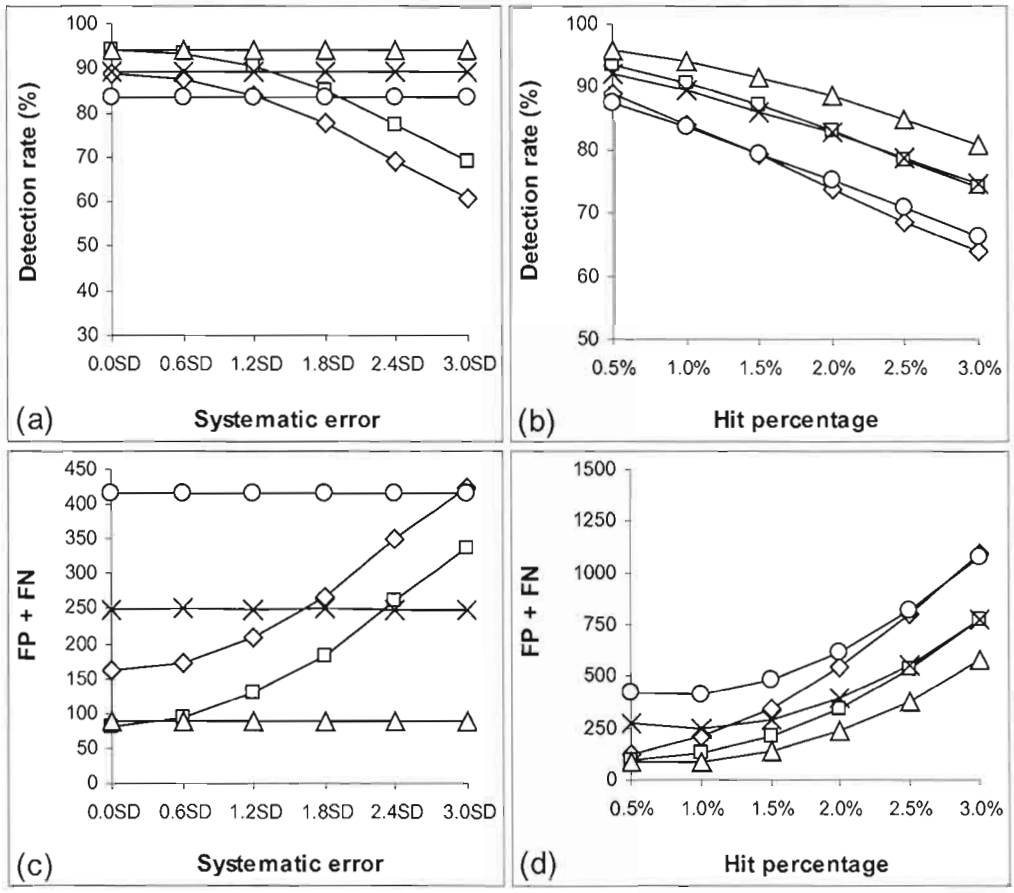


Figure 5sm. True hit rate and total of the number of false positive and false negative hits for the noisy *standard normal data with systematic error stemming from column effects only*. The results were obtained with the methods using plates' parameters (i.e., Method 1, ◊), assay parameters (i.e., Method 2, ❑), median polish (x), B score method (○), and the well correction procedure (Δ).The abscissa axis indicates the noise factor (a and c - with fixed hit percentage of 1%) and the percentage of added hits (b and d - with fixed error rate of 1.2*SD*).

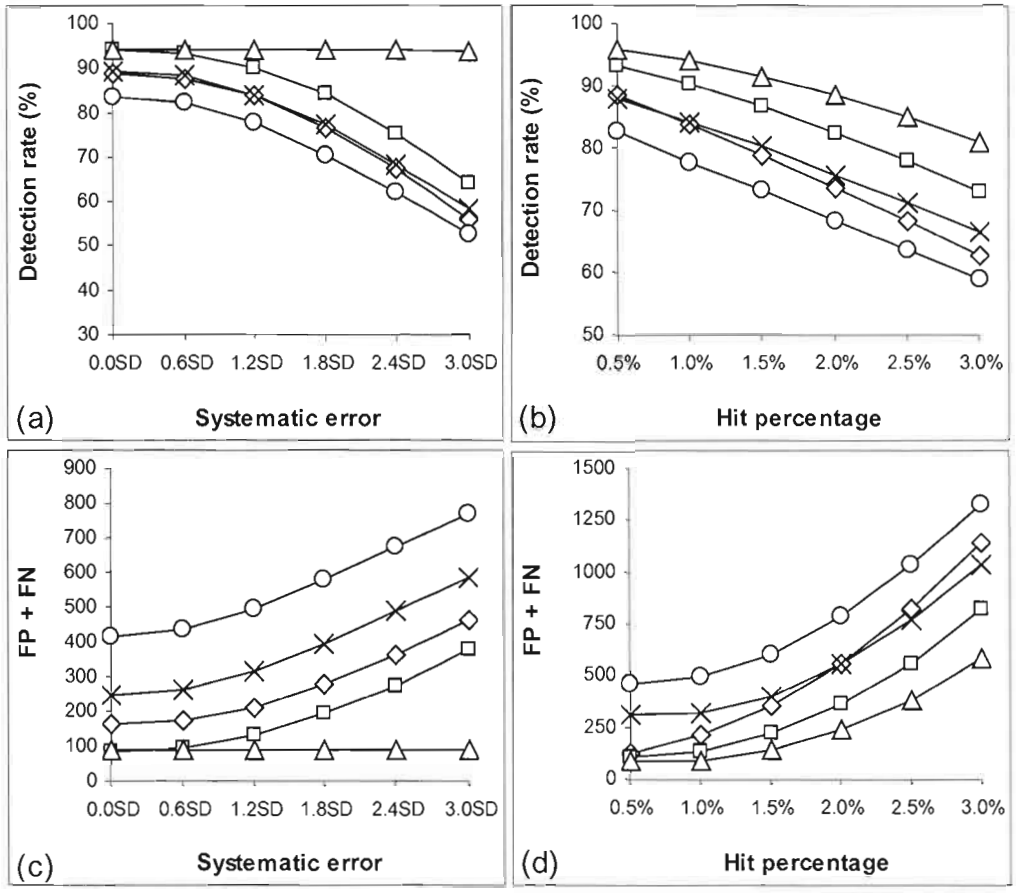


Figure 6sm. True hit rate and total of the number of false positive and false negative hits for the noisy *standard normal data with systematic error different for all wells (no row x column interactions was involved)*. The results were obtained with the methods using plates' parameters (i.e., Method 1, ◊), assay parameters (i.e., Method 2, ❑), median polish (x), B score method (○), and the well correction procedure (Δ). The abscissa axis indicates the noise factor (a and c - with fixed hit percentage of 1%) and the percentage of added hits (b and d - with fixed error rate of 1.2*SD*).

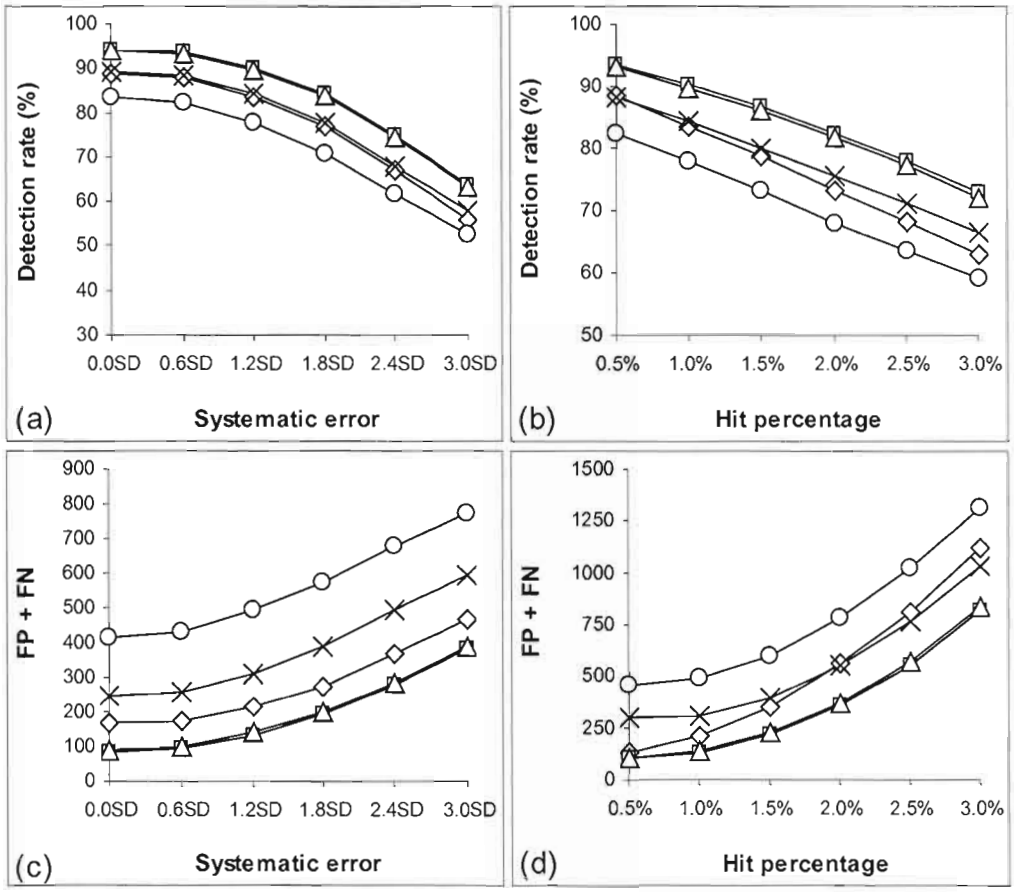


Figure 7sm. True hit rate and total of the number of false positive and false negative hits for the noisy *standard normal data with random error only stemming and varying from plate to plate*. The results were obtained with the methods using plates' parameters (i.e., Method 1, ◊), assay parameters (i.e., Method 2, ❑), median polish (x), B score method (○), and the well correction procedure (Δ). The abscissa axis indicates the noise factor (a and c - with fixed hit percentage of 1%) and the percentage of added hits (b and d - with fixed error rate of 1.2*SD*).

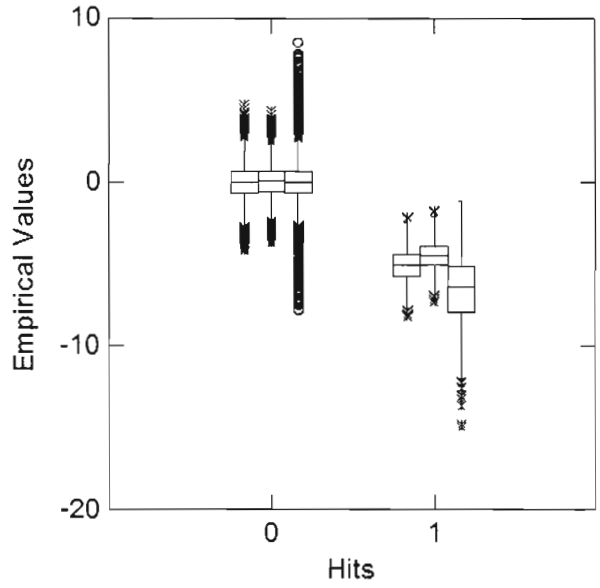


Figure 8sm. Box plots for null (Hits = 0) and "standard normal + 1% hits" > (Hits = 1) data. From left to right, empirical values are for raw (Method 2), well-corrected, and B-score data.

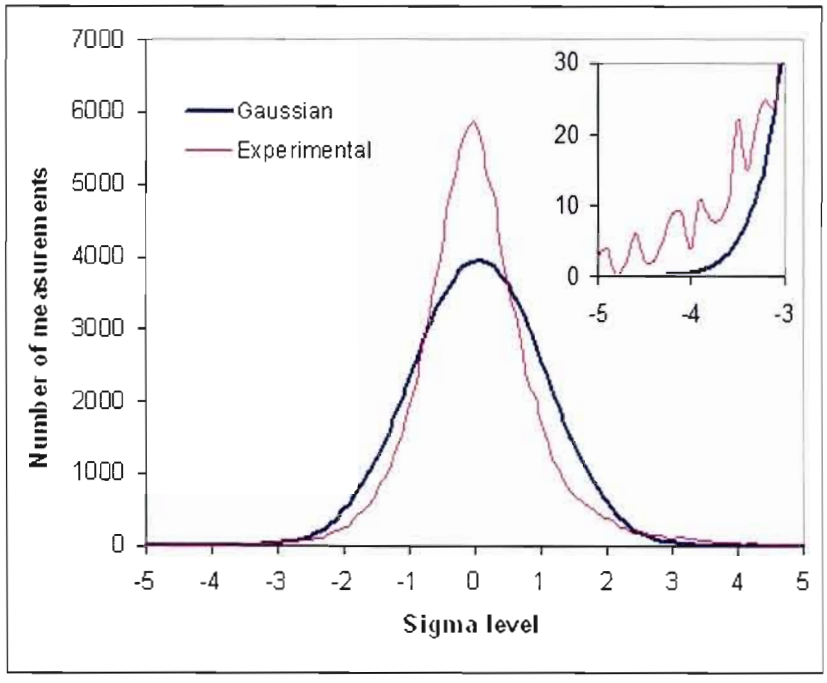


Figure 9sm. Distribution of measurements in the McMaster assay (1250 plates) and its comparison to a Gaussian distribution.

Table 5sm. Results of the χ-square contingency tests carried out for the raw and well-corrected (Well Cor.) McMaster datasets with the hit selection thresholds: $\bar{x} - SD$, $\bar{x} - 1.5SD$, and $\bar{x} - 2SD$.

$\alpha = 0.01$	$\bar{x} - SD$		$\bar{x} - 1.5SD$		$\bar{x} - 2SD$	
	Raw	Well Cor.	Raw	Well Cor.	Raw	Well Cor.
Mean number of hits per well	137.7	134.3	49.8	46.8	18.4	16.9
χ-square value	2377.4	204.6	1258.4	173.6	438.6	74.8
χ-square critical value	111.1	111.1	111.1	111.1	111.1	111.1
χ-square contingency hypothesis H_0	No	No	No	No	No	Yes
Figure 6	a	b	c	d	e	f

Table 6sm. Results of the χ-square contingency tests carried out for the raw and well-corrected (Well Cor.) McMaster datasets with the hit selection thresholds: $\bar{x} - 2.5SD$, $\bar{x} - 3SD$, and $\bar{x} - 3.5SD$.

$\alpha = 0.01$	$\bar{x} - 2.5SD$		$\bar{x} - 3SD$		$\bar{x} - 3.5SD$	
	Raw	Well Cor.	Raw	Well Cor.	Raw	Well Cor.
Mean number of hits per well	7.3	7.1	3.2	3.1	1.5	1.5
χ-square value	172.0	86.6	129.0	110.7	106.2	105.3
χ-square critical value	111.1	111.1	111.1	111.1	111.1	111.1
χ-square contingency hypothesis H_0	No	Yes	No	Yes	Yes	Yes
Figure 10sm	a	b	c	d	e	f

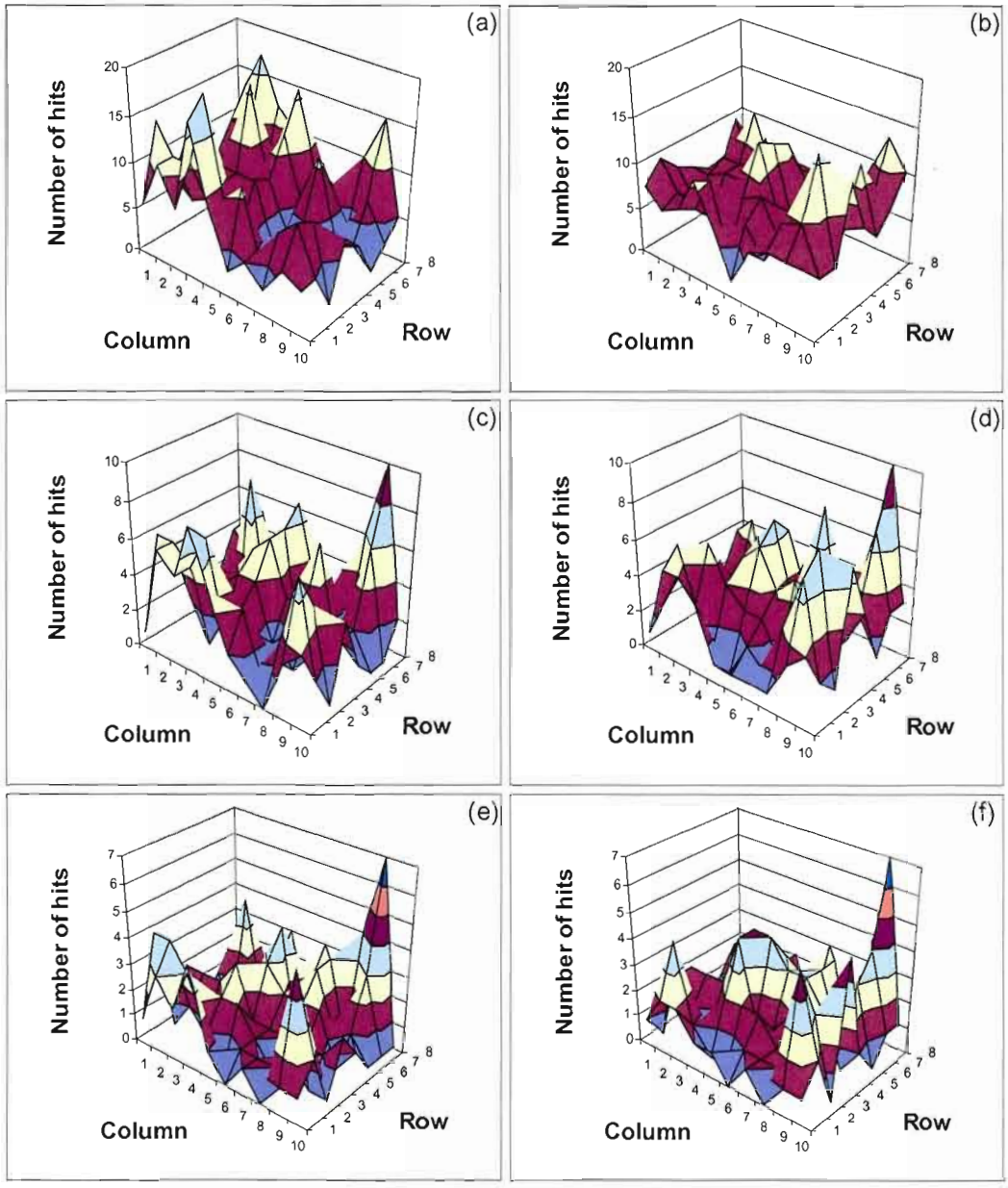


Figure 10sm. Hit distributions for the raw (a, c, and e) and well-corrected (b, d, and f) McMaster datasets obtained with the hit selection thresholds $\bar{x} - 2.5SD$ (a and b), $\bar{x} - 3SD$ (c and d), and $\bar{x} - 3.5SD$ (e and f).

Table 7sm. Hit distribution of the raw McMaster dataset computed for the $\bar{x} - 3SD$ threshold (mean value of hits per well is 3.18 and standard deviation is 2.28).

Row\Column	1	2	3	4	5	6	7	8	9	10
1	1	6	5	8	5	2	1	0	3	4
2	6	6	5	7	2	4	1	3	7	2
3	5	2	0	2	3	2	1	1	5	0
4	5	4	3	4	4	1	2	1	2	4
5	3	0	1	5	6	3	3	2	2	1
6	2	2	8	1	6	0	6	1	2	0
7	4	5	1	3	7	3	3	1	6	0
8	3	5	1	6	1	3	1	6	10	2

Table 8sm. Hit distribution of the well-corrected McMaster dataset computed for the $\bar{x} - 3SD$ threshold (mean value of hits per well is 3.10 and standard deviation is 2.08).

Row\Column	1	2	3	4	5	6	7	8	9	10
1	1	3	5	4	1	1	1	1	3	4
2	4	4	3	1	0	2	1	4	7	2
3	5	2	0	4	3	3	1	2	8	1
4	4	5	3	5	6	4	5	1	2	7
5	3	0	2	5	7	3	4	2	2	4
6	2	2	4	2	6	3	8	1	3	1
7	4	5	0	3	2	2	3	2	7	3
8	3	1	0	3	1	3	1	5	10	3

Table 9sm. Consensus hits (i.e. hits identified in both copies) obtained by both Method 1 and Method 5 (with hit selection carried out on the plate-by-plate basis). Method 1 identified as hits all compounds with the consensus residual score <=75%. Method 5 identified as hits all compounds with the consensus residual score lower than $\bar{x} - 2.29SD$ (this threshold corresponds to the 75% residual score used in Method 1).

McMaster samples identified as hits in both copies by Methods 1 and 5	
MAC-0103980	MAC-0115794
MAC-0104038	MAC-0116655
MAC-0104867	MAC-0117820
MAC-0107329	MAC-0119733
MAC-0108994	MAC-0122586
MAC-0109949	MAC-0122661
MAC-0110019	MAC-0122959
MAC-0110027	MAC-0127264
MAC-0110039	MAC-0130938
MAC-0112108	MAC-0131221
MAC-0112179	MAC-0136174
MAC-0112287	MAC-0139408
MAC-0112764	MAC-0140910
MAC-0114159	MAC-0144586
MAC-0114615	MAC-0145030
MAC-0114842	
McMaster samples identified as hits in both copies only by Method 1 (consensus of 75%) and not identified by Method 5	
MAC-0110562	
MAC-0115469	
MAC-0117240	
MAC-0128921	
MAC-0130772	
MAC-0132669	
MAC-0133856	
MAC-0140989	
MAC-0145361	
MAC-0149343	
MAC-0150029	

CHAPITRE V

LOGICIEL « HTS CORRECTOR » : PRÉSENTATION ET INTRODUCTION

Note : dans ce chapitre, pour alléger la présentation, nous avons numéroté seulement les figures montrant un résultat ou présentant un intérêt particulier.

5.1 Présentation

Le projet « HTS Corrector » a été initialisé par Dymtro Kevorkov et Vladimir Makarenkov en réponse aux problèmes rencontrés lors des criblages HTS. Au départ, leurs études se sont concentrées sur le développement de nouveaux algorithmes d'analyse, de correction et de recherche des « hits », s'appuyant sur des méthodes statistiques. L'utilité d'une solution informatique, intégrant les algorithmes développés, s'est vite imposée par le grand nombre de données produites lors d'une campagne de criblage HTS. L'analyse aidée par ordinateur, fournit un outil de décision précieux et peut contribuer à minimiser les erreurs dues aux manipulations des données.

Pour pouvoir partager facilement leurs résultats avec la communauté scientifique internationale, Kevorkov et Makarenkov ont entrepris le développement d'un logiciel intégrant les algorithmes qu'ils avaient développés. Leurs efforts ont donné naissance aux premières versions de « HTS Correcteur ». Ces premières

versions étaient développées en C++ et sont téléchargeables gratuitement (voir aussi l'article 2 au chapitre 4).

L'accueil des méthodes proposées a été très favorable, notamment lors de la conférence SBS 2005 à Genève, où l'affiche présentant les méthodes, a gagné le prix de la meilleure affiche parmi plus de 300 concurrents. Suite à ces encouragements, le professeur Makarenkov a décidé de poursuivre le développement du logiciel, de mieux l'adapter aux praticiens et d'en améliorer la conception, notamment en vue de faciliter sa mise à jour et sa maintenance. Il m'a délégué cette tâche.

Malgré le travail incroyable fourni par Dymtro Kevorkov, chimiste de formation, les premières versions soufraient d'un manque évident de connaissances en génie logiciel et des notions de base de programmation structurée. Le programme se présentait comme un agglomérat de modules indépendants regroupés par une interface graphique commune. La répétition de grandes portions de code était la norme. Le découpage fonctionnel était très faible et le couplage très fort. La plupart du code, des divers modules, se trouvait dans d'interminables procédures principales. Cela impliquait beaucoup de problèmes lors des mises à jour et lors du débogage.

Il m'a semblé primordial d'utiliser des méthodes de programmation plus robustes, pour faciliter l'intégration de nouvelles fonctionnalités et l'ajout des nouveaux algorithmes qui étaient en cours de développement. Le choix d'un langage de programmation objet de dernière génération, me semblait aussi important. Après avoir hésité entre Java et C#, mon choix s'est porté sur ce dernier. La facilité de programmation des interfaces graphiques et la rapidité d'exécution du code l'ont emporté sur la portabilité. Ce dernier point n'était pas un obstacle fondamental, vu que la plupart des utilisateurs ont accès à des machines Windows. De plus, il existe une version inter-plateforme du langage, développée par la communauté Open Source. Pour le moment nous n'avons pas testé cette solution.

Il me semblait nécessaire, également, de confronter le logiciel aux utilisateurs finaux, pour en améliorer l'interface et pour tirer profit de leur expertise et de leurs remarques. La mise en situation réelle me semble la meilleure approche pour trouver de nouvelles pistes de solutions et pour mieux répondre aux besoins des utilisateurs. Une première étape a été franchie avec les études menées en collaboration avec Philippe Garneau lors d'un travail dans le cadre du cours BIF 7001 (voir le rapport par Garneau et Zentilli, 2006, non publié). L'expertise de praticien, non informaticien, de M. Garneau, a permis quelques améliorations et a mis en évidence des problématiques auxquelles nous n'avions pas pensé. Il reste, néanmoins, que le logiciel doit encore subir des tests poussés pour contrôler son adaptation aux besoins des utilisateurs finaux.

La version actuelle (décembre 2006) est disponible gratuitement sur le site : http://www.info2.uqam.ca/~makarenv/HTS/home.php.

5.2 Guide d'introduction

Nous présentons dans ce paragraphe un bref aperçu des fonctionnalités principales du logiciel « HTS Corrector » dans sa version de décembre 2006. Ces informations suivent la présentation qui en est faite dans le site de présentation et de téléchargement que nous avons crée. On peut s'y connecter à l'adresse suivante : http://www.info2.uqam.ca/~makarenv/HTS/home.php.

Bienvenue à « HTS Corrector »

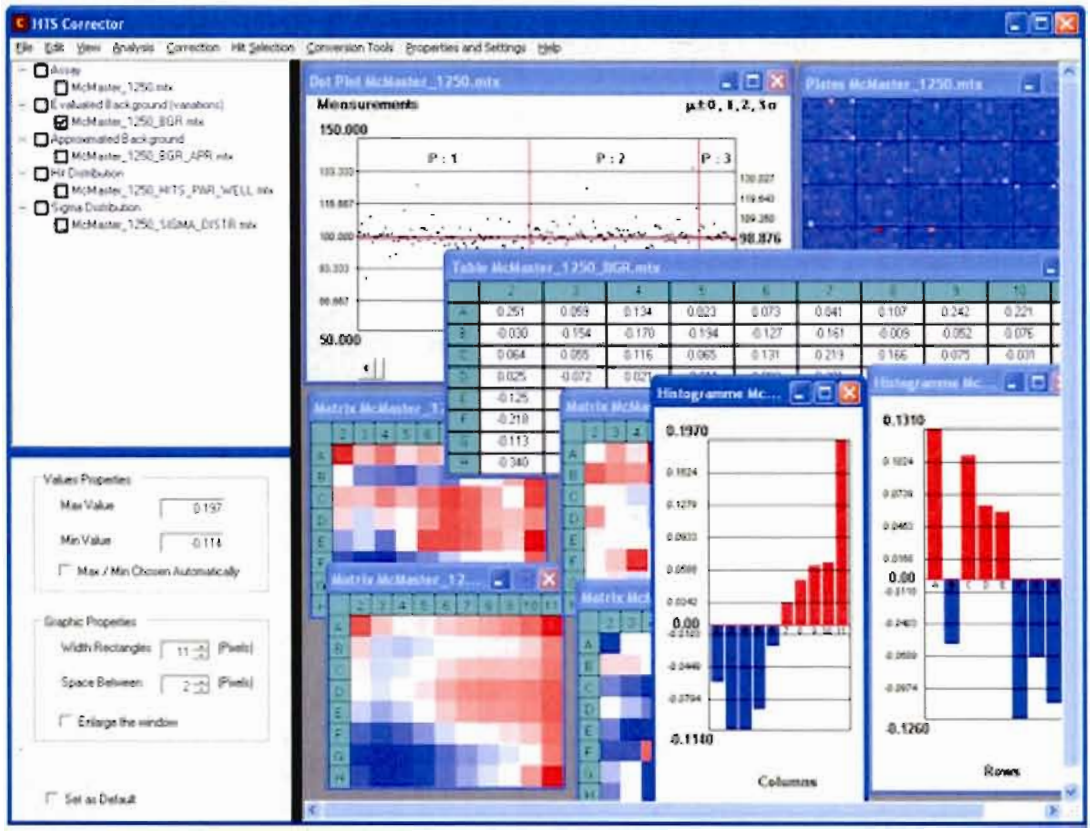


Figure 5.1 : Vue d'ensemble de « HTS Corrector ».

Développé au département d'Informatique de l'Université du Québec à Montréal.

Auteurs : Pablo Zentilli, Vladimir Makarenkov, Dmytro Kevorkov, Andrei Gagarin, Martin Lacroix et Rémi Lavoie.

Le logiciel « HTS Corrector » a été conçu pour visualiser, analyser et corriger les données provenant de campagnes de criblage à haut débit (HTS).

Téléchargements :

- Le logiciel : HTS_Corrector.zip;
- Fichiers d'exemples : McMaster_1250.zip, Harvard_164.zip, Harvard_54.zip.

Références :

Kevorkov D. et Makarenkov V. « Statistical analysis of systematic errors in high-throughput screening. », *Journal of Biomolecular Screening*, **10**, 557-567, 2005.

Kevorkov D. and Makarenkov V. « Quality control and data correction in high-throughput screening. », *Proceedings of SFC 2005*, pp. 159-163, Montréal, Canada.

Makarenkov V., Kevorkov D., Zentilli P., Gagarin A., Malo N. and Nadon R. « HTS-Corrector: software for the statistical analysis and correction of experimental high-throughput screening data. », *Bioinformatics*, 2006, **22**, 1408 - 1409.

Andrei Gagarin, Vladimir Makarenkov, and Pablo Zentilli. « Using Clustering Techniques to Improve Hit Selection in High-Throughput Screening. », *Journal of Biomolecular Screening*, **11**, 903 - 914.

A. Gagarin, D. Kevorkov, V. Makarenkov, and P. Zentilli. « Comparison of two methods for detecting and correcting systematic error in high-throughput screening data. », *Data Science and Classification*, V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Ziberna (Eds.), 2006, Series: *Studies in Classification, Data Analysis, and Knowledge Organization*, Springer, pp. 241-249.

Résumé :

Le criblage à haut débit (« High-throughput screening » ou HTS) est une première étape dans la découverte de médicaments. Le criblage HTS permet de tester des milliers de composés chimiques en une seule étude. L'objectif principal de notre travail était de développer et d'implémenter des méthodes pour minimiser l'impact des erreurs systématiques lors d'une campagne HTS. « HTS Corrector » est un logiciel permettant d'analyser les données brutes provenant d'un criblage HTS, de

détecter, de visualiser et de corriger les erreurs systématiques, de même que de sélectionner les composés prometteurs (« hits »). Les méthodes statistiques, nouvellement développées par notre équipe, intégrées à « HTS Corrector » sont les suivantes : évaluation et correction de l'arrière-plan (« background correction »), correction par puits (« well correction ») et sélection des « hits » par des méthodes de groupement (*i.e.*, « clustering »).

Principales options du logiciel (tel que présenté sur le site web) :

- Home
- Features
  - **File Formats**
  - **Data Analysis**
    - Background Evaluation
    - Background Approximation
    - Hit Distribution
    - Sigma Distribution
    - Chi-Square Test
    - K-means Partitionning
  - **Data Correction**
    - Well Correction
    - Remove Background
    - Median Polish
    - B Score
  - **Hit Selection**
    - Sigma Selection
    - Clustering Methods
  - **Data Visualisation**
  - **Properties & Settings**
  - **Others Features**
    - Excel to HTS Conversion
    - Hit List
    - Old to new file format
- Downloads
- Bugs & Requests
- Contact
- Team
- References

Principales fonctionnalités du logiciel :

- Identification et analyse statistique des variations systématiques du signal lors de criblages HTS;
- Visualisation des données provenant de criblages HTS;
- Séparation des données de la campagne HTS en groupes homogènes par un groupement utilisant l'algorithme *k-means*;
- Évaluation de la surface d'arrière-plan qui reflète l'impact des erreurs systématiques sur les données expérimentales;
- Calcul de la surface approximée pour une surface d'évaluation de l'arrière-plan;
- Correction automatique du jeu de données expérimentales;
- Sélection des « hits » selon différentes méthodes;
- Analyse des données selon le test de contingence χ^2 (pour la distribution des « hits »).

Formats de fichiers utilisés :

- *Format 1* (ancien format) : pour les données d'une campagne HTS. Extension de fichier : *.mtr*.
  - La première ligne du fichier indique le nombre de plateaux, de lignes et de colonnes du criblage HTS;
  - Chaque colonne correspond aux données d'un plateau;
  - Dans la figure 5.2 ci-dessous R_i représente la valeur de la ligne i et C_j les valeurs de la colonne j.

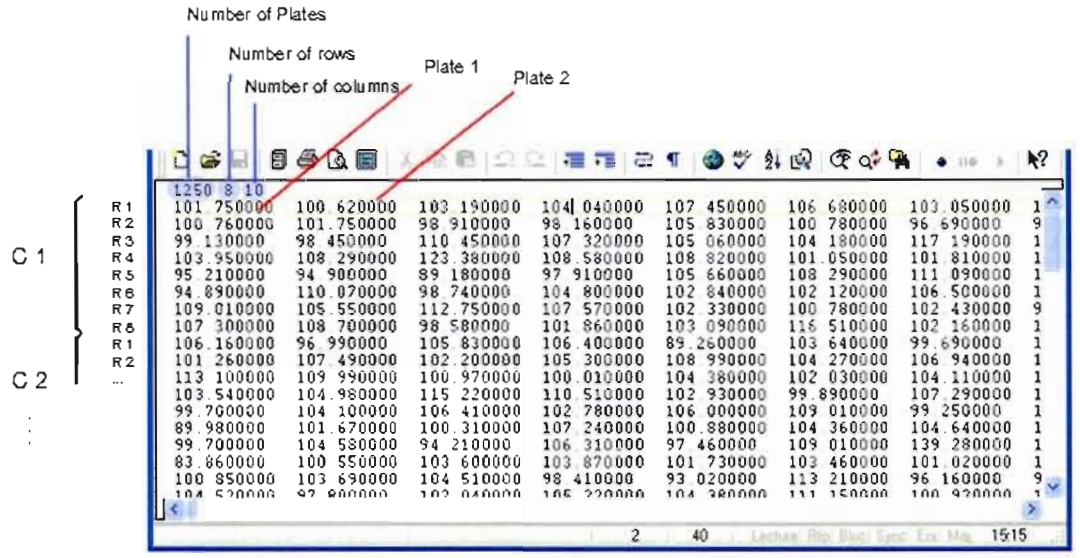


Figure 5.2 : Ancien format de fichier pour une campagne HTS. Extension : *.mtr*.

- *Format 1* (ancien format) : pour les données d'un plateau. Extension des fichiers : *.bgr*, *.apr*, *.hit*, *.sgm*.
  - La première ligne du fichier indique le nombre de lignes et de colonnes du plateau;
  - Ce format correspond au format d'un plateau réel.

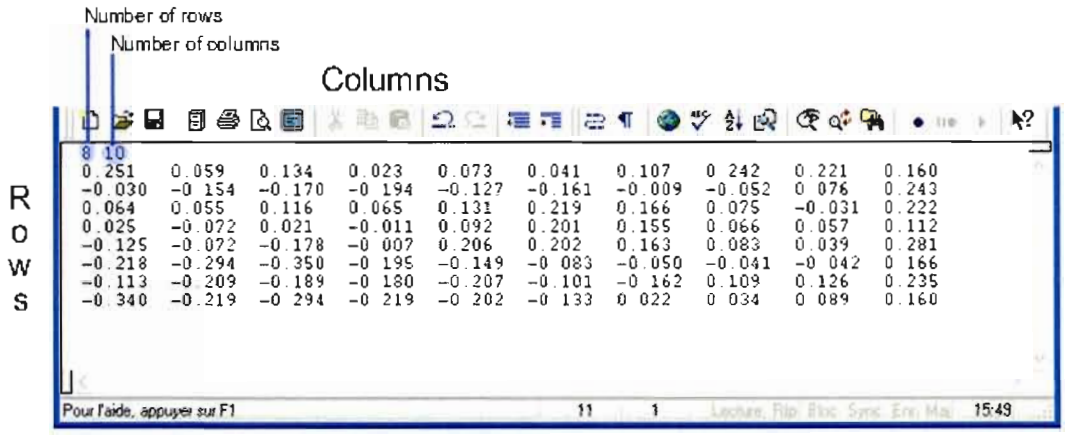


Figure 5.3 : Ancien format de fichier pour les données des plateaux. Extensions : *.bgr*, *.apr*, *.hit*, *.sgm*.

- *Format 2* (nouveau format) : pour les données d'une campagne HTS. Extension de fichier : *.mtx*.

  - Ce format se présente comme un format XML (sans être strictement conforme à la norme XML);
  - La première ligne du fichier indique le type de fichier, le nombre de plateaux, de lignes et de colonnes du criblage HTS;
  - Le type de fichier peut être « ASSAY » ou « HITS »;
  - Chaque plateau doit avoir un nom. Le numéro de plateau est facultatif;
  - Ce format correspond au format d'un plateau réel;
  - Il faut fermer chaque balise avec ***</nomBalise>***.

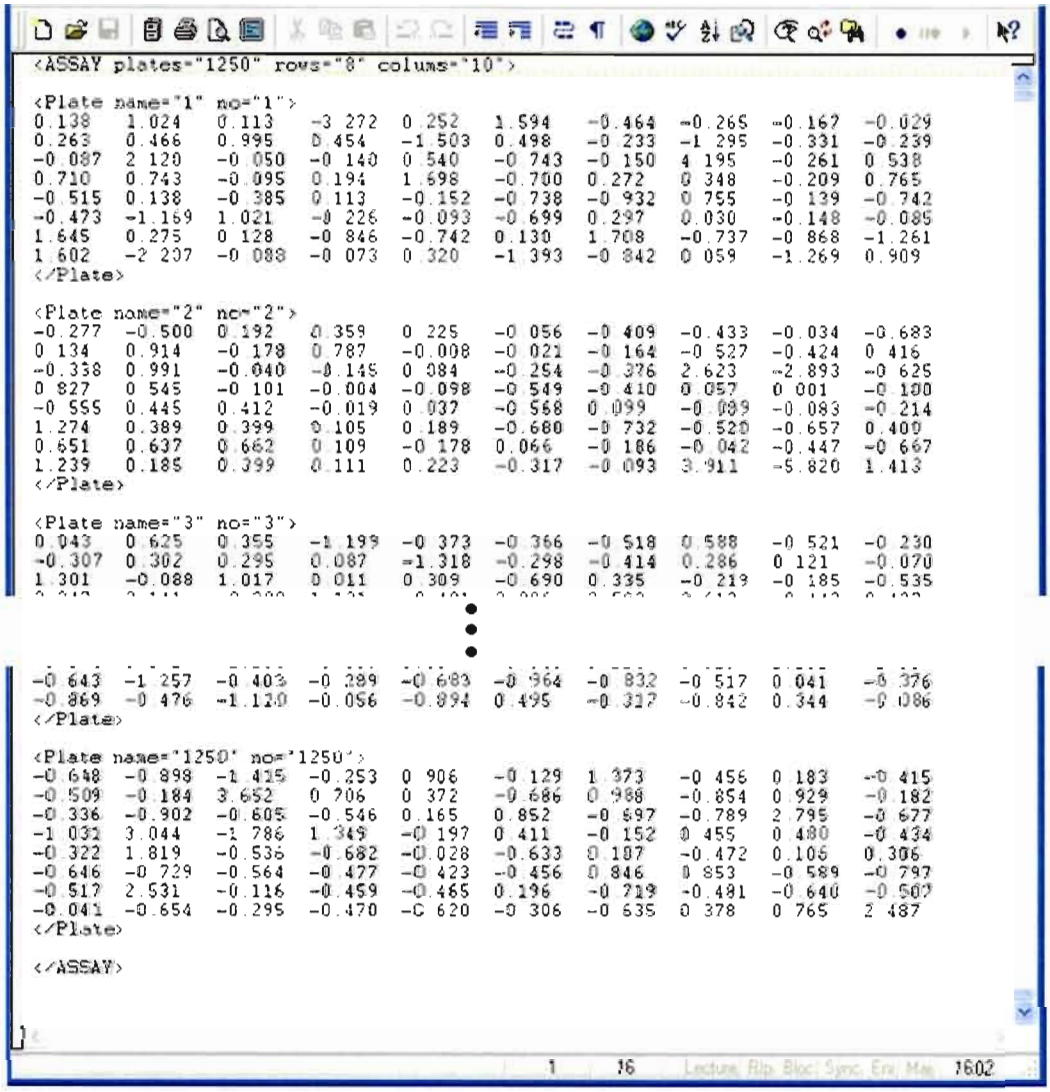


Figure 5.4 : Nouveau format de fichier pour une campagne HTS. Extension : *.mtx*.

- *Format 2* (nouveau format) : pour les données d'un plateau. Extension de fichier : *.mtx*.
  - Ce format se présente comme un format XML (sans être strictement conforme à la norme XML);
  - La première ligne du fichier indique le type de fichier, le nombre de lignes et de colonnes du plateau;

- Le type de fichier peut être : « BACKGROUND », « APPROXIMATION », « HIT_DISTRIBUTION » ou « SIGMA_DISTRIBUTION »;
- Chaque plateau doit avoir un nom (l'attribut doit être spécifié mais peut être vide). Le numéro de plateau est facultatif;
- Ce format correspond au format d'un plateau réel;
- Il faut fermer chaque balise avec ***</nomBalise>***.

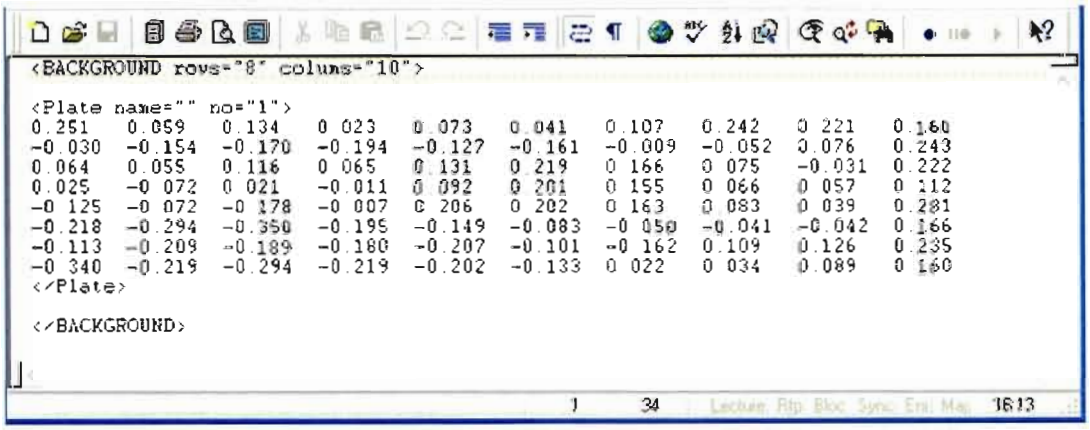


Figure 5.5 : Nouveau format de fichier pour les données des plateaux. Extension : *.mtx*.

- *Format Excel* : pour l'importation de données externes.
  - Vous pouvez utiliser le format Excel de votre choix, mais les valeurs analysées doivent être numériques;
  - Complétez les puits vides avec la moyenne du plateau. Assurez-vous que toutes les feuilles Excel ont les données des plateaux dans la même zone (dans la figure ci-dessous, la zone sélectionnée est : 46 – C à 53 – L);
  - Enlevez toutes les feuilles vides ou incorrectes de votre fichier Excel;
  - « HTS Corrector » procédera par un parcours sur l'ensemble du fichier Excel et prendra, pour valeurs des plateaux, les valeurs correspondant à la zone sélectionnée. Les noms des feuilles deviendront les noms des plateaux;
  - Les valeurs des zones non sélectionnées n'influencent pas le résultat de l'opération. Vous pouvez y inscrire toute information que vous semble pertinente, mais elle ne sera pas prise en compte par « HTS Corrector ».

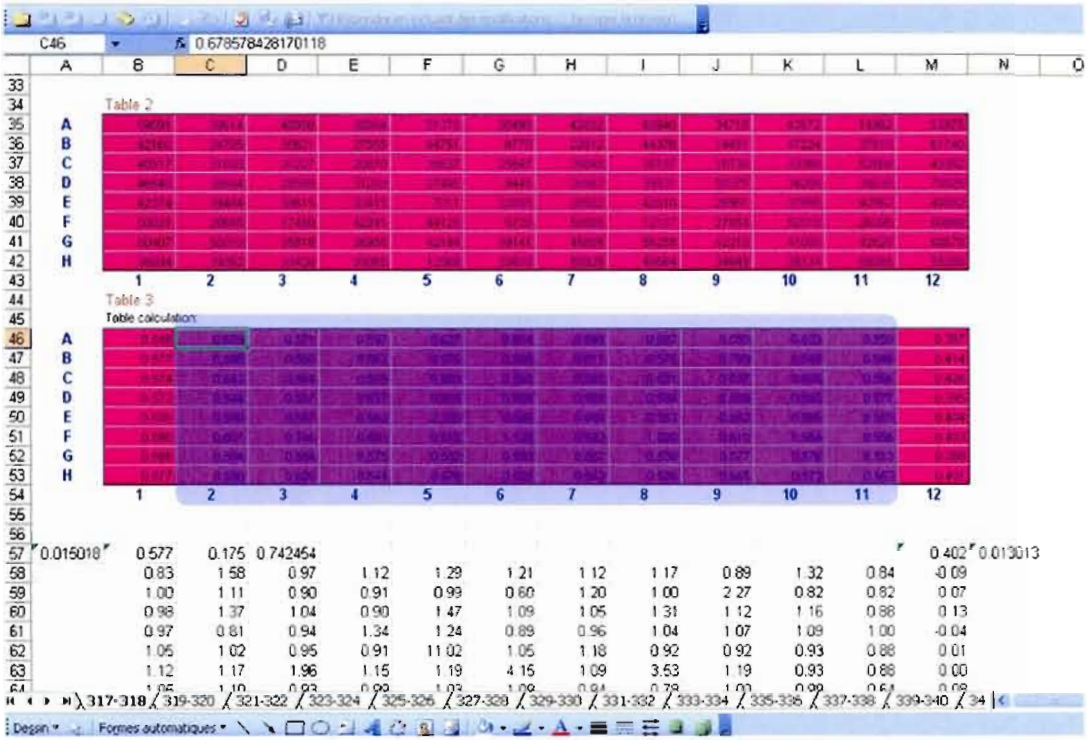
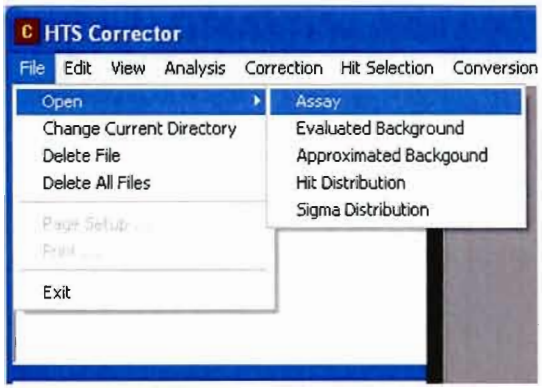
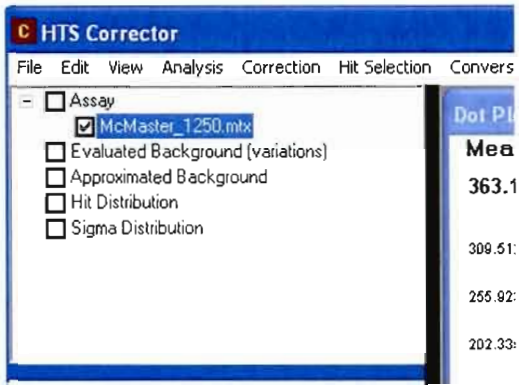


Figure 5.6 : Exemple d'un fichier Excel pouvant être traité par « HTS Corrector ». La zone prise en considération, dans cet exemple, est délimitée par les lignes 46 et 53 et par les colonnes C et L.

Évaluation de l'arrière-plan (« background evaluation ») :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour le calcul de l'arrière-plan (*Properties and Settings : General & Analysis*);
- Sélectionnez l'option ***Analysis => Background Evaluation***;
- Sauvegardez le fichier;
- Le fichier est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).





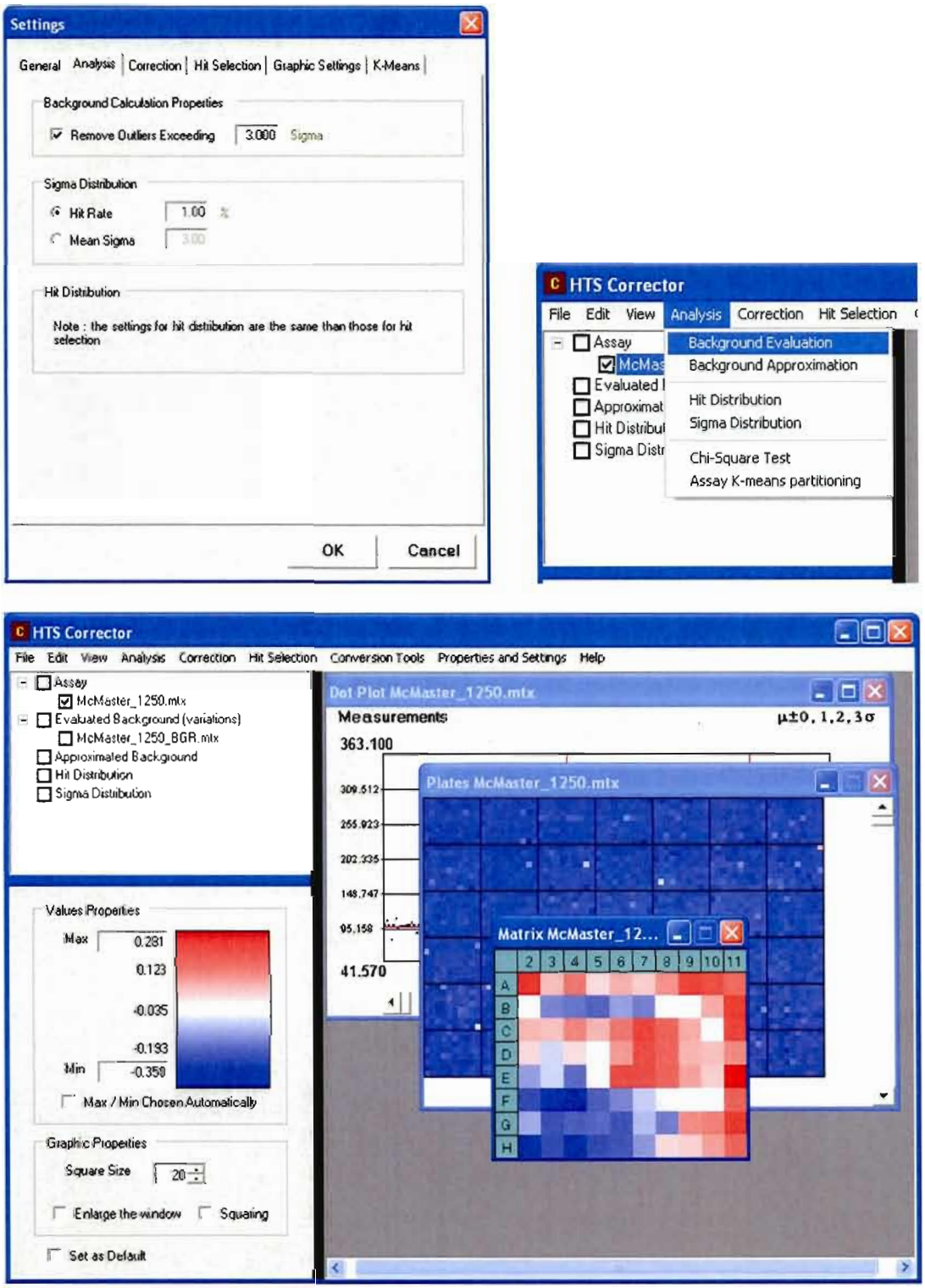
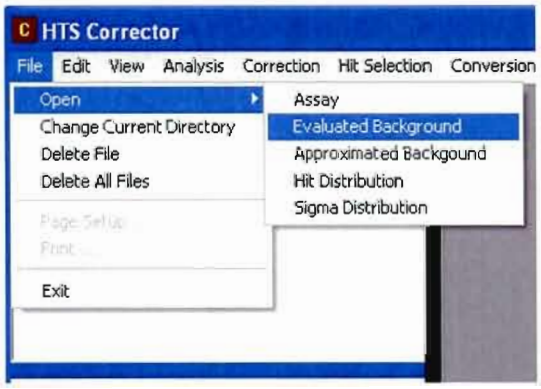
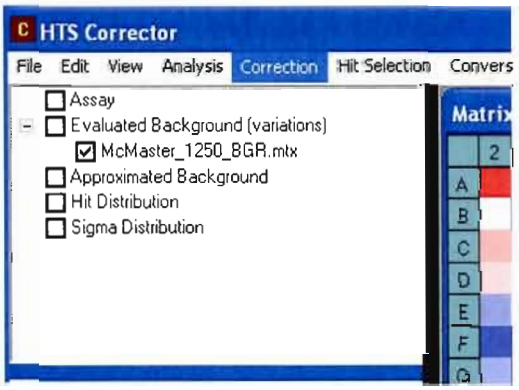


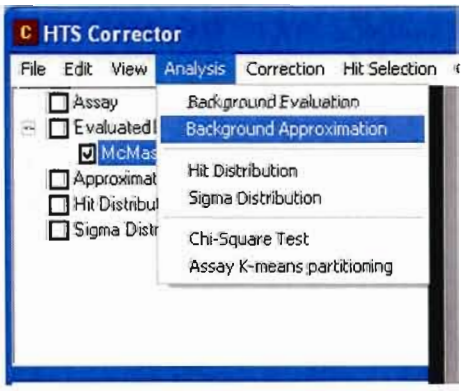
Figure 5.7 : Résultats d'une évaluation de l'arrière-plan.

Approximation de l'arrière-plan (« background approximation ») :

- Ouvrez un fichier contenant les données d'un arrière-plan;
- Sélectionnez le fichier dans l'arborescence;
- Sélectionnez l'option ***Analysis => Background Approximation***;
- Sauvegardez le fichier;
- Le fichier est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).







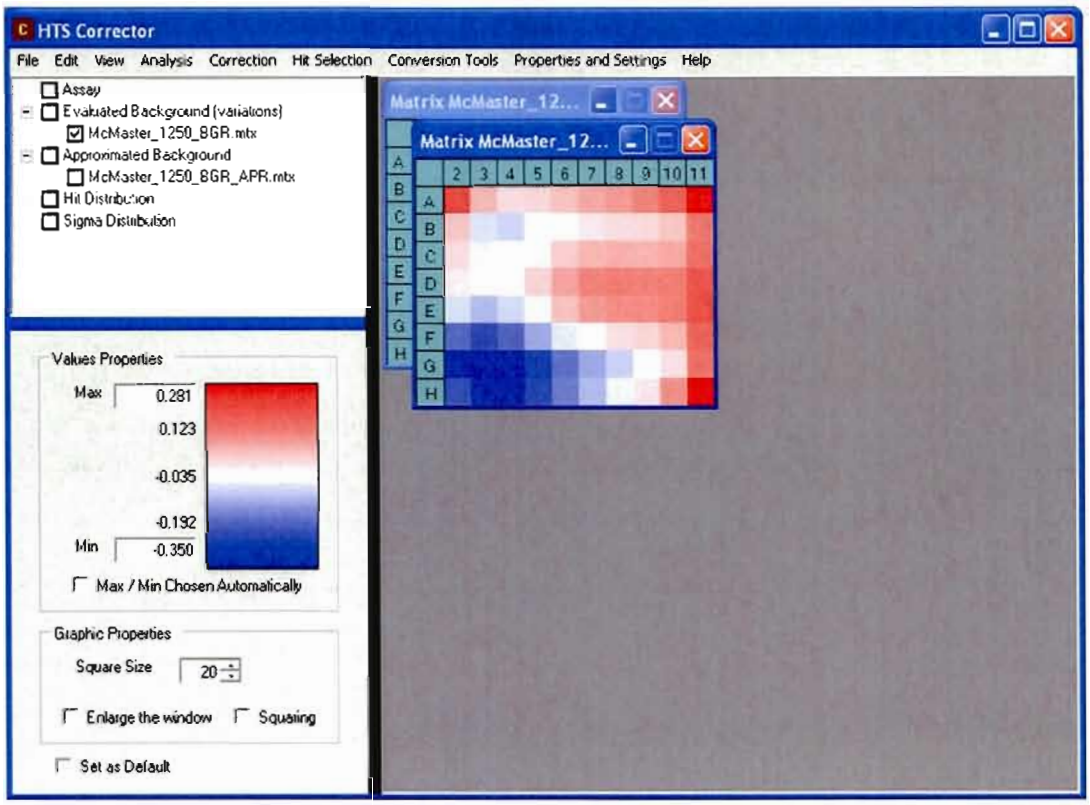
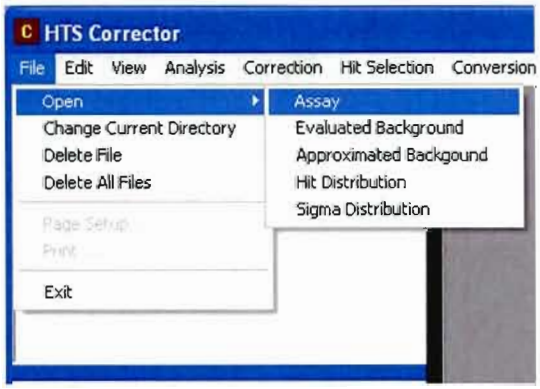
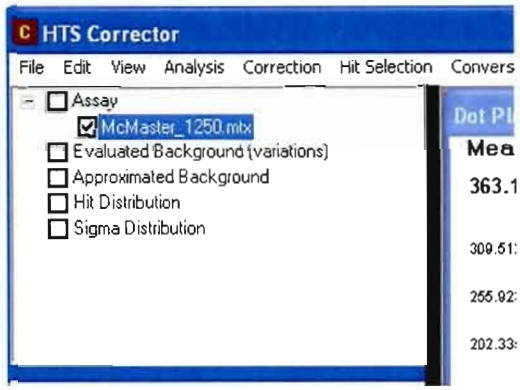


Figure 5.8 : Résultats d'une approximation de l'arrière-plan.

Distribution des « hits » (« hit distribution ») :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour le calcul de la distribution des « hits » (*Properties and Settings : General & Hit Selection*);
- Sélectionnez l'option ***Analysis => Hit Distribution***;
- Sauvegardez le fichier;
- Le fichier est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).





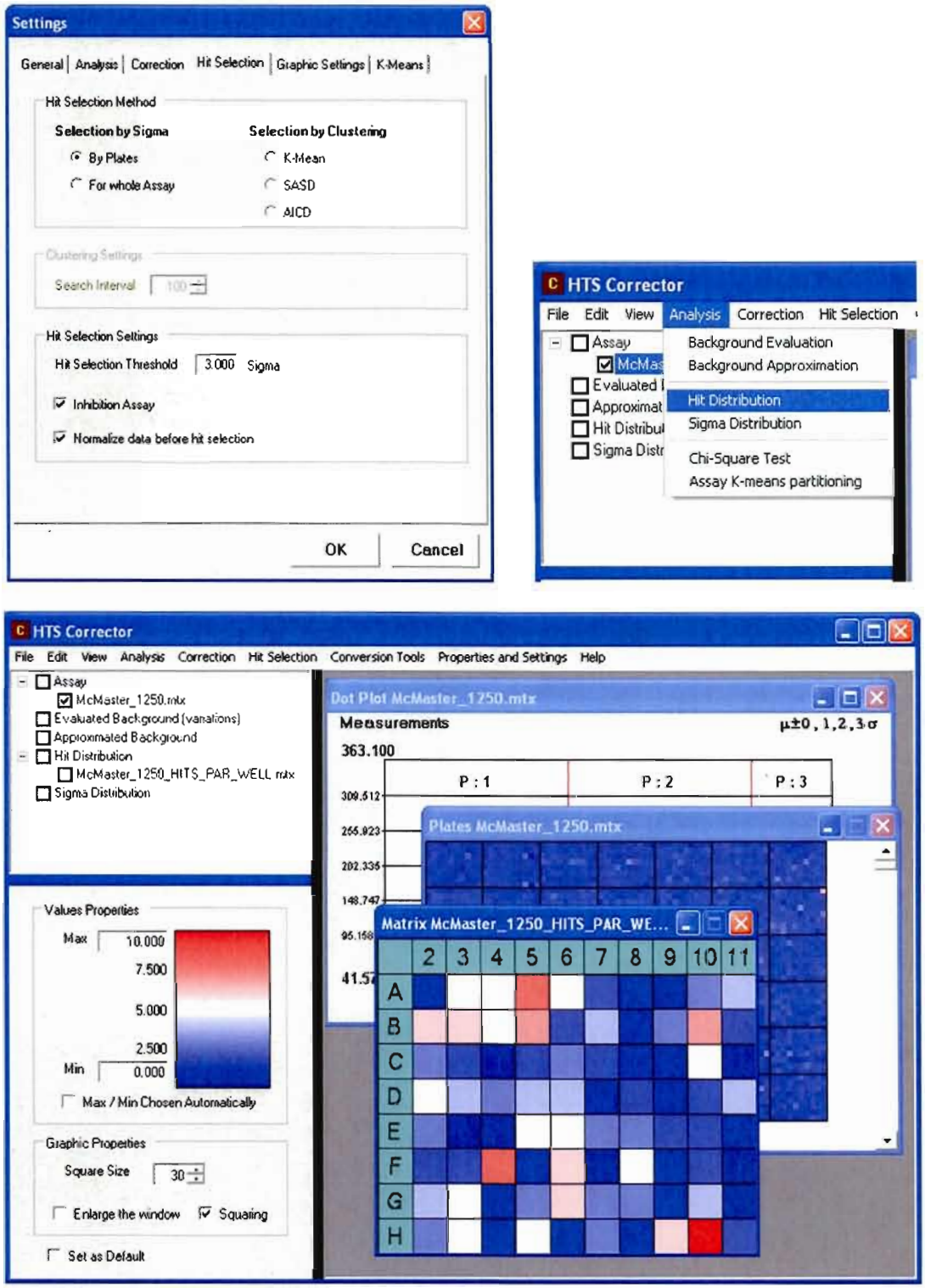
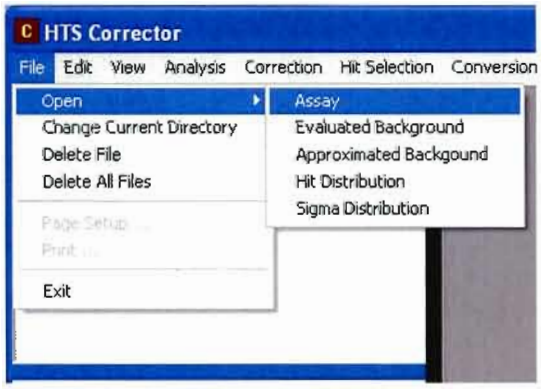
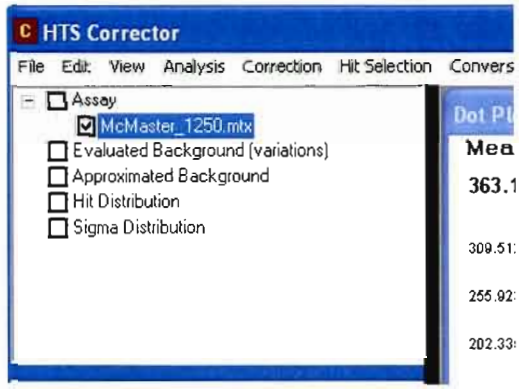


Figure 5.9 : Résultats d'une distribution des « hits ».

Distribution des seuils de sélection des « hits » (« sigma distribution ») :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour le calcul de la distribution des seuils de sélection des « hits » (*Properties and Settings : General & Analysis*);
- Sélectionnez l'option ***Analysis => Sigma Distribution***;
- Sauvegardez le fichier;
- Le fichier est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif);
- Sélectionnez le fichier dans l'arborescence;
- Sélectionnez l'option ***View => Table*** pour visualiser la distribution des seuils de sélection des « hits » des puits.





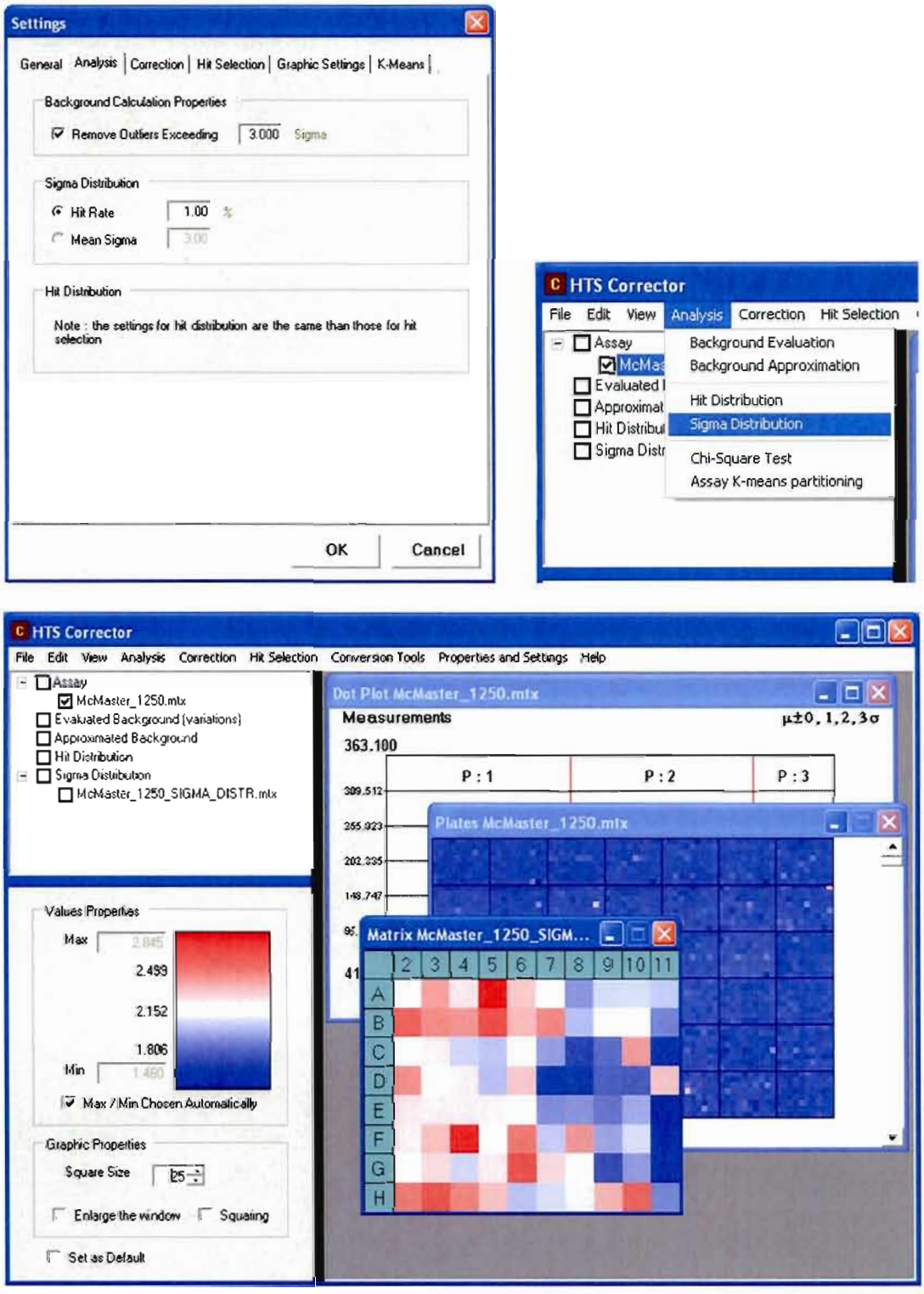


Figure 5.10 : Résultats d'une distribution des seuils de sélection des « hits ».

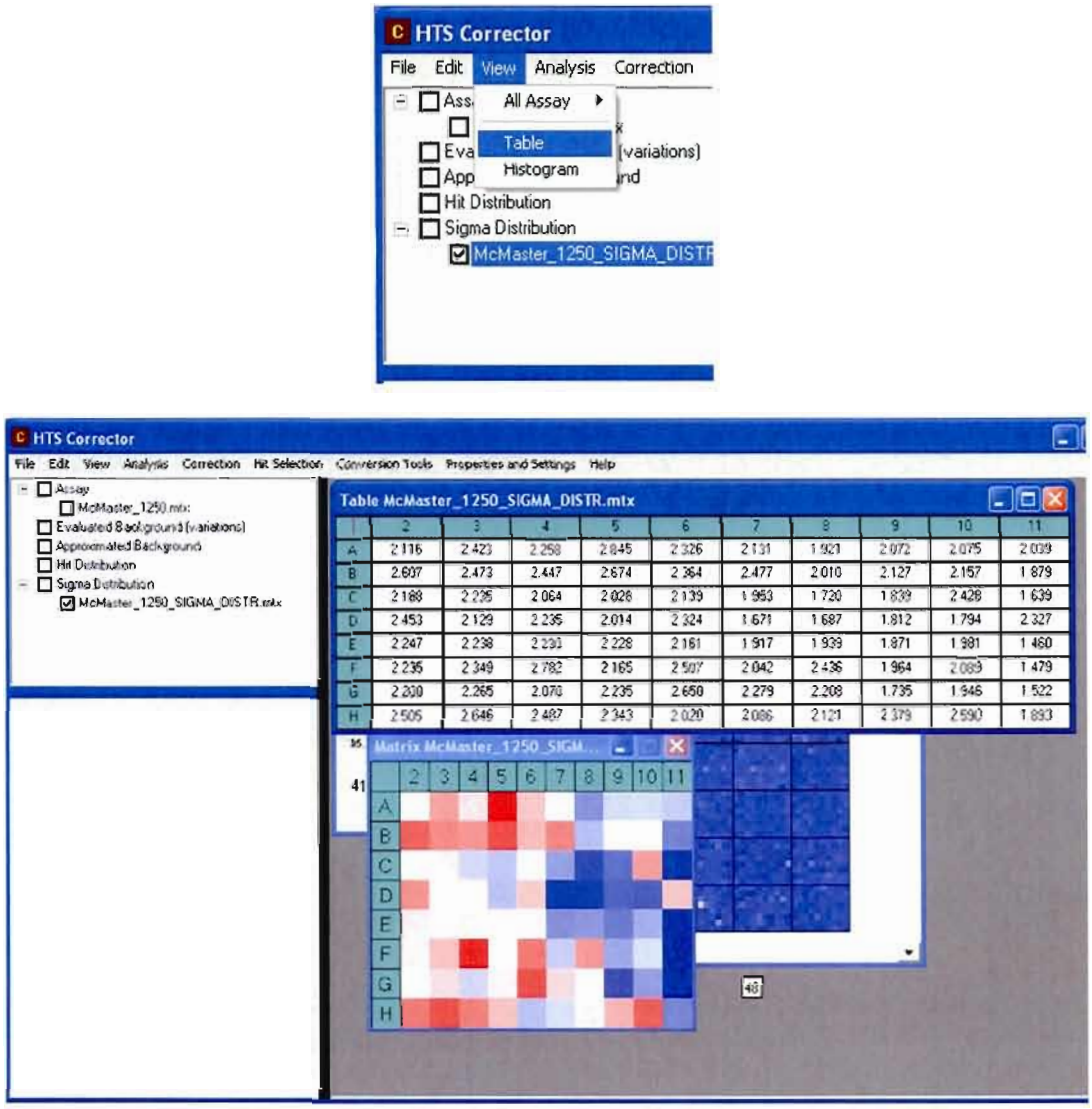
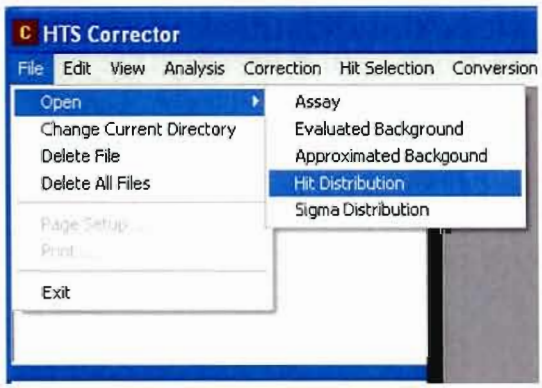
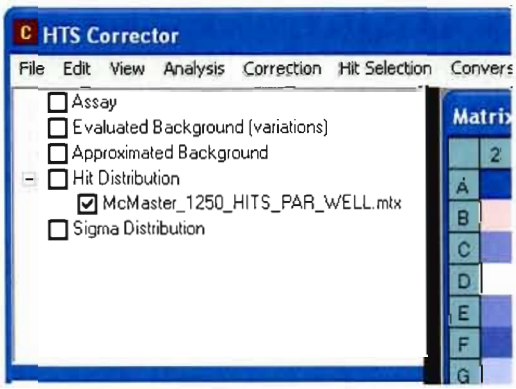


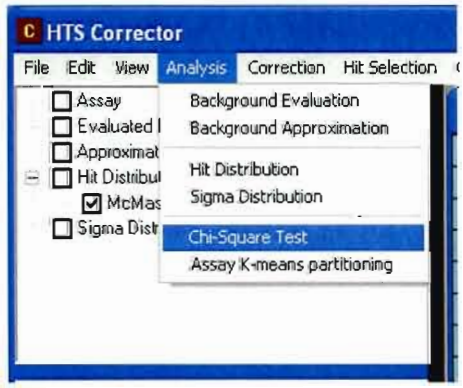
Figure 5.11 : Visualisation des valeurs numériques d'une distribution des seuils de sélection des « hits ».

Test de contingence de χ^2 (« chi-square contingency test ») :

- Ouvrez un fichier contenant une distribution des « hits » d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Sélectionnez l'option ***Analysis => Chi-Square Test***;
- Choisissez le paramètre *alpha* du test;
- Les résultats du test apparaissent dans une boite de dialogue.







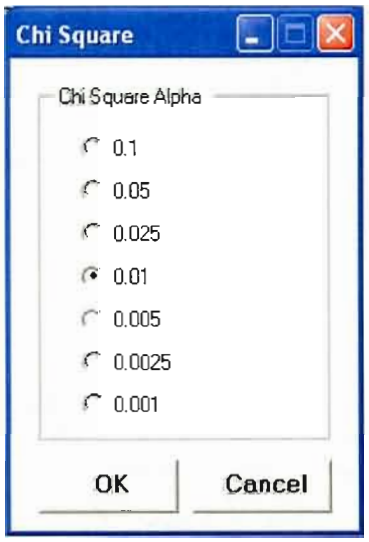
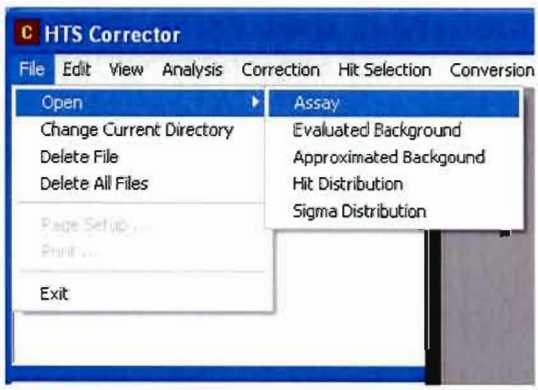
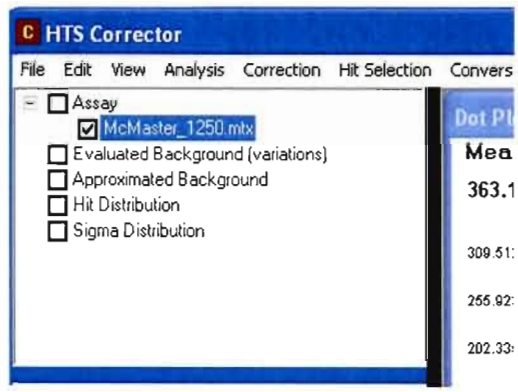


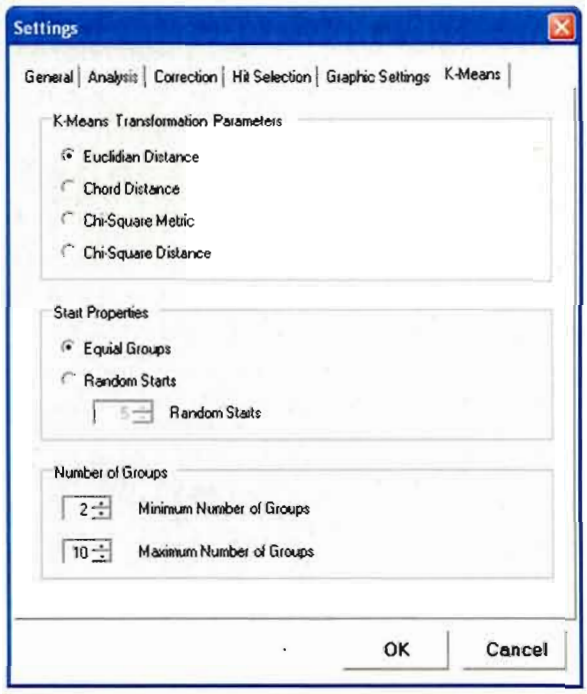
Figure 5.12 : Résultats d'un test de contingence de χ^2.

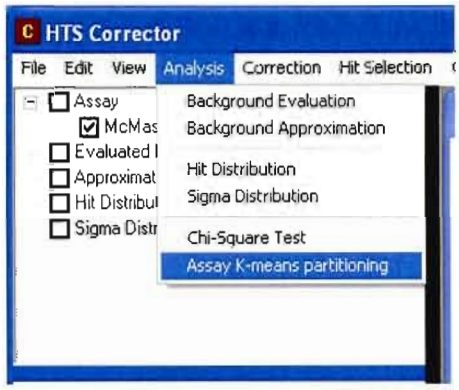
Groupement d'une campagne de criblage HTS par l'algorithme k-mean (« k-means partitioning ») :

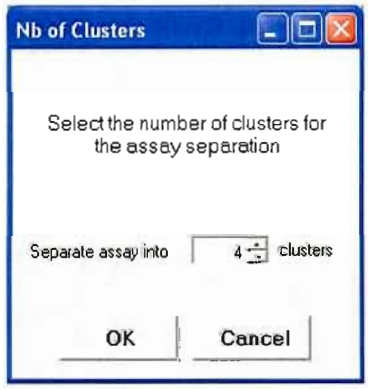
- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour le calcul du groupement par *k-means* (*Properties and Settings : General & K-means*);
- Sélectionnez l'option ***Analysis => Assay K-means partitioning***;
- Le résultat de l'analyse par l'algorithme *k-means* apparait dans une boite de dialogue;
- Sélectionnez le nombre de groupes que vous voulez créer à partir des données initiales;
- Sauvegardez les fichiers;
- Les fichiers sont ajoutés à l'arborescence;
- « Double-cliquez » sur les noms des fichiers dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).

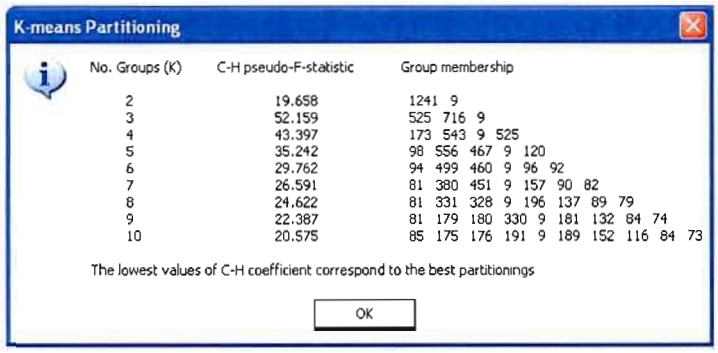





Settings
General
Analysis
Correction
Hit Selection
Graphic Settings
K-Means
K-Means Transformation Parameters
Euclidian Distance
Chord Distance
Chi-Square Metric
Chi-Square Distance
Start Properties
Equal Groups
Random Starts
Random Starts
Number of Groups
2
Minimum Number of Groups
10
Maximum Number of Groups
OK
Cancel


HTS Corrector
File
Edit
View
Analysis
Correction
Hit Selection
Assay
Evaluated
Approximat
Hit Distribut
Sigma Distr
Background Evaluation
Background Approximation
Hit Distribution
Sigma Distribution
Chi-Square Test
Assay K-means partitioning


Nb of Clusters
Select the number of clusters for the assay separation
Separate assay into
4
clusters
OK
Cancel


K-means Partitioning
No. Groups (K)
C-H pseudo-F-statistic
Group membership
2 19.658 1241 9
3 52.159 525 716 9
4 43.397 173 543 9 525
5 35.242 98 556 467 9 120
6 29.762 94 499 460 9 96 92
7 26.591 81 380 451 9 157 90 82
8 24.622 81 331 328 9 196 137 89 79
9 22.387 81 179 180 330 9 181 132 84 74
10 20.575 85 175 176 191 9 189 152 116 84 73
The lowest values of C-H coefficient correspond to the best partitionings
OK

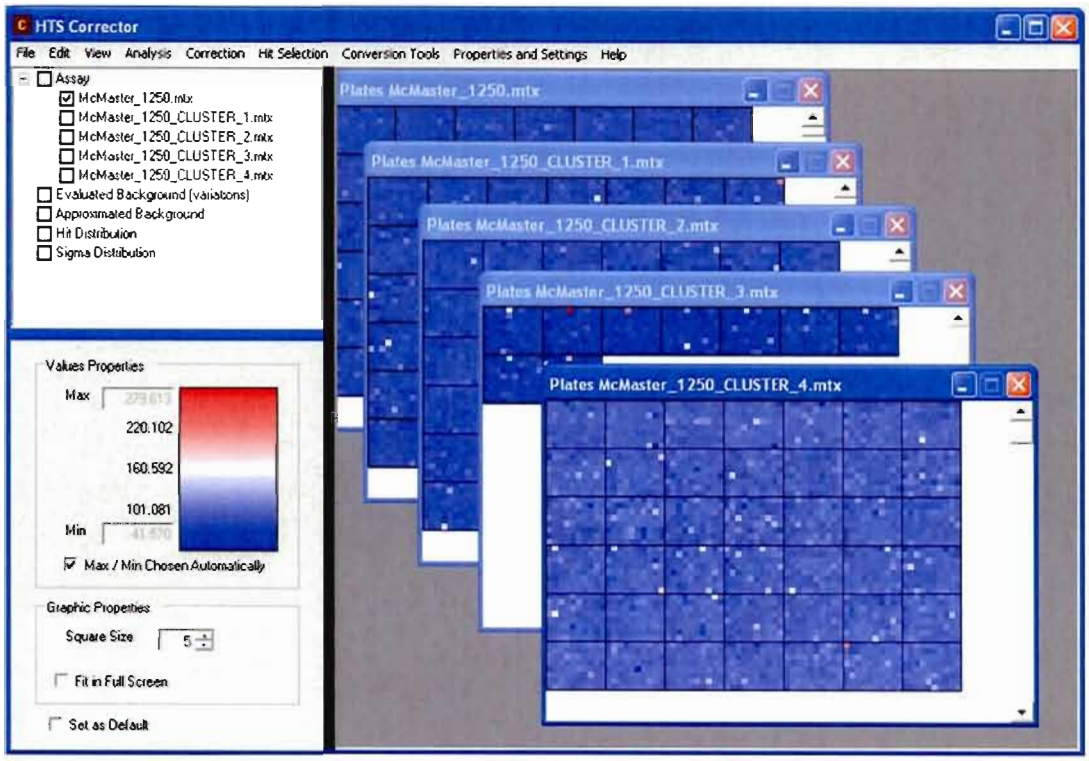
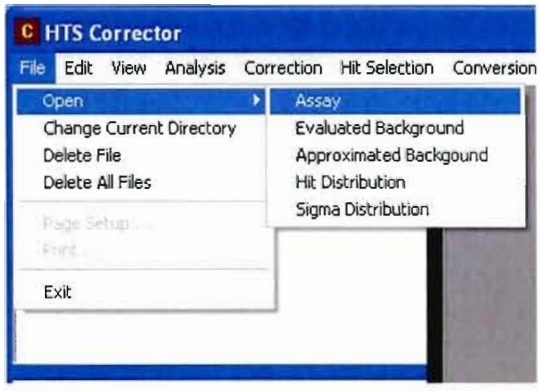
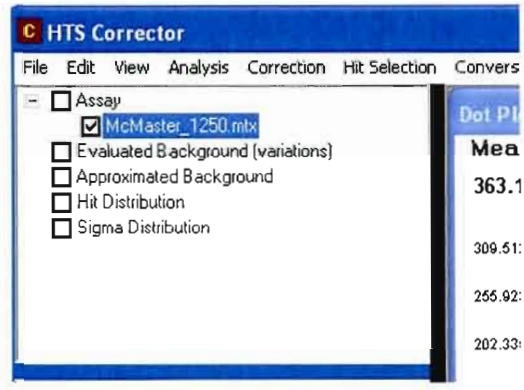


Figure 5.13 : Résultats d'un groupement par *k-means*.

Correction par puits (« well correction ») :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour le calcul de la correction par puits (*Properties and Settings : General & Correction*);
- Sélectionnez l'option ***Correction => Well Correction***;
- Sauvegardez le fichier;
- Le fichier corrigé est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).





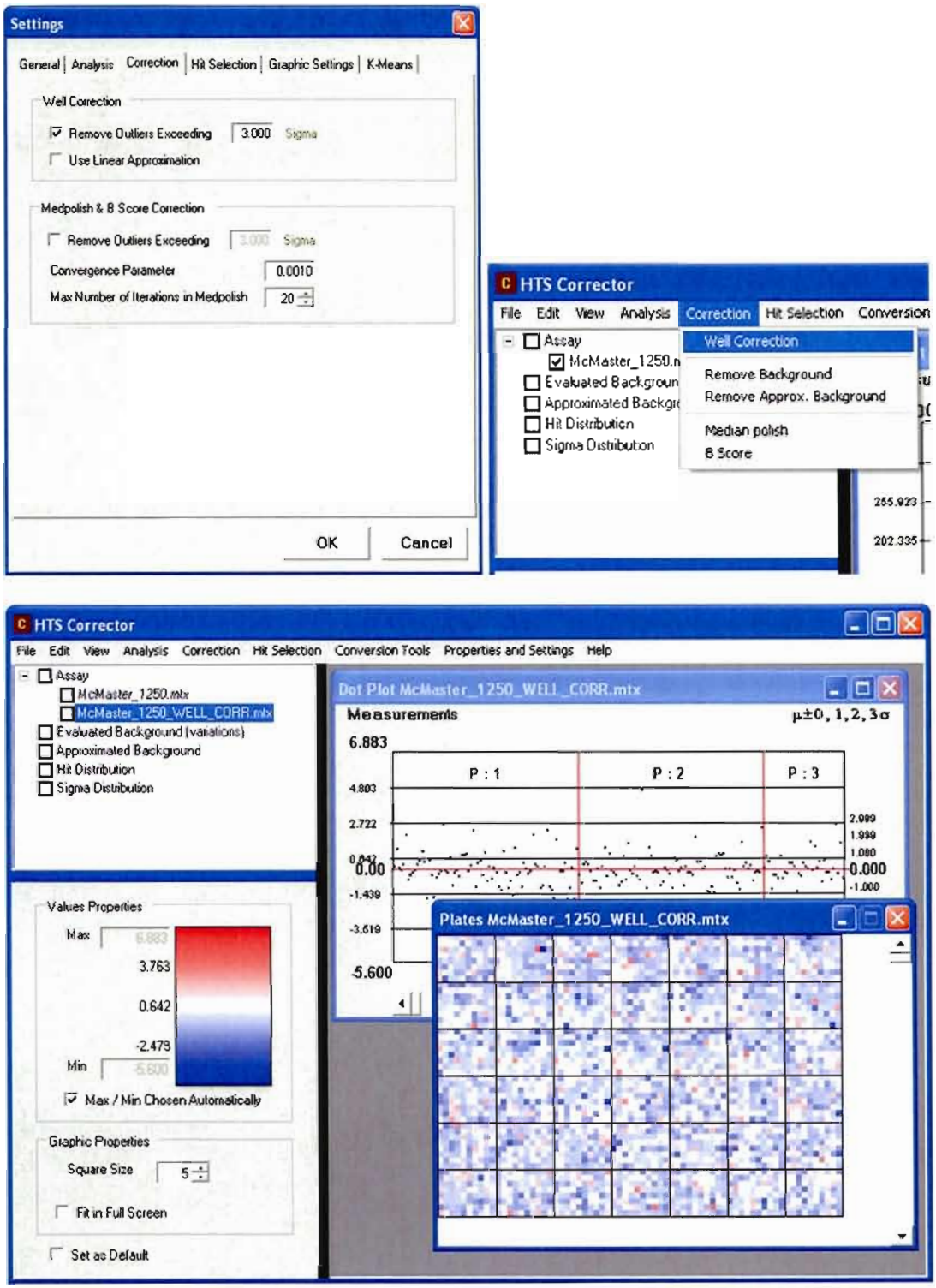
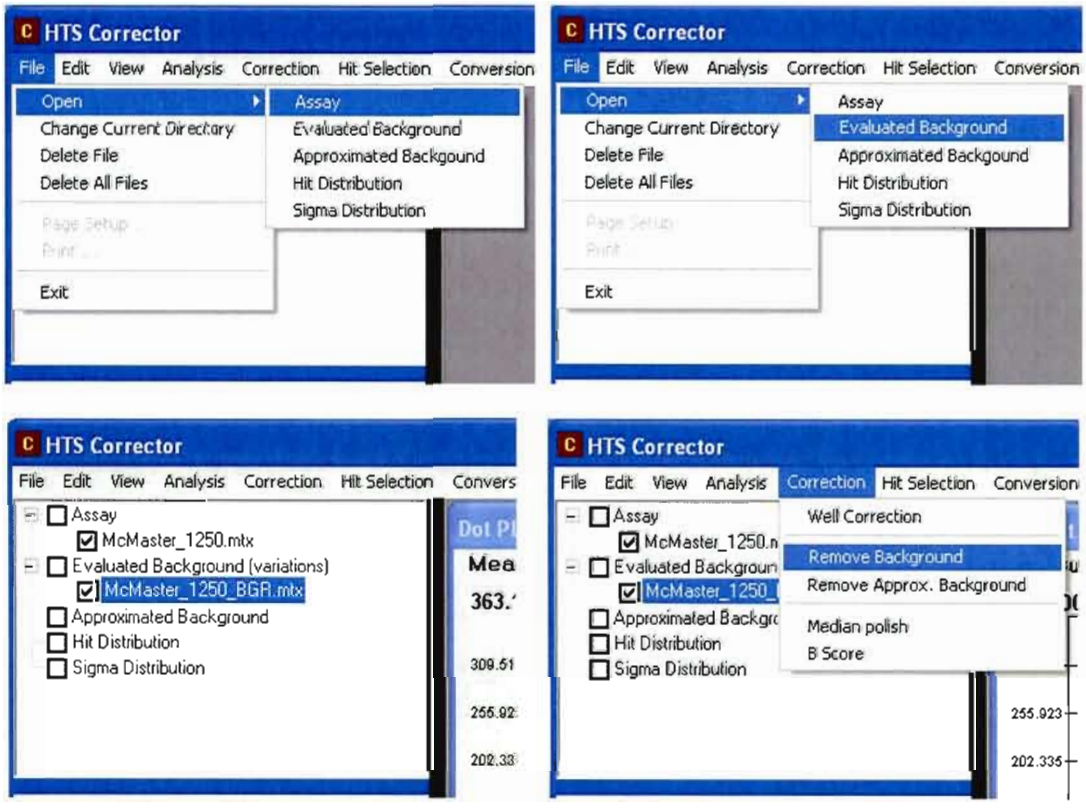


Figure 5.14 : Résultats d'une correction par puits.

Correction par soustraction de l'arrière-plan (« remove background ») :

- Ouvrez un fichier contenant les données d'une campagne HTS et un fichier contenant les données d'un arrière-plan (évaluée ou approximée);
- Sélectionnez les fichiers dans l'arborescence;
- Sélectionnez l'option ***Correction => Remove Background (or Remove Aprrox. Background)***;
- Sauvegardez le fichier;
- Le fichier corrigé est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).



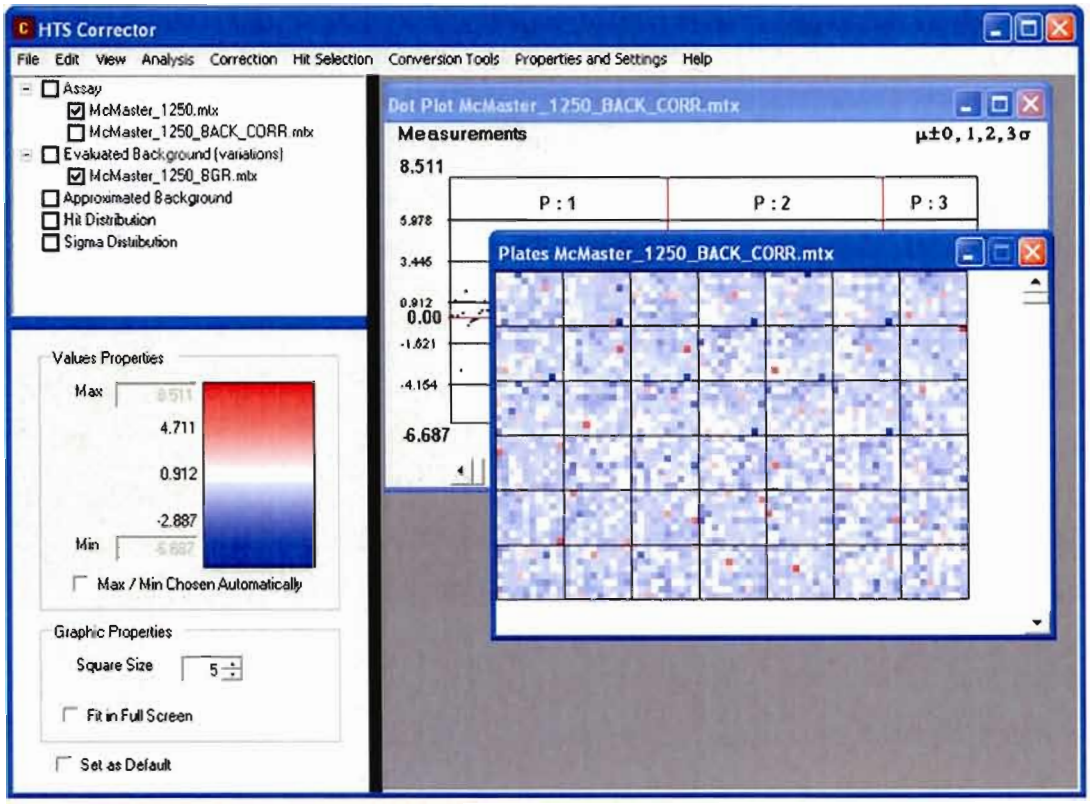
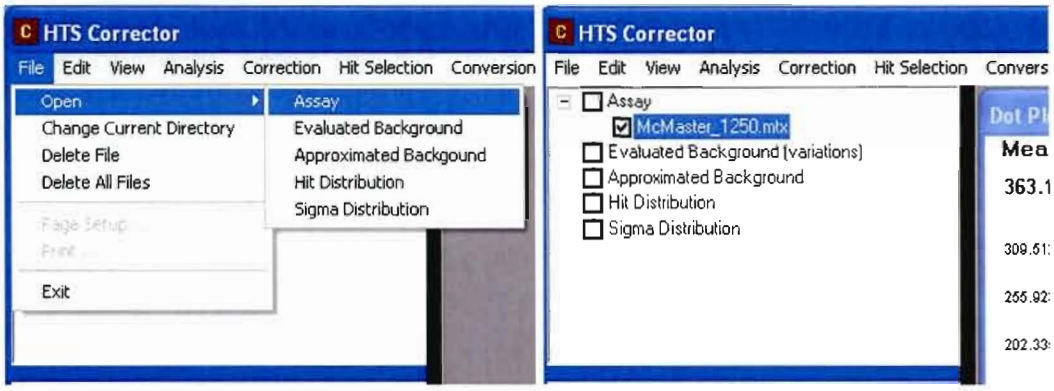


Figure 5.15 : Résultats d'une correction par soustraction de l'arrière-plan.

Correction par « median polish » :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour le calcul de « *median polish* » (*Properties and Settings : General & Correction*);
- Sélectionnez l'option ***Correction => Median polish***;
- Sauvegardez le fichier;
- Le fichier corrigé est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).



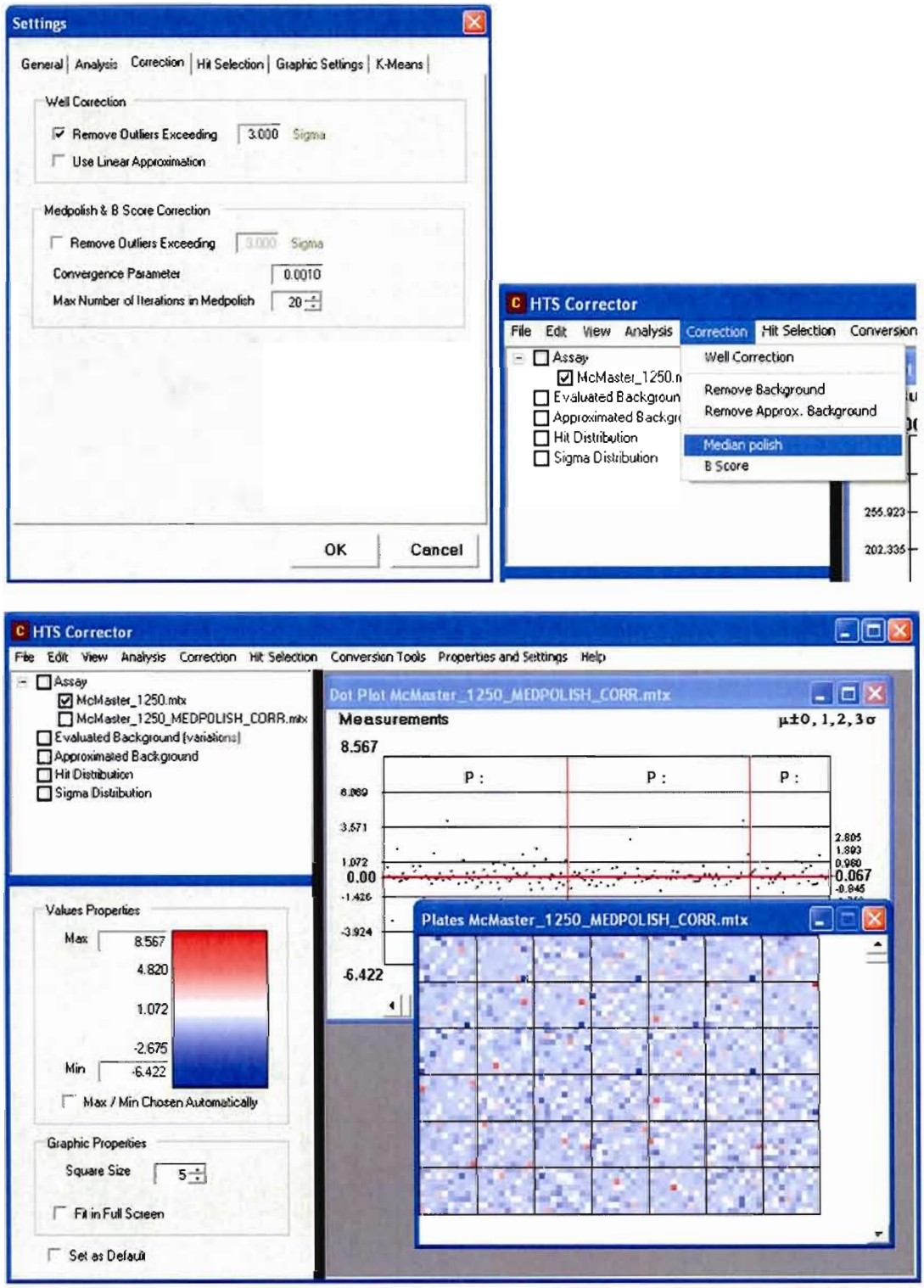
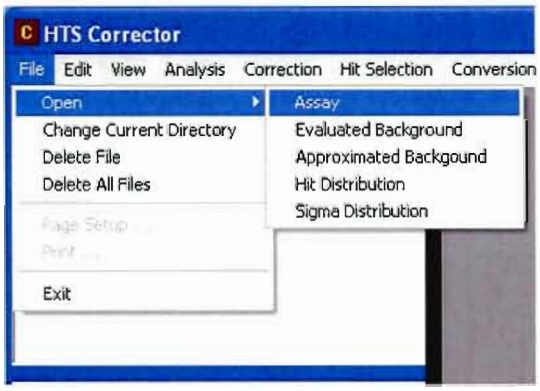
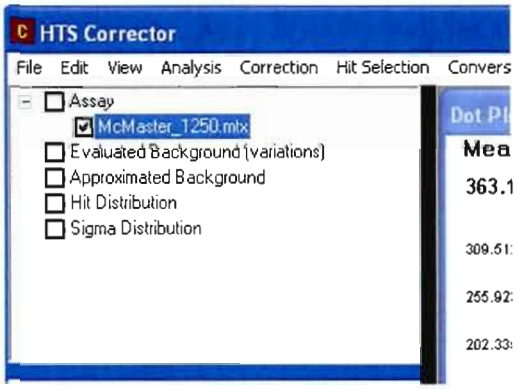


Figure 5.16 : Résultats d'une correction par « *median polish* ».

Correction par B score :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour le calcul de B score (*Properties and Settings : General & Correction*);
- Sélectionnez l'option ***Correction => B Score***;
- Sauvegardez le fichier;
- Le fichier corrigé est ajouté à l'arborescence;
- « Double-cliquez » sur le nom du fichier dans l'arborescence;
- Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).





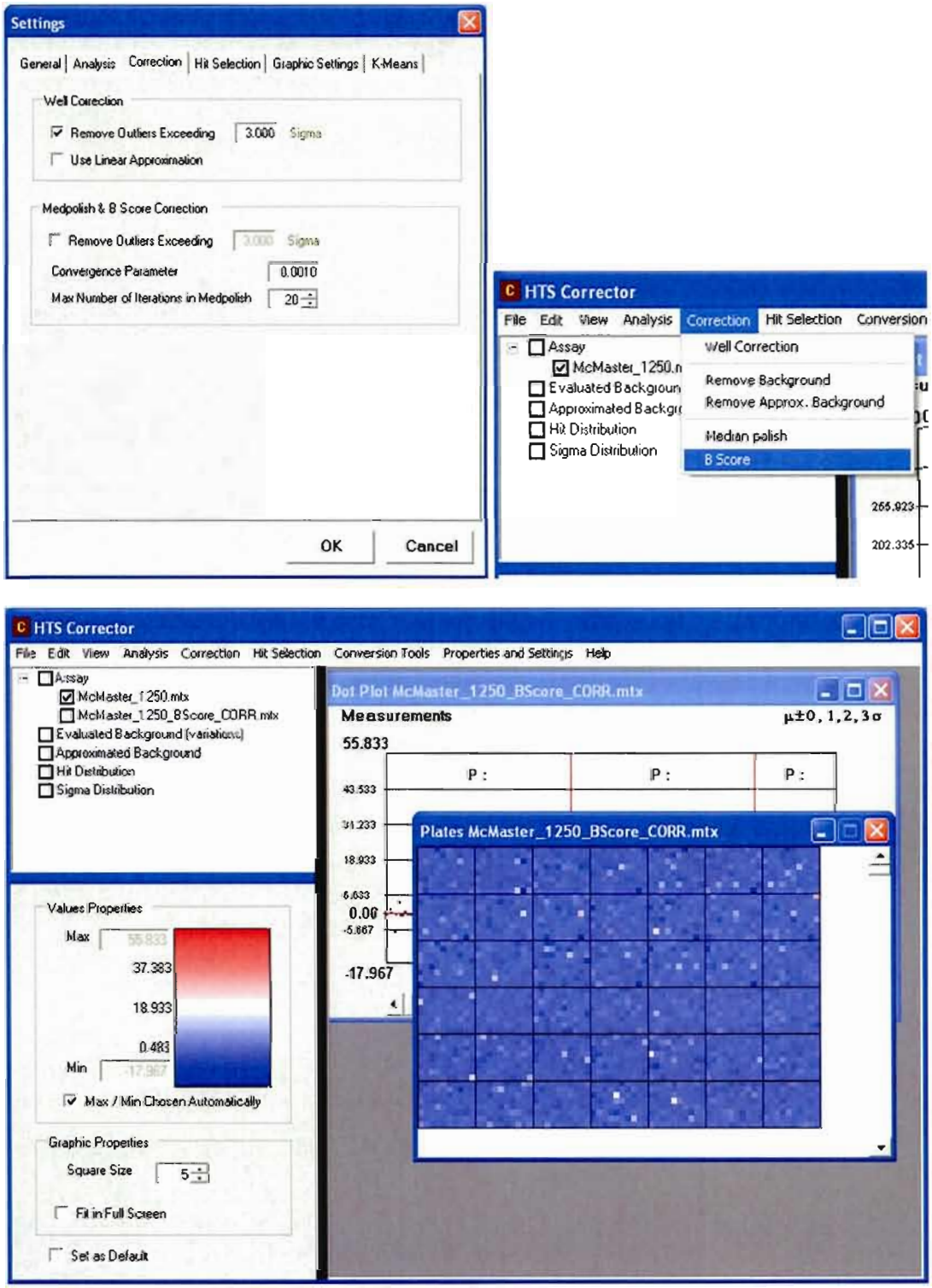
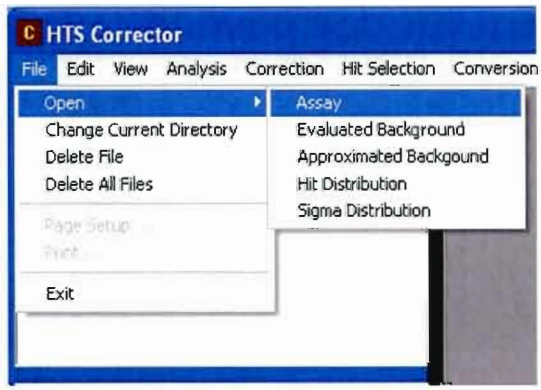
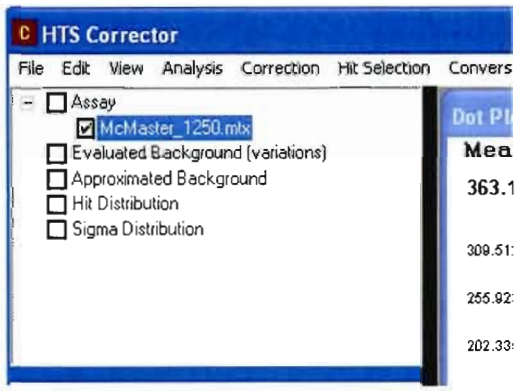


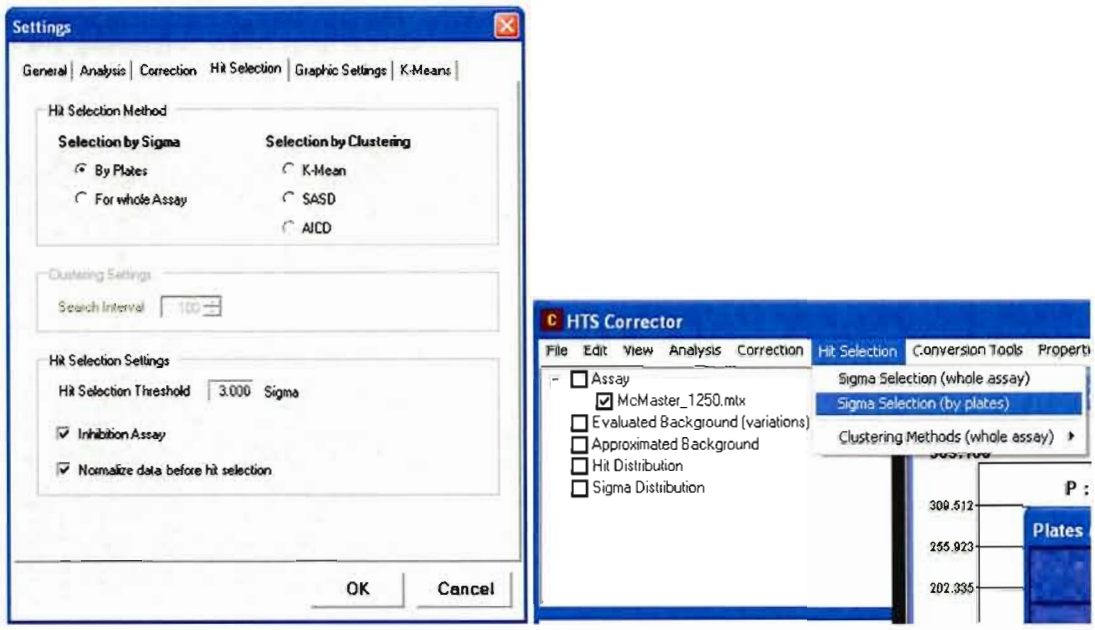
Figure 5.17 : Résultats d'une correction par B score.

Sélection des « hits » par un seuil prédéfini :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour la sélection des « hits » par un seuil prédéfini (*Properties and Settings : General & Hit Selection*);
- Sélectionnez l'option ***Hit Selection => Sigma Selection (whole assay)*** ou ***Sigma Selection (by plates)***;
- Sauvegardez le fichier;
- Vous pouvez ouvrir le fichier des résultats avec un éditeur de texte; les valeurs identifiées comme des « hits » seront suivies par « * »;
- Vous pouvez créer une liste des « hits » en utilisant l'outil fourni dans « HTS Corrector » sous l'option ***Conversion Tools => Hit List from the Hit mtx file.***








```
<HITS plates="1250" rows="8" colums="10" hits="245">

<Plate name="1" no="1">
101 750 106.160 100.850 78.630* 101.340 109.670 97.010  99.130  99.620  100.110
100 760 101 260 104.520 100.930 88.920  101.420 97.740  90 720  97.660  99.300
99.130  113.100 99 700  98.800  103.540 95.950  99.380  126.410 97 420  104.110
103.950 103.540 98.810  100.440 110.650 96.110  101 990 101 910 98.310  104.850
95.210  99 700  95.700  99 950  99.620  95.870  94.390  104.600 98.640  96.350
94.890  89.980  103.540 96.600  97 740  94.310  100.850 99.210  98.070  99 790
109.010 99 700  98.890  92 760  93.250  99.460  109.100 95.290  94.560  92.760
107.300 83.860  96.850  97 420  100.030 89.580  94.070  99.870  91 780  106.070
</Plate>

<Plate name="2" no="2">
100.620 96.990  103.690 104.180 103.450 100.710 98.200  99.170  102.480 96.270
101.750 107.490 97.800  106.030 99.660  99.250  99.330  95.780  97.800  106.600
98.450  109.990 101.510 100.140 102.720 100.540 99.010  124.430 75.280  97.320
108.290 104.980 100.140 100 710 100.790 97.800  98.610  101.920 101.350 100 950
94.900  104.100 102.890 100.620 102.970 97.640  103.130 100.790 100.460 101.430
110.070 101.670 101.270 100.060 101.190 94.170  94.010  95.940  94.730  105.790
105.550 104.580 104.980 100.220 97.480  100.540 97.800  101.430 98.040  97.070
108.700 100 550 101 760 99 900  101.030 96.910  100.220 135.330 50 750* 114.590
</Plate>

<Plate name="3" no="3">
103.190 105.830 104.510 93.220  99.160  98.990  98.410  106.820 99.160  100.720
98.910  102.200 102.040 100.470 91.400  98.080  98.330  102.780 102.530 102.370
110.450 100.970 108.880 101.710 104.180 98.000  104.590 100.220 99.730  99.070
```

Figure 5.18 : Fichier résultat pour une recherche des « hits ».

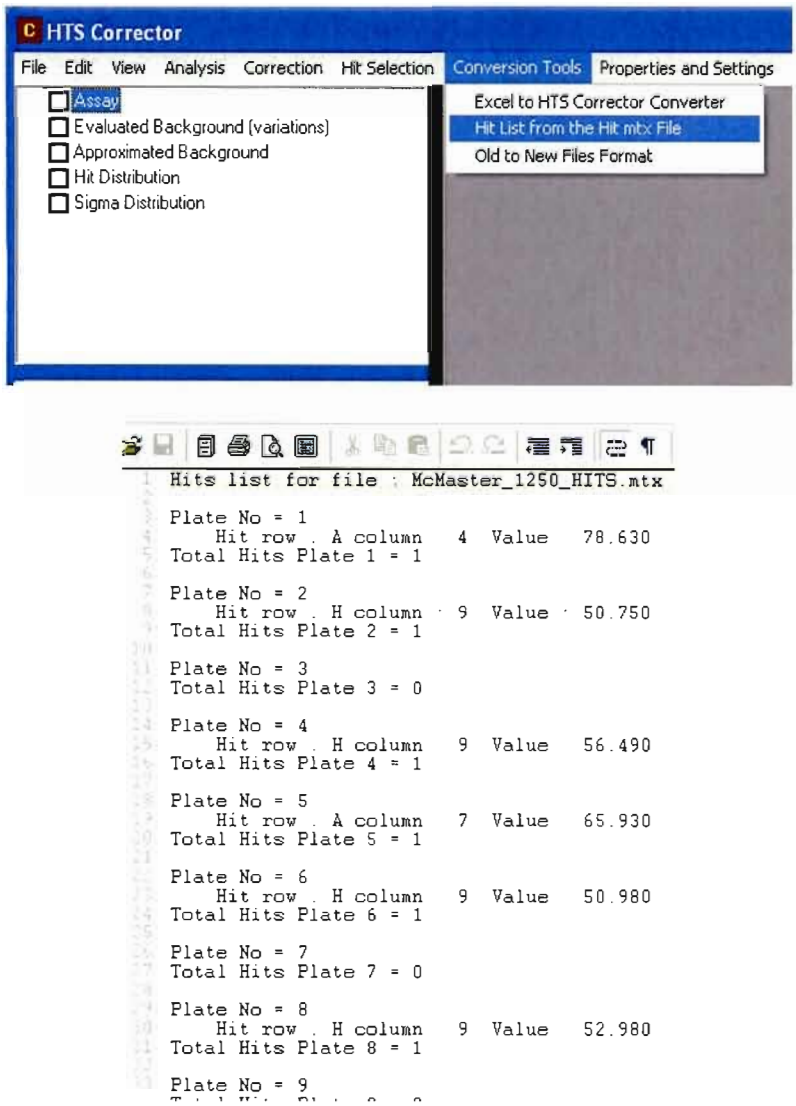
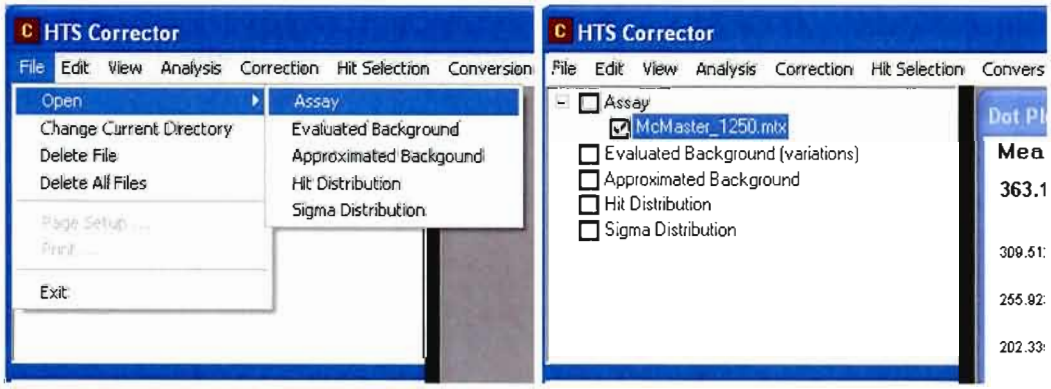


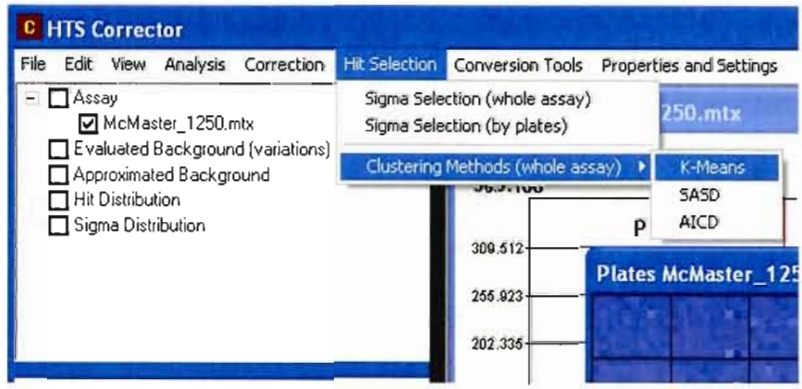
Figure 5.19 : Fichier résultat pour une transformation d'un fichier de « hits » en une liste de « hits ».

Sélection des « hits » par des méthodes de groupement (« clustering ») :

- Ouvrez un fichier contenant les données d'une campagne HTS;
- Sélectionnez le fichier dans l'arborescence;
- Choisissez les paramètres voulus pour la sélection des « hits » par des méthodes de groupement (*Properties and Settings : General & Hit Selection*);
- Sélectionnez l'option ***Hit Selection => Clustering Methods => K-Means*** ou ***SASD*** ou ***AICD)***;
- Sauvegardez le fichier;
- Vous pouvez ouvrir le fichier des résultats avec un éditeur de texte; les valeurs identifiées comme des « hits » seront suivies par « * »;
- Vous pouvez créer une liste des « hits » en utilisant l'outil fourni dans « HTS Corrector » sous l'option ***Conversion Tools => Hit List from the Hit mtx file.***

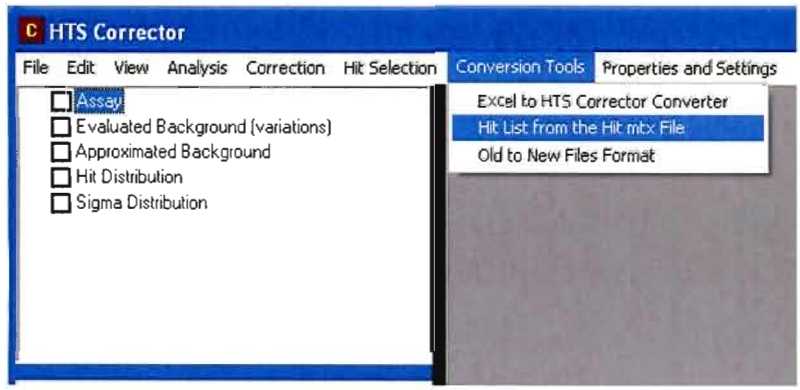



Settings
General
Analysis
Correction
Hit Selection
Graphic Settings
K-Means
Hit Selection Method
Selection by Sigma
By Plates
For whole Assay
Selection by Clustering
K-Mean
SASD
AICD
Clustering Settings
Search Interval
100
Hit Selection Settings
Hit Selection Threshold
3.000
Sigma
Inhibition Assay
Normalize data before hit selection
OK
Cancel


HTS Corrector
File
Edit
View
Analysis
Correction
Hit Selection
Conversion Tools
Properties and Settings
Assay
McMaster_1250.mtx
Evaluated Background (variations)
Approximated Background
Hit Distribution
Sigma Distribution
Sigma Selection (whole assay)
Sigma Selection (by plates)
Clustering Methods (whole assay)
K-Means
SASD
AICD
250.mtx
309.512
255.923
202.335
Plates McMaster_125

```
<HITS plates="1250" rows="8" colums="10" hits="245">

<Plate name="1" no="1">
101.750 106.160 100.850 78.630* 101 340 109.670 97.010  99.130  99.620  100.110
100.760 101.260 104.520 100.930 88.920  101.420 97 740  90.720  97.660  99.300
99.130  113.100 99 700  98.800  103 540 95.950  99.380  126.410 97.420  104.110
103.950 103.540 98.810  100.440 110.650 96.110  101.990 101.910 98.310  104.850
95.210  99 700  95.700  99.950  99.620  95.870  94.390  104.600 98.640  96.350
94.890  89.980  103.540 96.600  97.740  94.310  100.850 99.210  98.070  99 790
109.010 99 700  98.890  92.760  93.250  99.460  109.100 95.290  94.560  92.760
107.300 83.860  96.850  97 420  100.030 89.580  94.070  99.870  91 780  106.070
</Plate>

<Plate name="2" no="2">
100.620 96.990  103.690 104.180 103.450 100 710 98.200  99.170  102.480 96.270
101.750 107 490 97.800  106.030 99.660  99.250  99.330  95 780  97.800  106.600
98.450  109.990 101.510 100.140 102.720 100.540 99.010  124.430 75.280  97.320
108.290 104.980 100.140 100 710 100 790 97.800  98.610  101.920 101.350 100 950
94.900  104 100 102.890 100.620 102.970 97 640  103.130 100.790 100.460 101.430
110.070 101.670 101.270 100.060 101.190 94 170  94.010  95.940  94 730  105 790
105.550 104.580 104.980 100.220 97.480  100.540 97.800  101.430 98.040  97.070
108.700 100.550 101 760 99 900  101.030 96 910  100.220 135.330 50 750* 114.590
</Plate>

<Plate name="3" no="3">
103.190 105.830 104.510 93.220  99.160  98.990  98.410  106.820 99.160  100.720
98.910  102.200 102.040 100.470 91 400  98.080  98.330  102 780 102.530 102.370
110.450 100.970 108.880 101.710 104.180 98.000  104.590 100.220 99 730  99.070
```

Figure 5.20 : Fichier résultat pour une recherche des « hits ».



```
Hits list for file : McMaster_1250_HITS.mtx

Plate No = 1
    Hit row : A column : 4  Value   78.630
Total Hits Plate 1 = 1

Plate No = 2
    Hit row : H column : 9  Value   50.750
Total Hits Plate 2 = 1

Plate No = 3
Total Hits Plate 3 = 0

Plate No = 4
    Hit row : H column : 9  Value   56.490
Total Hits Plate 4 = 1

Plate No = 5
    Hit row : A column   7  Value   65.930
Total Hits Plate 5 = 1

Plate No = 6
    Hit row : H column   9  Value   50.980
Total Hits Plate 6 = 1

Plate No = 7
Total Hits Plate 7 = 0

Plate No = 8
    Hit row : H column : 9  Value   52.980
Total Hits Plate 8 = 1

Plate No = 9
```

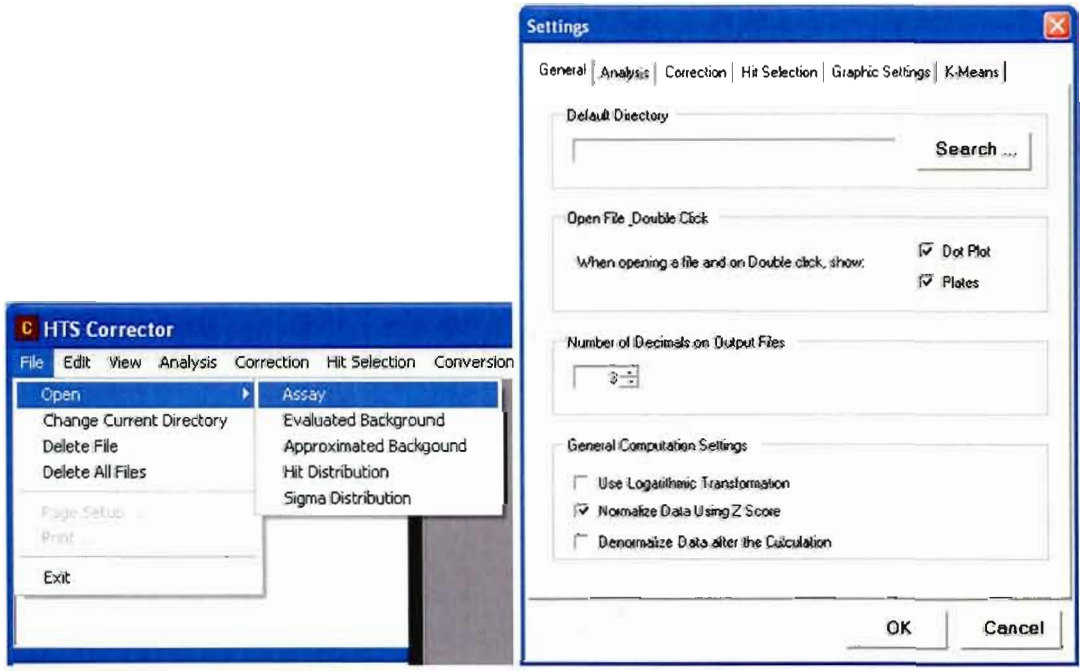
Figure 5.21 : Fichier résultat pour une transformation d'un fichier de « hits » en une liste de « hits ».

Visualisation des données :

- Pour les données d'une campagne HTS
  - Ouvrez un fichier contenant les données d'une campagne HTS;
  - Choisissez le fonctionnement par défaut pour l'ouverture des fichiers et le « Double-click » (*Properties and Settings : General*);
  - « Double-cliquez » sur le nom du fichier dans l'arborescence;

  ou

  - Sélectionnez le fichier dans l'arborescence;
  - Sélectionnez l'option ***View*** => ***All Assay*** => ***Dot Plot*** ou ***Plates***;
  - Sélectionnez l'option ***View*** => ***Table***.



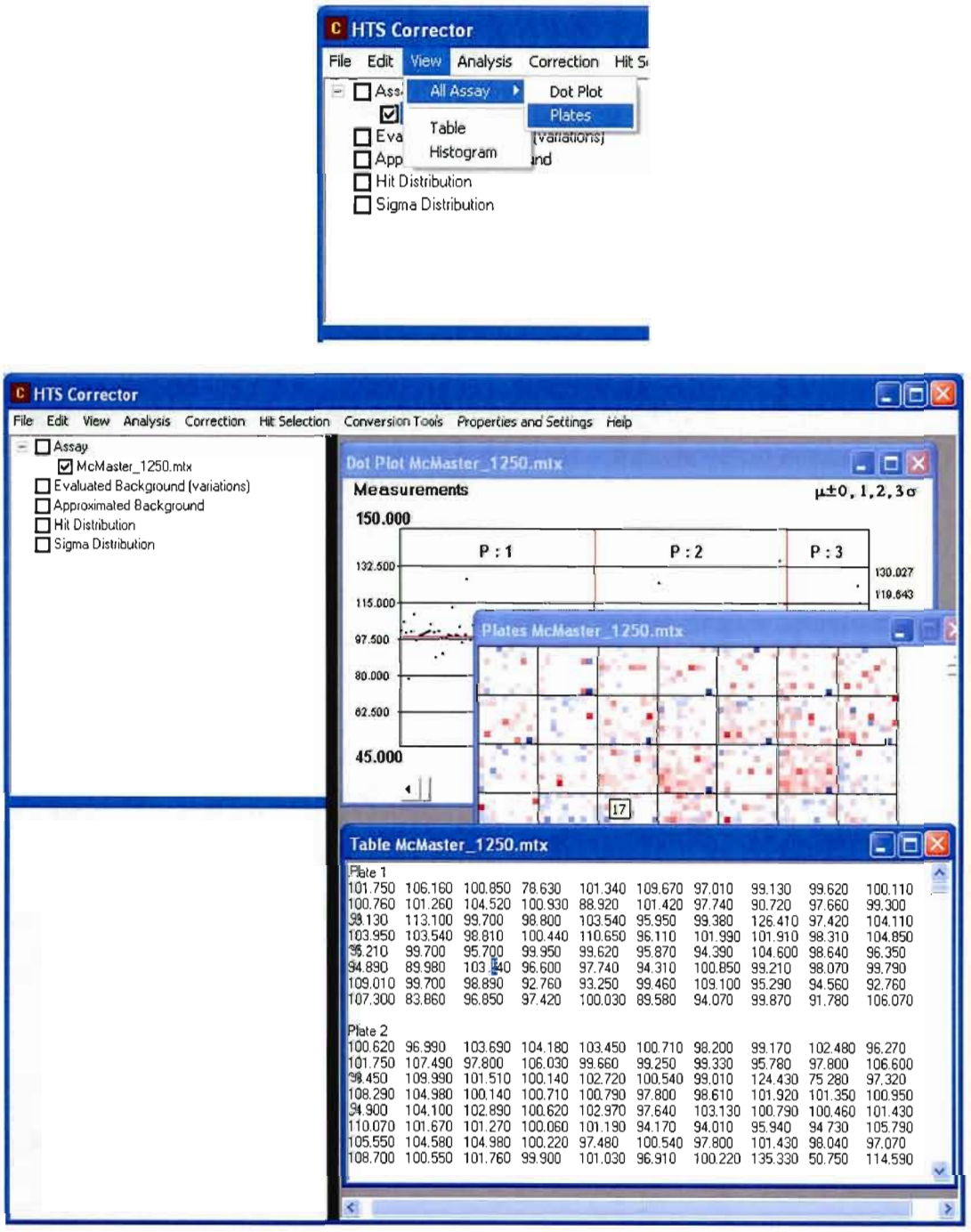


Figure 5.22 : Différentes options de visualisation pour les données d'une campagne HTS.

- Pour les données d'un plateau

  – Ouvrez un fichier contenant les données d'un arrière-plan (« background »), d'un arrière-plan approché, d'une distribution de « hits » ou d'une distribution des seuils de sélection des « hits »;

  – « Double-cliquez » sur le nom du fichier dans l'arborescence pour obtenir une visualisation en couleur des valeurs du plateau;

  ou

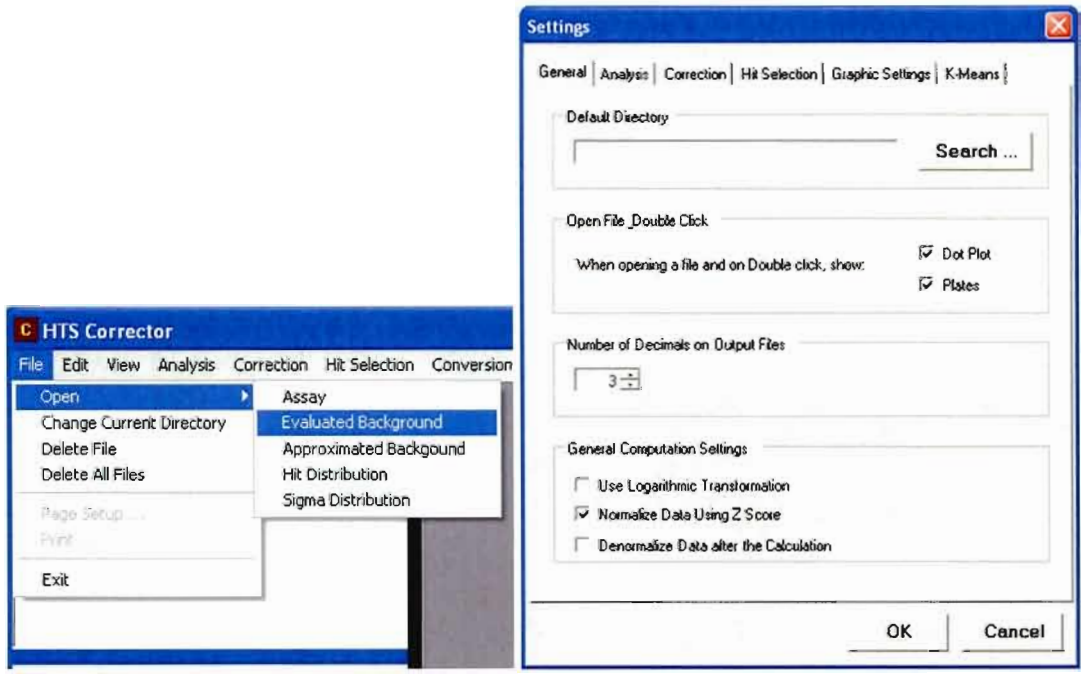
  – Sélectionnez le fichier dans l'arborescence;

  – Sélectionnez l'option ***View*** => ***Table*** pour visualiser les données numériques du plateau;

  – Sélectionnez l'option ***View*** => ***Histogram*** pour visualiser les histogrammes des valeurs moyennes des lignes et des colonnes du plateau.

Le graphique choisi (nuage de points (« dot plot »), plateaux du criblage, tables numériques ou histogrammes) sera affiché;

Changez les paramètres de visualisation dans le panneau inferieur gauche (facultatif).



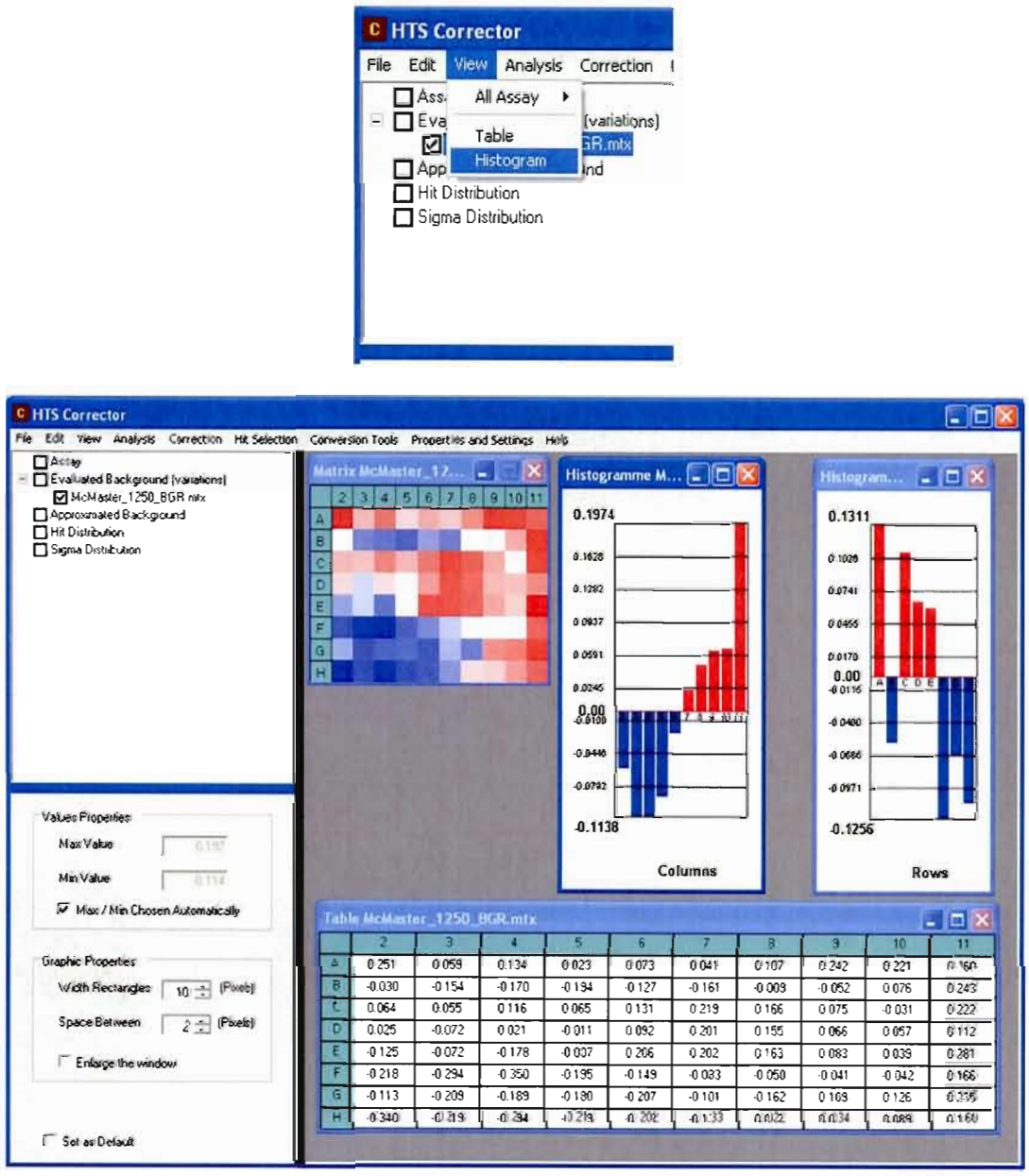


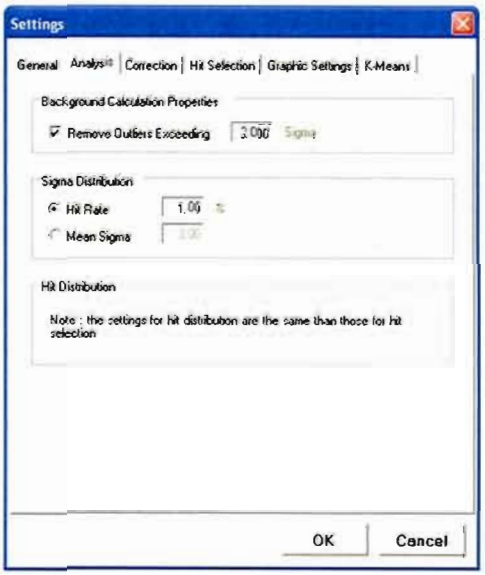
Figure 5.23 : Différentes options de visualisation pour les données d'un plateau.

Paramètres et propriétés :

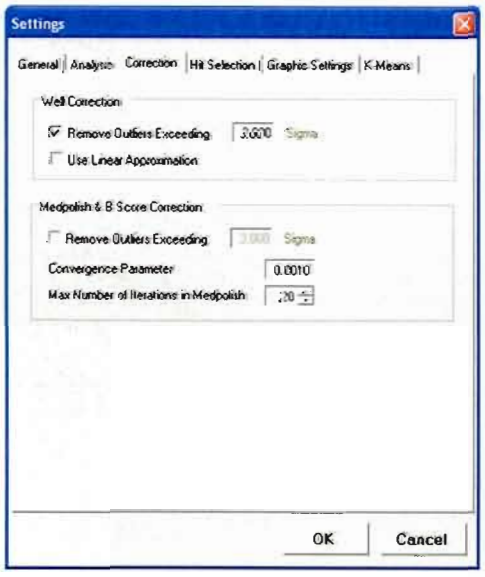
- Sélectionnez l'option ***Properties and Settings => Properties and Settings***;
- Choisissez l'onglet approprié;
- Choisissez les paramètres voulus.

Onglet ***General***

Onglet ***Analysis***

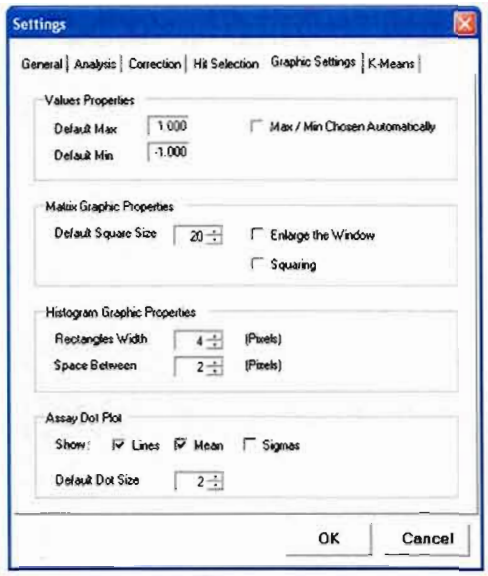


Onglet ***Correction***

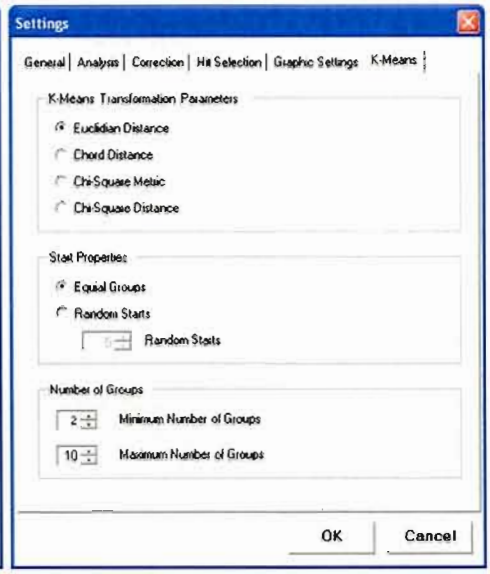


Onglet ***Hit Selection***

Onglet ***Graphical Settings***



Onglet ***K-Means***

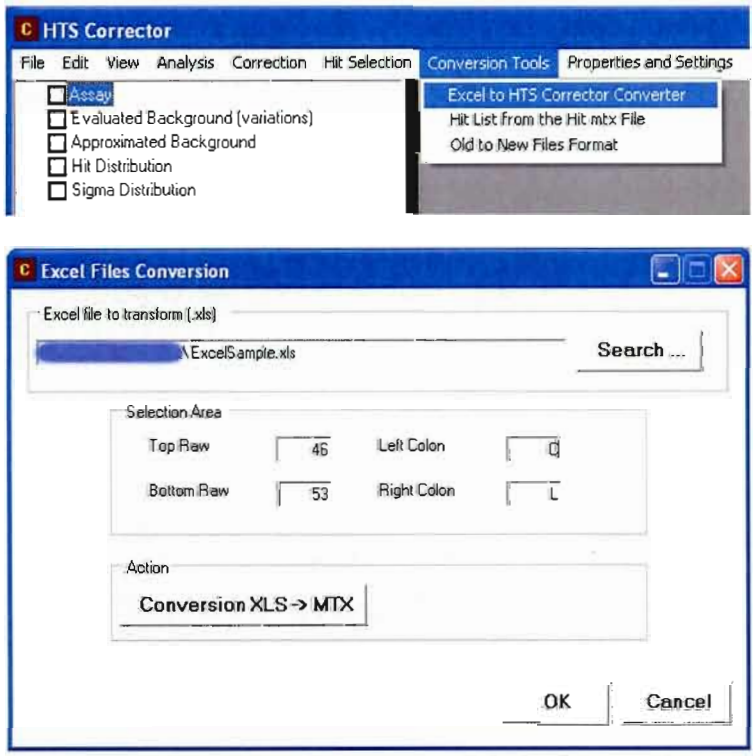


Conversion du format Excel au format « HTS Corrector » (.mtx) :

- Sélectionnez l'option ***Conversion Tools => Excel to HTS Corrector Converter;***
- Sélectionnez le fichier Excel à convertir (voir formats de fichiers);
- Sélectionnez la zone de la feuille Excel que vous voulez traiter;
- « Cliquez » sur le bouton ***Conversion XLS -> MTX***;
- Sauvegardez le nouveau fichier;
- Répétez l'opération pour d'autres fichiers si nécessaire.

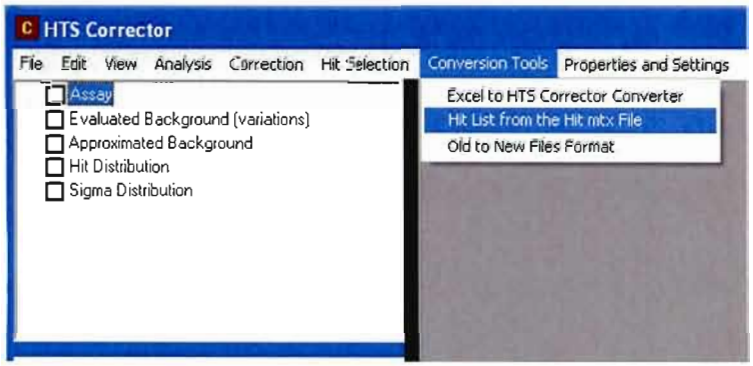
L'outil de conversion analysera toutes les feuilles du fichier Excel, en faisant correspondre chaque feuille à un plateau. Le nom de chaque plateau correspondra au nom de la feuille.

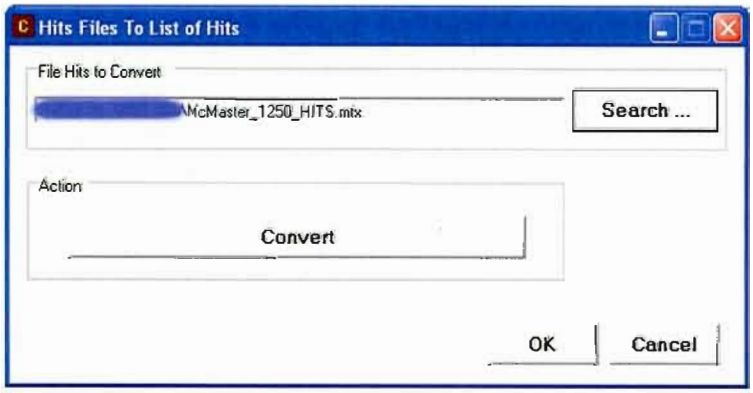
Contrôlez que le fichier Excel soit conforme au format, notamment qu'il ne comporte que des données numériques non vides dans la zone sélectionnée, et aucune feuille vide.

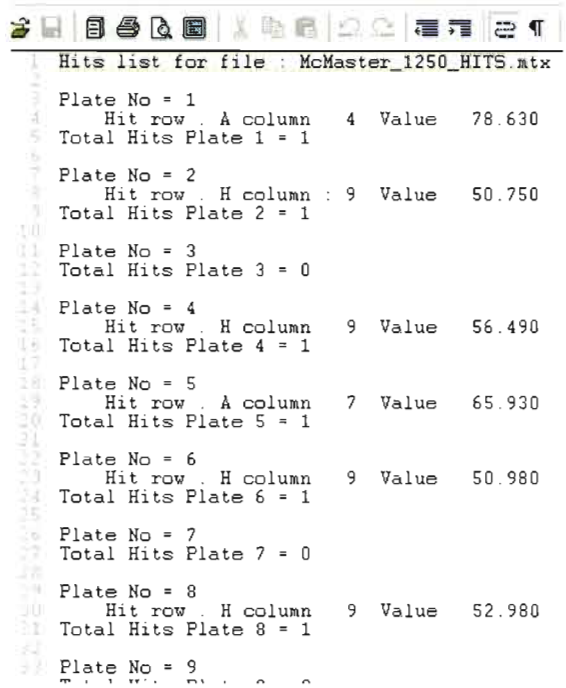


Obtenir la liste des « hits » :

- Sélectionnez l'option ***Conversion Tools => Hit list from the Hit mtx file;***
- Sélectionnez le fichier des « hits » à convertir;
- « Cliquez » sur le bouton ***Convert***;
- Sauvegardez le nouveau fichier;
- Répétez l'opération pour d'autres fichiers si nécessaire;
- Vous pouvez ouvrir le fichier avec un éditeur de texte.







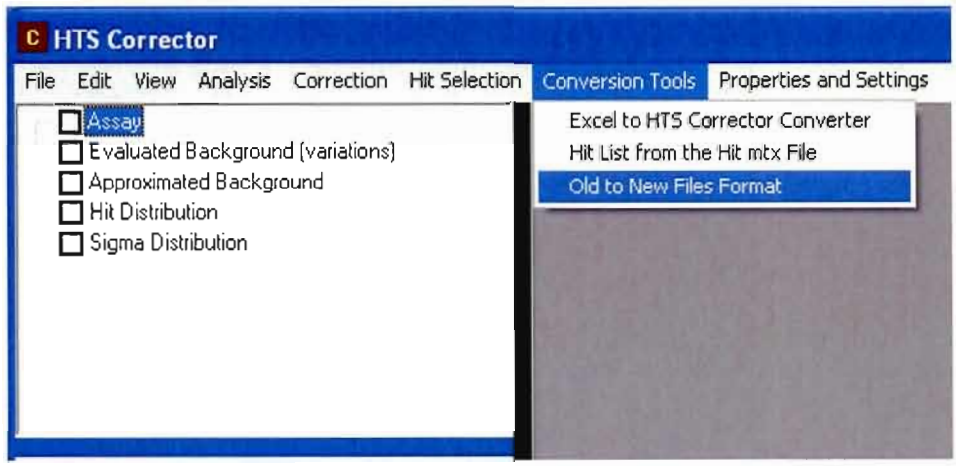
```
Hits list for file : McMaster_1250_HITS.mtx

Plate No = 1
    Hit row : A column   4  Value   78.630
Total Hits Plate 1 = 1

Plate No = 2
    Hit row : H column : 9  Value   50.750
Total Hits Plate 2 = 1

Plate No = 3
Total Hits Plate 3 = 0

Plate No = 4
    Hit row : H column   9  Value   56.490
Total Hits Plate 4 = 1

Plate No = 5
    Hit row : A column   7  Value   65.930
Total Hits Plate 5 = 1

Plate No = 6
    Hit row : H column   9  Value   50.980
Total Hits Plate 6 = 1

Plate No = 7
Total Hits Plate 7 = 0

Plate No = 8
    Hit row : H column   9  Value   52.980
Total Hits Plate 8 = 1

Plate No = 9
```

Figure 5.24 : Fichier résultat pour une transformation d'un fichier de « hits » en une liste de « hits ».

Conversion des fichiers ayant l'ancien format au nouveau format :

- Sélectionnez l'option ***Conversion Tools => Old to New Files Format***;
- Sélectionnez le fichier à convertir;
- Le nouveau fichier sera enregistré dans le même répertoire et aura le même nom que l'ancien fichier. Seule l'extension changera (*.mtx*).

Pour des exemples de fichiers, voir le paragraphe *Format de fichiers* plus haut.



Prochains développements et étapes prévues :

- Permettre au logiciel d'accepter des fichiers avec des données manquantes dans des puits;
- Améliorer les outils de conversion;
- Proposer un manuel de l'utilisateur complet;
- Finaliser les procédures de test unitaires et fonctionnels;
- Corriger quelques bogues connus, notamment dans les affichages graphiques;
- Intégrer une fonctionnalité de mise à jour automatique;
- Procéder à l'internationalisation du logiciel. Proposer une version française et préparer le logiciel pour qu'il soit facilement traduit dans d'autres langues (fonctionnalités prévues à cet effet dans le langage C#);
- Procéder à des tests en conditions réelles pour recueillir les rétroactions (« feedback ») des utilisateurs;
- Intégrer de nouvelles fonctionnalités selon les demandes des utilisateurs.

CONCLUSION

Dans ce mémoire nous avons présenté différentes approches informatiques et statistiques pouvant aider lors du processus de recherche de médicaments par criblage à haut débit. Nos recherches se sont concentrées sur la problématique du traitement des erreurs systématiques apparaissant lors d'une campagne HTS. Nous avons défini leurs causes les plus probables et présenté des méthodes pouvant les détecter et les corriger. La nature intrinsèque du problème limitant les solutions matérielles, nous avons privilégié la voie de l'analyse statistique des données. Nos travaux ont permis de comparer diverses solutions présentées dans la littérature et utilisées dans l'industrie.

Nos études nous ont conduit au développement d'une nouvelle approche pour la correction des données d'un criblage HTS. L'idée principale consiste en un traitement du criblage dans sa totalité, et non par plateaux. Nous avons appelé cette méthode : *correction par puits* (ou « well correction »). Nous avons également étudié et développé des méthodes de recherche des « hits » par des approches de groupement (« clustering »). Tous nos tests ont été effectués sur des données simulées permettant de varier de manière contrôlée les paramètres des expériences. Nous avons aussi procédé à des tests sur des données de criblages réels.

Toutes les méthodes étudiées, qu'elles aient été proposées par notre équipe ou par d'autres auteurs, ont été intégrées dans le logiciel « HTS Corrector ». Ce logiciel permet une meilleure analyse des campagnes HTS et apporte une aide considérable aux praticiens.

La comparaison entre les différentes méthodes a montré que la méthode de correction par puits était plus performante que ses concurrentes, et ce dans diverses situations. Nous pensons qu'elle peut être appliquée dans la plupart des cas, avec de très bons résultats. D'autres méthodes, considérées dans ce mémoire, ont montré des problèmes dans certaines circonstances. Des approches de recherche des « hits » par des méthodes de groupement, ont aussi été étudiées dans ce mémoire. Leurs résultats étaient, cependant, moins probants. Leur efficacité, quoique parfois supérieure à la méthode de recherche par seuil prédéfini, dépendait fortement du type de données analysées, notamment de la taille des plateaux. Néanmoins, il reste possible de les améliorer, notamment en appliquant d'autres algorithmes de groupement non étudiés dans ce mémoire.

Si les solutions informatiques, et statistiques, ne sont pas la seule approche à privilégier, elles permettent de traiter des quantités de données qu'il serait impossible d'analyser par des méthodes manuelles. Elles présentent aussi l'avantage de pouvoir visualiser des problèmes qui auraient pu échapper même au praticien averti, et de minimiser les erreurs qui pourraient être ajoutées lors de la manipulation des données. L'utilisation d'outils graphiques intégrés permet de détecter visuellement, et rapidement, des patrons dans les données analysées. Ils indiquent souvent des erreurs, ou des problèmes lors des procédures de mesures, qui seraient indétectables par une étude visuelle des valeurs numériques.

Le développement de nouveaux algorithmes efficaces demeure une priorité, que ce soit pour la sélection des « hits » et / ou la détection et la correction des erreurs dans les données de criblages à haut débit. Nous pensons qu'il est primordial, pour leur validation, de mettre en place des procédures de tests standardisées. La création de jeux de données simulés, facilement paramétrables et correspondant le plus possible aux cas réels, nous semble être une étape importante pour l'avancement de la recherche. Une procédure de comparaison, acceptée par l'industrie, permettrait de classifier les méthodes, actuelles ou futures, et fixer ainsi des degrés de qualité acceptables et reconnus. Lors de la recherche de nouveaux composés pouvant mener au développement de nouveaux médicaments, on pourrait exiger à toute méthode utilisée de s'y conformer pour d'être acceptée.

Le développement d'outils informatiques répondant aux besoins et exigences du domaine nous semble une nécessité. La priorité devant être mise, bien sûr, sur les méthodes offertes, mais surtout, à notre avis, sur la facilité d'emploi et sur la robustesse de leur conception. Ces derniers points nous semblant indispensables à l'acceptation, et donc à l'utilisation effective du produit. Nous espérons avoir fait un premier pas dans cette direction lors du développement de « HTS Corrector ».

> « *Utilisation d'outils informatiques peut réduire les coût de recherche d'un médicament de 50%* » (Traduction libre de Geldenhuys *et al*, 2006).

GLOSSAIRE (FRANÇAIS / ANGLAIS)

Campagne de criblage HTS (« assay ») : Procédure de criblage à haut débit effectuée sur un nombre important de données, généralement au moins quelques milliers de plateaux de composés. Une campagne peut durer plusieurs mois. Généralement une campagne de criblage est effectuée par l'analyse complète d'une ou plusieurs chimiothèques contre une cible thérapeutique particulière. Dans ce mémoire nous utilisons aussi les formulations suivantes : analyse HTS, criblage HTS, campagne HTS. Définition complémentaire : système biochimique ou biologique contrôlé expérimentalement dont on mesure les perturbations.

Cible thérapeutique (« target ») : Récepteur (enzyme, lipide, acide nucléique ou autres molécules) dont on veut modifier le régulateur moléculaire.

Composés actifs (ou prometteurs) (« hits ») : Composés se démarquant des autres composés d'un criblage. Possèdent une activité dont la mesure est sensiblement différente des autres composés. Seront probablement choisis pour des études supplémentaires (criblage secondaire). Par commodité dans ce mémoire on utilise le terme « hits ».

Criblage (« screen ») : Procédure de criblage compatible avec la procédure HTS.

Criblage homogène : Criblage nécessitant seulement le mélange des composés et un temps d'incubation avant la lecture des résultats.

Criblage non homogène : Criblage nécessitant une séparation et / ou un nettoyage des composés avant la lecture des résultats.

Criblage primaire (« primary screen ») : Criblage HTS utilisé pour identifier les composés ayant une activité moléculaire ou cellulaire prometteuse parmi une large collection. Fait partie des étapes préliminaires lors du processus de recherche de médicaments. Permet de faire un premier tri grossier parmi les composés testés.

Criblage secondaire (« secondary screen ») : Criblage utilisé pour mesurer quantitativement l'activité des composés choisis lors du criblage primaire.

Dose-réponse (DR) ou concentration-effet (CE) : Test de plusieurs concentrations en vue de générer une courbe sigmoïdale typique de l'activité de la cible thérapeutique par rapport au composé testé.

Échantillon (« sample ») : La substance testée contre une cible thérapeutique dans une campagne HTS.

Erreurs aléatoires (« random errors ») : Erreurs imprévisibles se produisant lors d'une mesure. Elles ont tendance à s'annuler dans des échantillons de grande taille. Elles sont intrinsèques à tout processus de mesure.

Erreurs systématiques (« systematic errors ») : Erreurs ayant une tendance orientée dans un même sens. Se produisent de manière plus ou moins constante tout au long d'un processus. Souvent dues à des erreurs de procédures ou à des défauts des appareils de mesure.

Faux négatif (« false negatif ») : Composé actif dont le signal est trop faible pour être détecté lors du criblage. Possible perte d'un composé pouvant mener à un médicament potentiel.

Faux positif (« false positif ») : Composé inactif présentant des résultats pouvant l'identifier comme actif lors du criblage. Pourra être éliminé lors des procédures subséquentes.

« Hit » confirmé (« confirmed hit ») : Composé montrant une activité reproductible par rapport à un seuil prédéfini. Sera souvent considéré comme candidat pour une tête de série (« lead »).

« Hit » initial (« initial hit ») : Composé identifié lors d'un criblage primaire ayant une activité apparente par rapport à un seuil prédéfini. Appelé aussi « hit » non confirmé.

HTS : « High-Throughput Screening », criblage à haut débit. Méthode d'analyse de composés chimiques par une sélection (criblage) de composés présentant des propriétés intéressantes pour le domaine d'application. Le qualificatif de « haut débit » est donné car la méthode permet de traiter, rapidement, des quantités importantes de composés (pouvant avoisiner 100'000 composés par jour dans le cas du Ultra HTS).

In silico **:** Utilisé pour qualifier les études faites par l'entremise d'un outil informatique. Vient de la composition des processeurs faits essentiellement de silicium.

In vitro **:** Utilisé pour qualifier les études faites essentiellement sur des cultures cellulaires ou des composants de la cellule. Les tests sont faits dans des éprouvettes.

In vivo **:** Utilisé pour qualifier les études faites sur des organismes vivants « complets », par exemple des souris ou des êtres humains.

Optimisation d'une tête de série (« lead optimisation ») : Processus d'amélioration des propriétés d'un composé tête de série. Deviendra un candidat au développement de médicaments.

Pourcentage de contrôle (« percent of control ») : Mesure qualitative de l'activité d'un composé. Résultat typique d'un criblage HTS. Pourcentage de l'activité d'un composé en présence d'une concentration du composé testé par rapport à l'activité sans la présence du composé testé.

Tête de série (« lead ») : « hit » ayant des propriétés de biocompatibilité (« druggability », « drug-like ») correspondant aux standards reconnus pour les composés pouvant mener au développement d'un médicament. Peut devenir un candidat pour des études précliniques.

RÉFÉRENCES

Brideau C., Gunter B., Pikounis W., Pajni N. et Liaw A. (2003) "Improved Statistical Methods for Hit Selection in High-Throughput Screening." *J. Biomol. Screen.*, **8**, 634-647.

Chen W.L. (2006) "Chemoinformatics: Past, Present, and Future." *J. Chem. Inf. Model.*, **46**, 2230-2255.

Dove A. (2003) "Screening for content – the evolution of high throughput." *Nature biotechnology*, **21**, 859-864.

Drew J. (2000) "*Quo vadis*, biotech? (Part 1)" *Drug Discovery Today*, **5**, 547-553.

Drew J. (2001) "*Quo vadis*, biotech? (Part 2)" *Drug Discovery Today*, **6**, 21-26.

Fassina G. (2006) "High Throughput Screening of Combinatorial Libraries. Survey of Applications of Combinatorial Technologies." *Training Course Presentation*, on line.

Fomenko I., Durst M. et Balaban D. (2006) "Robust regression for high throughput drug screening." *Computer Methods and Programs in Biomedicine*, **82**, 31-37.

Fox S., Farr-Jones S., Sopchak L., Boggs A., Wang Nicely H., Khoury R. et Biros M. (2006) "High-Throughput Screening: Update on Practices and Success." *J. Biomol. Screen.*, **11**, 864-869.

Gagarin A., Kevorkov D., Makarenkov V. et Zentilli P. (2006) "Comparison of two methods for detecting and correcting systematic error in high-throughput screening data." *In Data Science and Classification*, IFCS 2006, 241-249.

Gagarin A., Makarenkov V. et Zentilli P. (2006) "Using Clustering Techniques to Improve Hit Selection in High-Throughput Screening." *J. Biomol. Screen.*, **11**, 903-914.

Garneau P. et Zentilli P. (2006) "Validation d'un outil de correction de données HTS par comparaison à une étude *in-vivo*." *Rapport dans le cadre du cours BIF 7001, UQAM*, non publié.

Geldenhuys W.J., Gaasch K.E., Watson M., Allen D.D. et Van der Schyf C.J. (2006) "Optimizing the use of open-source software applications in drug discovery." *Drug Discovery Today*, **11**, 127-132.

Gribbon P. et Sewing A. (2005) "High-throughput drug discovery: what can we expect from HTS?" *Drug Discovery Today*, **10**, 17-22.

Gribbon P., Lyons R., Laflin P., Bradley J. Chambers C., Williams B.S., Keighley W. et Sewing A. (2005) "Evaluating Real-Life High-Throughput Screening Data." *J. Biomol. Screen.*, **10**, 99-107.

Gunter B., Brideau C., Pikounis B., Pajni N. et Liaw A. (2003) "Statistical and Graphical Methods for Quality Control Determination of High-Throughput Screening Data." *J. Biomol. Screen.*, **8**, 624-633.

Hamdan FF, Audet M, Garneau P, Pelletier J, Bouvier M. (2005) "High-throughput screening of G protein-coupled receptor antagonists using a bioluminescence resonance energy transfer 1-based beta-arrestin2 recruitment assay." *J. Biomol. Screen.*, **10**, 463-475.

Heuer C., Haenel T. et Prause B. (2003) "A novel approach for quality control and correction of HTS data based on artificial intelligence." *The Pharmaceutical Discovery & Development Report*, PharmaVentures Ltd.

Heyse S. (2002) "Comprehensive analysis of high-throughput screening data." *Proceedings of SPIE* 2002, **4626**, 535-547.

Inglese J., Auld D.S., Jadhav A., Johnson R.L., Simeonov A., Yasgar A., Zheng W. et Austin C.P. (2006) "Quantitative high-throughput screening: A titration-based approach that efficiently identifies biological activities in large chemical libraries." *Proc. Natl. Acad. Sci. U S A*, **103**, 11473-11478.

Kaul A. (2005) "The impact of sophisticated data analysis on the drug discovery process." *Business Briefing: Future Drug Discovery*, Online.

Kevorkov D. et Makarenkov V. (2005a) "Statistical analysis of systematic errors in high-throughput screening." *J. Biomol. Screen.*, **10**, 557-567.

Kevorkov D. et Makarenkov V. (2005b) "Quality control and data correction in high-throughput screening." *Proceedings of SFC* 2005, 159-163.

Kolossov E. et Lemon A. (2006) "Medicinal chemistry tools: making sense of HTS data." *European Journal of Medicinal Chemistry*, **41**, 166-175.

Lahana R. (1999) "How many leads from HTS?" *Drug Discovery Today*, **4**, 447-448.

Landro J.A., Taylor I.C.A. Stirtan W.G., Osterman D.G., Kristie J., Hunnicutt E.J., Rae P.M.M. et Sweetnam P.M. (2000) "HTS in the new millennium. The role of pharmacology and flexibility." *Journal of Pharmacological and Toxicological Methods*, **44**, 273-289.

Lipinski C.A., Lombardo F., Dominy B.W. et Feeney P.J. (2001) "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings." *Advanced Drug Delivery Reviews*, **46**, 3-26.

Lutz M.W., Menius A., Laskody R.G., Domanico P.L., Goetz A.S., Saussy D.L. et Rimele T. (1996) "Statistical Considerations in High Throughput Screening." *Network Science*, on-line.

Macarron R. (2006) "Critical review of the role of HTS in drug discovery." *Drug Discovery Today.*, **11**, 277-279.

MacQueen J. (1967) "Some methods for classification and analysis of multivariate observations." In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Volume I, Statistics. L. M. Le Cam and J. Neyman (Eds.), University of California Press.

Makarenkov V., Kevorkov D., Zentilli P., Gagarin A., Malo N. et Nadon R. (2006) "HTS-Corrector: software for the statistical analysis and correction of experimental high-throughput screening data." *Bioinformatics*, **22**, 1408-1409.

Makarenkov V., Zentilli P., Kevorkov D., Gagarin A., Malo N. et Nadon R. (2007) "An efficient method for the detection and elimination of systematic errors in high-throughput screening." *Bioinformatics*, **23**, 1648-1657.

Malo N., Hanley J.A., Cerquozzi S., Pelletier J. et Nadon R. (2006) "Statistical practice in high-throughput screening data analysis." *Nature Biotechnol.*, **24**, 167-175.

Nelson R.M. et Yingling J.D. (2004) "Introduction to High Throughput Screening for Drug Discovery" *IBC Life Sciences Conferences*, document de cours.

Parker C.N., Shamu C.E., Kraybill B., Austin C.P. et Bajorath J. (2006) "Measure, mine, model, and manipulate: the future for HTS and chemoinformatics?" *Drug Discovery Today*, **11**, 863-865.

Reinmann S., Lindemann M., Rinn B., Lefèvre O. et Heyse S. (2003) "Large-Scale, Comprehensive Quality Control and Analysis of High-Througput Screening Data" *European BioPharmaceutical Review*, Spring 2003.

Rippin M. (2003) "High Throughput Screening." *Tessella Support Services PLC*, on line.

Schnecke V. et Boström J. (2006) "Computational chemistry-driven decision making in lead generation." *Drug Discovery Today*, **11**, 43-50.

Sirois S., Hatzakis G., Wei D., Du Q. et Chou K-C. (2005) "Assessment of chemical libraries for their druggability." *Computational Biology and Chemistry*, **29**, 55-67.

Wu G. et Doberstein S.K. (2006) "HTS technologies in biopharmaceutical discovery" *Drug Discovery Today*, **11**, 718-724.

Woodward P.W., Williams C., Sewing A. et Beson N. (2006) "Improving the Design and Analysis of High-Throughput Screening Technology Comparison Experiments Using Statistical Modeling." *J. Biomol. Screen.*, **11**, 5-12.

Zhang J. H., Chung T.D.Y. et Oldenburg K.R. (1999) "A Simple Statistic Parameter for Use in Evaluation and Validation of High Throughput Screening Assays." *J. Biomol. Screen.*, **4**, 67-73.

Zhang J.H., Chung T.D.Y et Oldenburg,K.R. (2000) "Confirmation of primary active substances from high throughput screening of chemical and biological populations: a statistical approach and practical considerations." *J. Comb. Chem.*, **2**, 258-265.

"Building a High Throughput Screening Facility in an Academic Setting." *Harvard Medical School*, on line, (2002, updated 2003).

http://iccb.med.harvard.edu/screening/guidelines.htm#introduction. Tutoriel de HTS en anglais.

Statistiques et Programmation :

Benard, J., L. Bossavit, R. Medina et D. Williams. 2002. L'Extreme Programming - Avec deux études de cas. Eyrolles.

Concordet Didier. 2006. Notes de cours : Introduction à la statistique inférentielle. Unité de Biométrie. École Vétérinaire de Toulouse. Sur le site du cours.

Cwalina, Krzysztof, et Brad Abrams. 2006. Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET. Addison-Wesley.

Fowler, Martin. 2003. Patterns of Enterprise Application Architecture. Addison-Wesley.

Fowler, Martin et Kent Beck. 2005. Refactoring: Improving the Design of Existing Code. Pearson Education.

Gamma, Eric, Richard Helm, Ralph Johnson et John Vlissides. 1999. Design patterns. Catalogue des modèles de conception réutilisables. Vuibert informatique.

Hunt, Andrew et David Thomas. 2005. The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley.

Kerievsky, Joshua. 2005. Refactoring to Patterns. Addison-Wesley.

Kernighan, Brian W. 2001. La programmation en pratique. Vuibert.

Kernighan, Brian W. et Denis M. Ritchie. 2000. Le langage C, norme ANSI. Dunod.

Legendre, Pierre et Louis Legendre. 1998. Numerical Ecology. 2ème édition. Elsevier.

Legendre, Pierre. 2006. Notes de cours : Biostatistique I (BIO 2041) offert au niveau Baccalauréat à l'Université de Montréal. Sur le site du cours.

Leroux, Pierre. 1983. Algèbre linéaire : Une approche matricielle. Modulo.

Mattei, Aurelio. 1994. Inférence et décision statistiques. 2ème édition. Peter Lang.

McConnell, Steve. 2005. Tout sur le code : Pour concevoir du logiciel de qualité. 2ème édition. Dunod.

Moore, David S. et George P. McCabe. 2003. Introduction to the Practice of Statistics. 4ème édition. Freeman.

Newkirk, J. W. et Alexei A. Vorontsov. 2004. Test-Driven Development in Microsoft .NET. Microsoft Press.

Raymond, Eric S. 2004. The Art of UNIX Programming. Addison-Wesley.

Robinson, Simon et K. Scott Allen. 2003. C# Professionnel. 2ème édition. Wrox.

Shalloway, Alan et James R. Trott. 2002. Design Patterns par la pratique. Eyrolles.

Tukey, John W. 1977. Exploratory Data Analysis. Addison Wesley.

ANNEXE : CODE SOURCE

Note : A cause du volume important du code (plus de 16'000 lignes de code), nous ne présenterons que les parties qui nous semblent pertinentes pour ce mémoire, notamment les parties traitant des algorithmes étudiés. Les parties non pertinentes à l'intérieur d'une classe, tels que les procédures graphiques, les attributs, le constructeur, les « getters » et « setters » seront occultées.

Nous présentons dans la section suivante les méthodes d'évaluation de l'arrière-plan, d'approximation de l'arrière-plan, de soustraction de l'arrière-plan, de correction par puits, de correction par « median polish », de correction par B Score, de sélection des « hits » par seuil et par « clustering », de calcul de la distribution des « hits », et, finalement, des méthodes statistiques importantes. La plupart de nos méthodes utilisent une classe de gestion de matrices (MatricePlus) qui n'est pas présentée dans ce document. Son implémentation n'est pas importante, mais elle doit présenter les options de base du calcul matriciel utilisées dans le code.

La version actuelle du logiciel « HTS Corrector » (décembre 2006) est disponible gratuitement sur le site : http://www.info2.uqam.ca/~makarenv/HTS/home.php.

```
/* ************************************
 * HTS Corrector                      *
 * Pablo Zentilli                     *
 * UQAM - Montréal                    *
 *                                    *
 * Création : Juillet 2005            *
 * Dernière mise à jour : Septembre 2006   *
 ************************************ */

using System;
using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour le calcul du background d'une liste de matrices.
    /// Le constructeur prend la liste de matrices et les propriétés nécessaires au calcul.
    /// Pour lancer le calcul exécuter la méthode <code>Compute()</code>;
    /// </summary>
    public class BackgroundEval {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance le calcul du Background selon les paramètres passés au constructeur
        /// </summary>
        /// <returns>Une matrice des valeurs moyennes de l'arrière-plan correspondant
        /// à la liste de matrices originale.
        /// </returns>
        public MatricePlus Compute() {

            // Effectue les transformations nécessaires avant calcul.
            PreCalcul();

            // Calcul Proprement dit.
            // Attention, ne pas denormaliser, sinon la correction sera faussée.
            return BackgroundMoyen(this.copieListeMatrices, this.HIT_S);
        }

        /// <summary>
        /// Effectue les transformations nécessaires avant calcul, selon les
        /// paramètres passés au constructeur.
        /// </summary>
        private void PreCalcul() {
            // TODO : Factoriser en plusieurs morceaux réutilisables

            // Fait une transformation logarithmique des valeurs de la liste de matrices
            if (this.props.LogTransf) {
                this.copieListeMatrices =
                    Outils.LogAjustement(this.copieListeMatrices);
            }

            // Normalisation des plateaux (obligatoire)
            this.copieListeMatrices =
                Outils.Norm_Centree_Reduite_Listes(this.copieListeMatrices);

            // Suppresion des valeurs aberrantes
            this.HIT_S.MiseAZero(); // Nécessaire si remOutliers = false
            if (this.props.RemOutliersBack) {
                this.HIT_S = Outils.RemoveOutliersListe(this.copieListeMatrices,
                        this.props.SigmaRemOutliersBack);
                // Attention : Modifie copieListeMatrices,
                // Met des 0 à la place des outliers

                // Renormalise après correction
                this.copieListeMatrices =
                    Outils.Norm_Centree_Reduite_Listes(this.copieListeMatrices);
```



```
            }
        }

        /// <summary>
        /// Calcule les valeurs moyennes par puits d'un ensemble de plateaux (matrices)
        /// </summary>
        /// <param name="listeMatrices">
        /// Liste de matrices dont on veut calculer le Background
        /// </param>
        /// <param name="Hits">
        /// Matrice indiquant le nombre de valeurs modifiées (mises à zero) par puits
        /// car considérées comme des outliers.
        /// </param>
        /// <returns>La matrice des valeurs moyennes (background moyen)</returns>
        /// <remarks>listeMatrices doit être normalisés</remarks>
        /// <exception cref="System.DivideByZeroException">
        /// Levée si le nombre d'outliers pour un puits égale le nombre de plateaux
        /// </exception>
        private MatricePlus BackgroundMoyen(MatricePlus[] listeMatrices, MatricePlus Hits) {

            MatricePlus sommeBGR_Moyen = new MatricePlus(listeMatrices[0].NbLignes,
                listeMatrices[0].NbColonnes);
            sommeBGR_Moyen.MiseAZero();

            // Somme des valeurs de chaque puits de l'assay
            // listeMatrices.Length = nbPlates
            for (int p = 0; p < listeMatrices.Length; p++) {
                for (int l = 0; l < listeMatrices[0].NbLignes; l++) {
                    for (int c = 0; c < listeMatrices[0].NbColonnes; c++) {
                        sommeBGR_Moyen[l, c] += listeMatrices[p][l, c];
                    }
                }
            }

            // Divise la somme par puits par le nombre de plateaux
            // (moins le nombre de modifications faites sur ce puits)
            // Si listeMatrices.Length = 1, le background moyen est égal aux valeurs
            if (listeMatrices.Length != 1) {
                for (int l = 0; l < listeMatrices[0].NbLignes; l++) {
                    for (int c = 0; c < listeMatrices[0].NbColonnes; c++) {
                        sommeBGR_Moyen[l, c] /= (listeMatrices.Length - Hits[l, c]);
                        // Exception si (nbPlates - Hits[l, c]) = 0.
                        // Presque impossible.
                        // Il faudrait qu'il y ait autant de hits que de plateaux
                        // pour cet Assay, ce qui indiquerait de toutes façons un
                        // problème majeur lors de la prise de mesures
                    }
                }
            }

            return sommeBGR_Moyen;
        }
    }
}
```

```
/* ***************************************
* HTS Corrector                          *
* Pablo Zentilli                         *
* UQAM - Montréal                        *
*                                        *
* Création : Juillet 2005                *
* Dernière mise à jour : Septembre 2006  *
*************************************** */

using System;
using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour le calcul du background approché d'un background moyen.
    /// Le constructeur prend la matrice du background moyen.
    /// Pour lancer le calcul exécuter la méthode <code>Compute()</code>;
    /// </summary>
    public class BackgroundApprox {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance le calcul de l'approximation de la matrice de Background passée au
        /// constructeur
        /// </summary>
        /// <returns>
        /// Une matrice des valeurs approchées de l'arrière-plan correspondant
        /// à la matrice de Background.
        /// </returns>
        /// <exception cref="HTS_MatrixException">
        /// Levée par les méthodes de gestion et calculs des matrices
        /// </exception>
        public MatricePlus Compute() {
            // TODO : Voir si on peut simplifier

            // Zr = X*([Xt*X]^-1)*(Xt*Z)
            // X = matrice des coefficients du polynôme d'approximation
            // Z = Vecteur construit à partir de la matrice en entrée

            // Matrice X des coefficients
            Matrice X = Coefs(this.matrice);
            Matrice Zr = null;

            // Presque toutes les méthodes des matrices peuvent retourner "HTS_MatrixException"
            try {
                // Matrice X transposée
                Matrice Xt = Matrice.Transposer(X);

                // Création du vecteur Z des éléments de la matrice en entrée
                Matrice Z = new Matrice(this.matrice.NbElements, 1);
                int j = 0;
                for (int l = 0; l < this.matrice.NbLignes; l++) {
                    for (int c = 0; c < this.matrice.NbColonnes; c++) {
                        Z[j, 0] = this.matrice[l, c];
                        j++;
                    }
                }

                // Calcul de Zr = X*([Xt*X]^-1)*(Xt*Z)

                // Calcul de [Xt*X]
                Matrice XtX = Matrice.MultMatricielle(Xt, X);

                // Calcul de ([Xt*X]^-1)
                Matrice invXtX = Matrice.Inverser(XtX);
```



```
                // Si non inversible throw HTS_MatrixException

                // Calcul de (Xt*Z)
                Matrice XtZ = Matrice.MultMatricielle(Xt, Z);

                // Calcul de ([Xt*X]^-1)*(Xt*Z)
                Matrice Y = Matrice.MultMatricielle(invXtX, XtZ);

                // Calcul de X*([Xt*X]^-1)*(Xt*Z)
                Zr = Matrice.MultMatricielle(X, Y);

            } catch { // TODO: Gérer les exceptions qui peuvent être traitées à ce niveau
                throw;
            }

            // Fabriquer matrice de sortie
            MatricePlus Zres = new MatricePlus(this.matrice.NbLignes, this.matrice.NbColonnes);
            int j1 = 0;
            for (int l = 0; l < this.matrice.NbLignes; l++) {
                for (int c = 0; c < this.matrice.NbColonnes; c++) {
                    Zres[l, c] = Zr[j1, 0];
                    j1++;
                }
            }

            return Zres;
        }

        /// <summary>
        /// Création de la matrice des coefficients du polynôme d'approximation pour la
        /// matrice étudiée
        /// </summary>
        /// <param name="matrice">Matrice a approximer</param>
        /// <returns>Matrice des coefficients du polynôme d'approximation</returns>
        /// <remarks>
        /// La matrice des coefficients est toujours la même. Ne créer qu'une seule fois.
        /// </remarks>
        /// <exception cref="Exception">
        /// Si la matrice étudiée comporte moins de 15 éléments.
        /// Dans la pratique presque impossible (plateaux de 80 éléments au minimum)
        /// Attention lors des tests avec des valeurs arbitraires.
        /// </exception>

        private static Matrice Coefs(Matrice matrice) {

            if (matrice.NbElements < 15) {
                // TODO : FxCop Personnaliser l'exception
                throw new Exception("****************\n\n" +
                    "Coefs: Taille de la matrice insuffisante pour effectuer le calcul. " +
                    " Nb d'éléments minimum = 15." +
                    "\n\n**********");
            }

            Matrice coefs = new Matrice(matrice.NbElements, 15);
            int i = 0;
            for (int l = 0; l < matrice.NbLignes; l++) {
                for (int c = 0; c < matrice.NbColonnes; c++) {
                    coefs[i, 0] = 1;
                    coefs[i, 1] = (c + 1);
                    coefs[i, 2] = (l + 1);
                    coefs[i, 3] = (c + 1) * (c + 1);
                    coefs[i, 4] = (c + 1) * (l + 1);
                    coefs[i, 5] = (l + 1) * (l + 1);
                    coefs[i, 6] = (c + 1) * (c + 1) * (c + 1);
                    coefs[i, 7] = (c + 1) * (c + 1) * (l + 1);
                    coefs[i, 8] = (c + 1) * (l + 1) * (l + 1);
                    coefs[i, 9] = (l + 1) * (l + 1) * (l + 1);
```

```
146                     coefs[i, 10] = (c + 1) * (c + 1) * (c + 1) * (c + 1);
147                     coefs[i, 11] = (c + 1) * (c + 1) * (c + 1) * (l + 1);
148                     coefs[i, 12] = (c + 1) * (c + 1) * (l + 1) * (l + 1);
149                     coefs[i, 13] = (c + 1) * (l + 1) * (l + 1) * (l + 1);
150                     coefs[i, 14] = (l + 1) * (l + 1) * (l + 1) * (l + 1);
151                     i++;
152                 }
153             }
154
155             return coefs;
156         }
157     }
158 }
```

```
/* ************************************** *
* HTS Corrector                           *
* Pablo Zentilli                          *
* UQAM - Montréal                         *
*                                         *
* Création : Juillet 2005                 *
* Dernière mise à jour : Septembre 2006   *
* ************************************** */

using System;
using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour la soustraction du Background d'une liste de matrices.
    /// Le constructeur prend la liste de matrices, la matrice de background
    /// à enlever et les propriétés nécessaires au calcul.
    /// </summary>
    public class BackgroundRemove {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance la soustraction du Background selon les paramètres passés au
    constructeur
        /// </summary>
        /// <returns>
        /// Une liste de matrices correspondant à celle en entrée
        /// moins la valeur de l'arrière-plan choisi.
        /// </returns>
        public MatricePlus[] Compute() {

            // Effectue les transformations nécessaires avant calcul.
            PreCalcul();

            // Calcul Proprement dit.
            // ATTENTION : copieBackground doit être le BGR_Moyen_Normalise de
            // BackgroundEval c-à-d, les variations de background et non pas des
            // Valeurs directes (par ex 120.. etc.) sinon la correction sera faussée.
            for (int p = 0; p < this.copieListeMatrices.Length; p++) {
                for (int l = 0; l < this.copieListeMatrices[0].NbLignes; l++) {
                    for (int c = 0; c < this.copieListeMatrices[0].NbColonnes; c++) {
                        this.copieListeMatrices[p][l, c] -= this.copieBackground[l, c];
                    }
                }
            }

            // Effectue les transformations nécessaires après calcul.
            PostCalcul();

            return this.copieListeMatrices;
        }

        /// <summary>
        /// Effectue les transformations nécesaires avant calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PreCalcul() {
            // TODO : Factoriser en plusieurs morceaux réutilisables

            // Transformation logarithmique des valeurs de la liste de matrices
            if (this.props.LogTransf) {
                this.copieListeMatrices = Outils.LogAjustement(this.copieListeMatrices);
            }

            // Normalisation des matrices
            if (this.props.Normalize) {
```



```
                this.copieListeMatrices =
                    Outils.Norm_Centree_Reduite_Listes(this.copieListeMatrices);
            }
        }

        /// <summary>
        /// Effectue les transformations après calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PostCalcul() {

            // Denormalisation des résultats
            if (this.props.DeNormalize) {
                this.copieListeMatrices = Outils.DE_Normalisation_Liste(
                    this.copieListeMatrices, this.moyenne, this.sigma);
            }
        }
    }
}
```

```
/* ************************************
* HTS Corrector                       *
* Pablo Zentilli                      *
* UQAM - Montréal                     *
*                                     *
* Création : Octobre 2005             *
* Dernière mise à jour : Septembre 2006   *
************************************* */

using System;
using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour le calcul de la correction par puits (well correction) d'une
    /// liste de matrices.
    /// Le constructeur prend la liste de matrices et les propriétés nécessaires au
    calcul.
    /// Pour lancer le calcul exécuter la méthode <code>Compute()</code>;
    /// </summary>
    public class WellCorrection {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance le calcul de la correction par puits
        /// selon les paramètres passés au constructeur
        /// </summary>
        /// <returns>Une liste de matrices corrigée.</returns>
        public MatricePlus[] Compute() {

            // Effectue les transformations nécessaires avant calcul.
            PreCalcul();

            // Correction des wells
            MatricePlus well = new MatricePlus(1, this.copieListeMatrices.Length);
            double a = 0;
            double b = 0;

            for (int l = 0; l < this.copieListeMatrices[0].NbLignes; l++) {
                for (int c = 0; c < this.copieListeMatrices[0].NbColonnes; c++) {

                    // Pour chaque well
                    well.MiseAZero();

                    // Copie du well
                    for (int p = 0; p < this.copieListeMatrices.Length; p++) {
                        well[0, p] = this.copieListeMatrices[p][l, c];
                    }

                    if (this.props.LinearApprox) {
                        // TODO: Offrir les options par droite ou parabole
                        CoefsMoindresCarres(well, out a, out b);

                        // Correction yCorr = y - g(x) = y - (a + b * x)
                        for (int p = 0; p < this.copieListeMatrices.Length; p++) {
                            well[0, p] -= (a + b * p);
                        }
                    }

                    // Calcule la matrice normalisée pour chaque well
                    well = MatricePlus.Upgrade(Statistiques.Norm_Zmean_Sigma(well));

                    // Recopie le well sur la matrice
                    for (int p = 0; p < this.copieListeMatrices.Length; p++) {
                        this.copieListeMatrices[p][l, c] = well[0, p];
                    }
```



```
                }
            }

            // Effectue les transformations nécessaires après calcul.
            PostCalcul();

            return this.copieListeMatrices;
        }

        /// <summary>
        /// Effectue les transformations nécessaires avant calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PreCalcul() {
            // TODO : Factoriser en plusieurs morceaux réutilisables

            // Transformation logarithmique des valeurs de la liste de matrices
            if (this.props.LogTransf) {
                this.copieListeMatrices = Outils.LogAjustement(
                    this.copieListeMatrices);
            }

            // Normalisation (obligatoire, sinon on ne peut évaluer le background)
            this.copieListeMatrices = Outils.Norm_Centree_Reduite_Listes(
                this.copieListeMatrices);

            // Suppression des valeurs aberrantes
            if (this.props.RemOutliersWell) {
                Outils.RemoveOutliersListe(this.copieListeMatrices,
                    this.props.SigmaRemOutliersWell);
                // On ne mémorise pas le retour de la fonction.
                // Ce qui nous intéresse est this.copieListeMatrices
                // qui a été modifiée par RemoveOutliersListe.
                // Attn: Modifie copieListeMatrices, met des 0 à la place des outliers

                // Renormaliser après correction
                if (this.props.Normalize) {
                    this.copieListeMatrices =
                        Outils.Norm_Centree_Reduite_Listes(this.copieListeMatrices);
                }
            }
        }

        /// <summary>
        /// Effectue les transformations après calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PostCalcul() {

            // Denormalisation des résultats
            if (this.props.DeNormalize) {
                this.copieListeMatrices = Outils.DE_Normalisation_Liste(
                    this.copieListeMatrices, this.moyenne, this.sigma);
            }
        }

        /// <summary>
        /// Recherche les coefficients a et b d'une régression linéaire
        /// par la méthode des moindres carrés
        /// </summary>
        /// <param name="matrice">Vecteur de Valeurs (Matrice(1, n)</param>
        /// <param name="a">Premier coefficient de la régression linéaire</param>
        /// <param name="b">Deuxième coefficient de la régression linéaire</param>
        /// <remarks>
        /// Utilise la formulation suivante:
        /// Aprox(y) = g(x) = a + bx
        /// avec    a = (moy(x^2)*moy(y) - moy(x)*moy(x*y)) / (moy(x^2) - (moy(x))^2)
        /// et      b = (moy(x*y) - moy(x)*moy(y)) / (moy(x^2) - (moy(x))^2)
        /// ou      moy(x) = 1/n Som(x(i))
```

```
162         ///         moy(y) = 1/n Som(y(i))
163         ///         moy(x^2) = 1/n Som((x(i)^2))
164         ///         moy(x*y) = 1/n Som(x(i)*y(i))
165         /// Équivalent à :
166         ///     a = (Som((x(i)^2))*Som(y(i)) - Som(x(i))*Som(x(i)*y(i)))
167         ///         / (n*Som((x(i)^2)) - (Som(x(i)))^2)
168         /// et  b = (n*Som(x(i)*y(i)) - Som(x(i))*Som(y(i)))
169         ///         / (n*Som((x(i)^2)) - (Som(x(i)))^2)
170         ///     moy(x^2) = 1/n Som((x(i)^2))
171         ///     moy(x*y) = 1/n Som(x(i)*y(i))
172         /// </remarks>
173         static public void CoefsMoindresCarres(MatricePlus matrice, out double a,
174             out double b) {
175             // TODO : Changer public par private ou mettre dans les outils mathématiques
176
177             double Sx = 0;  // Som(x(i))
178             double Sy = 0;  // Som(y(i))
179             double Sxy = 0; // Som(x(i)*y(i))
180             double Sx2 = 0; // Som((x(i)^2)
181
182             for (int i = 0; i < matrice.NbColonnes; i++) {
183                 Sx += i;                    // Som(x(i))
184                 Sy += matrice[0, i];        // Som(y(i))
185                 Sxy += i * matrice[0, i];   // Som(x(i)*y(i))
186                 Sx2 += i * i;               // Som((x(i)^2)
187             }
188
189             a = (Sx2 * Sy - Sx * Sxy) / (matrice.NbColonnes * Sx2 - Sx * Sx);
190             b = (Sxy * matrice.NbColonnes - Sx * Sy) / (matrice.NbColonnes * Sx2 - Sx *
    Sx);
191         }
192     }
193 }
```

```
/* ********************************************
* HTS Corrector                               *
* Pablo Zentilli                              *
* UQAM - Montréal                             *
*                                             *
* Création : Octobre 2006                     *
* Dernière mise à jour : Décembre 2006        *
******************************************** */

using System;
using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour le calcul de la correction par la méthode "median polish"
    /// de Tukey (1977), d'une liste de matrices.
    /// Proposée par Brideau et al. (2003) et Malo et al. (2006)
    /// Le constructeur prend la liste de matrices et les propriétés nécessaires au calcul.
    /// Pour lancer le calcul exécuter la méthode <code>Compute()</code>;
    /// </summary>
    public class MedpolishCorrection {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance le calcul "median polish" selon les paramètres passés au constructeur
        /// </summary>
        /// <returns>Une liste de matrices corrigée.</returns>
        public MatricePlus[] Compute() {

            // Effectue les transformations nécessaires avant calcul.
            PreCalcul();

            // Calcul "median polish". Effectué pour chaque matrice
            for (int p = 0; p < this.copieListeMatrices.Length; p++) {
                this.copieListeMatrices[p] = MatricePlus.Upgrade(
                    Statistiques.Medpolish(this.copieListeMatrices[p],
                        this.props.ConvergenceMedpolish,
                        this.props.NbIterationsMedpolish));
            }

            // Effectue les transformations nécessaires après calcul.
            // Non pertinent.
            // PostCalcul();

            return this.copieListeMatrices;
        }

        /// <summary>
        /// Effectue les transformations nécessaires avant calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PreCalcul() {
            // TODO : Factoriser en plusieurs morceaux réutilisables

            // Transformation logarithmique des valeurs de la liste de matrices
            if (this.props.LogTransf) {
                this.copieListeMatrices = Outils.LogAjustement(
                    this.copieListeMatrices);
            }

            // Normalisation des matrices
            if (this.props.Normalize) {
                this.copieListeMatrices = Outils.Norm_Centree_Reduite_Listes(
                    this.copieListeMatrices);
            }
```



```

            // Suppression des valeurs aberrantes
            if (this.props.RemOutliersMed_BScore) {
                Outils.RemoveOutliersListe(this.copieListeMatrices,
                    this.props.SigmaRemOutliersMed_BScore);
                // On ne mémorise pas le retour de la fonction.
                // Ce qui nous intéresse est this.copieListeMatrices
                // qui a été modifiée par RemoveOutliersListe.
                // Attn: Modifie copieListeMatrices, met des 0 à la place des
                // outliers.

                // Renormaliser après correction
                if (this.props.Normalize) {
                    this.copieListeMatrices = Outils.Norm_Centree_Reduite_Listes(
                        this.copieListeMatrices);
                }
            }
        }

        /// <summary>
        /// Effectue les transformations après calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PostCalcul() {
            // Non Pertinent car on ne peut pas denormaliser ni « delogariser »
            // car ce sont des matrices de résidus.
            // TODO : Étudier
        }
    }
}
```

```
/* ************************************
* HTS Corrector                       *
* Pablo Zentilli                      *
* UQAM - Montréal                     *
*                                     *
* Création : Octobre 2006             *
* Dernière mise à jour : Décembre 2006*
************************************** */

using System;
using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour le calcul de la correction par la méthode B score d'une liste de matrices.
    /// Proposée par Brideau et al. (2003) et Malo et al. (2006)
    /// Le constructeur prend la liste de matrices et les propriétés nécessaires au calcul
    /// Pour lancer le calcul exécuter la méthode <code>Compute()</code>;
    /// </summary>
    public class BScoreCorrection {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance le calcul B score selon les paramètres passés au constructeur
        /// </summary>
        /// <returns>Une liste de matrices corrigée.</returns>
        public MatricePlus[] Compute() {

            // Effectue les transformations nécessaires avant calcul.
            PreCalcul();

            Matrice[] temp = Statistiques.BScore(this.copieListeMatrices,
                this.props.ConvergenceMedpolish, this.props.NbIterationsMedpolish);

            for (int p = 0; p < this.copieListeMatrices.Length; p++) {
                this.copieListeMatrices[p] = MatricePlus.Upgrade(temp[p]);
            }

            // Effectue les transformations nécessaires après calcul.
            // Non pertinent.
            // PostCalcul();

            return this.copieListeMatrices;
        }

        /// <summary>
        /// Effectue les transformations nécesaires avant calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PreCalcul() {
            // TODO : Factoriser en plusieurs morceaux réutilisables

            // Transformation logarithmique des valeurs de la liste de matrices
            if (this.props.LogTransf) {
                this.copieListeMatrices = Outils.LogAjustement(
                    this.copieListeMatrices);
            }

            // Normalisation des matrices
            if (this.props.Normalize) {
                this.copieListeMatrices = Outils.Norm_Centree_Reduite_Listes(
                    this.copieListeMatrices);
            }

```



```
            // Suppression des valeurs aberrantes
            if (this.props.RemOutliersMed_BScore) {

                Outils.RemoveOutliersListe(this.copieListeMatrices,
                    this.props.SigmaRemOutliersMed_BScore);
                // On ne mémorise pas le retour de la fonction.
                // Ce qui nous intéresse est this.copieListeMatrices
                // qui a été modifiée par RemoveOutliersListe.
                // Attn: Modifie copieListeMatrices, met des 0 à la place des
                // outliers.

                // Renormaliser après correction
                if (this.props.Normalize) {
                    this.copieListeMatrices = Outils.Norm_Centree_Reduite_Listes(
                        this.copieListeMatrices);
                }
            }
        }

        /// <summary>
        /// Effectue les transformations après calcul,
        /// selon les paramètres passés au constructeur.
        /// </summary>
        private void PostCalcul() {
            // Non Pertinent car on ne peut pas denormaliser ni « delogariser » car
            // ce sont des matrices de résidus.
            // TODO : Étudier
        }
    }
}
```

```
/* ****************************************
* HTS Corrector                           *
* Pablo Zentilli                          *
* UQAM - Montréal                         *
*                                         *
* Création : Juillet 2005                 *
* Dernière mise à jour : Décembre 2006    *
**************************************** */

using System;
using System.Collections;
using System.Collections.Generic;

using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour la recherche des "hits" d'une liste de matrices,
    /// selon les méthodes classiques de seuil (par plateau ou pour tout l'assay)
    /// et par les méthodes de clustering (Gagarin et al., 2006)
    /// Le constructeur prend la liste de matrices et les propriétés nécessaires au calcul
    .
    /// Pour lancer le calcul exécuter la méthode <code>Compute()</code>;
    /// </summary>
    public class HitSelection {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance la recherche des "hits" selon la méthode et les paramètres
        /// passés au constructeur
        /// </summary>
        /// <returns>
        /// Une liste des "hits" trouvés ("hit" = 1; non "hit" = 0).
        /// </returns>
        /// <exception cref="HTS_MethodesException">
        /// Levée si la méthode de recherche est inconnue.
        /// </exception>
        public MatricePlus[] Compute() {

            // Effectue les transformations nécessaires avant calcul.
            PreCalcul();

            switch (this.method) {
                case HitsSelMethods.SigmaAllAssay:
                    SigmaAllAssay();
                    break;

                case HitsSelMethods.SigmaByPlates:
                    SigmaByPlates();
                    break;

                case HitsSelMethods.KMeansClustering:
                    KMeansClustering();
                    break;

                case HitsSelMethods.SASDClustering:
                    SASDClustering();
                    break;

                case HitsSelMethods.AICDClustering:
                    AICDClustering();
                    break;

                default:
                    throw new HTS_MethodesException("Unknown Hit Selection method.");
                //break;
```



```
            }

            // this.hits a été modifié par la méthode de recherche choisie.
            return this.hits;
        }

        Méthodes de Sigma

        Méthodes de Clustering

        // Résultats des simulations de gagarin et al., 2006.

        /* Clustering Methods */
        //double pourcentHits[6] = {0.005, 0.01, 0.02, 0.03, 0.04, 0.05};

        //(STANDARD_NORMAL)
        //double seuilSigma[6] = {3.1, 3.0, 2.58, 2.3, 2.10, 1.94};
        //int intervalleRechercheK[6]   = { 5,   5, 100, 200, 200, 300};
        //int intervalleRechercheMin[6] = {50, 100, 100, 200, 200,   5};
        //int intervalleRechercheMax[6] = { 5,   5,   5,   5,   5,   5};

        //(LONG TAIL)
        //double seuilSigma[6] = {3.37, 3.12, 2.70, 2.38, 2.1, 1.95};
        //int intervalleRechercheK[6]   = { 5, 100, 100, 200, 300, 600};
        //int intervalleRechercheMin[6] = {50, 100, 100, 200, 300,   5};
        //int intervalleRechercheMax[6] = { 5,  10,   5,   5,   5,   5};
    }
}
```

```
/* ************************************
* HTS Corrector                       *
* Pablo Zentilli                      *
* UQAM - Montréal                     *
*                                     *
* Création : Juillet 2005             *
* Dernière mise à jour : Décembre 2006*
************************************** */

using System;
using HTS_Corrector.Auxiliaires;

namespace HTS_Corrector.Domaine {
    /// <summary>
    /// Classe pour le calcul de la distribution des hits
    /// Utilise les méthodes de recherche des hits de la classe HitSelection.
    /// Soit les méthodes classiques de seuil (par plateau ou pour tout l'assay)
    /// et par les méthodes de clustering (Gagarin et al., 2006)
    /// Le constructeur prend la liste de matrices et les propriétés nécessaires au calcul
    .
    /// Pour lancer le calcul exécuter la méthode <code>Compute()</code>;
    /// </summary>
    class HitDistribution {

        Attributs du calcul (Settings)

        Constructeurs

        /// <summary>
        /// Lance le calcul de la distribution des hits
        /// selon la méthode et les paramètres passés au constructeur
        /// </summary>
        /// <returns>
        /// Une matrice comptabilisant les hits par puits
        /// </returns>
        public MatricePlus Compute() {

            HitSelection hits = new HitSelection(
                this.listeMatrices,
                this.method,
                this.props);

            MatricePlus[] matricesHits = hits.Compute();

            MatricePlus hitsDistribution = new MatricePlus(
                this.listeMatrices[0].NbLignes, this.listeMatrices[0].NbColonnes);
            hitsDistribution.MiseAZero();

            for (int p = 0; p < this.listeMatrices.Length; p++) {
                for (int l = 0; l < this.listeMatrices[0].NbLignes; l++) {
                    for (int c = 0; c < this.listeMatrices[0].NbColonnes; c++) {
                        hitsDistribution[l, c] += matricesHits[p][l, c];
                    }
                }
            }

            return hitsDistribution;
        }
    }
}
```

```
/* ****************************************
* HTS Corrector                           *
* Pablo Zentilli                          *
* UQAM - Montréal                         *
*                                         *
* Création : Juillet 2005                 *
* Dernière mise à jour : Décembre 2006    *
**************************************** */

using System;
using System.IO;
using System.Collections;
using System.Globalization;

using System.Collections.Generic;

namespace HTS_Corrector.Auxiliaires {
    /// <summary>
    /// Diverses méthodes statistiques utiles pour le projet HTS
    /// </summary>
    /// <exception cref="HTS_StatistiquesException"></exception>
    /// <exception cref="DivideByZeroException"></exception>
    public static class Statistiques {

        #region Méthodes Statiques

        /// <summary> ...
        public static double Mediane(List<double> liste) ...

        /// <summary> ..
        public static double Mediane(Matrice matrice) ...

        /// <summary> ...
        public static double Moyenne(Matrice matrice) ...

        /// <summary> ...
        public static double Somme(Matrice matrice) ...

        /// <summary> ...
        static public double EcartType(Matrice matrice, double moyenne) ...

        /// <summary> ...
        static public Matrice Norm_Zmean_Sigma(Matrice matrice) ...

        /// <summary> ...
        public static Matrice DE_Normalisation(Matrice matriceNorm, double moyenne,
            double ecartType) ...

        /// <summary>
        /// Calcul de median polish (Tukey, 1977)
        /// Permet de corriger des erreurs systématiques sur des lignes et
        /// des colonnes de la matrice.
        /// </summary>
        /// <remarks>
        /// Selon nos simulations et tests cette méthode n'est pas très robuste
        /// si les erreurs sont présents sur plusieurs ligne et / ou colonnes
        /// Voir Makarenkov et al. (2006)
        /// </remarks>
        /// <param name="matrice">Matrice à corriger</param>
        /// <param name="convergence">
        /// Limite de convergence pour arrêter les itérations
        /// </param>
        /// <param name="nbIterations">Nombre maximal d'itérations</param>
        /// <returns>La matrice corrigée</returns>
        public static Matrice Medpolish(Matrice matrice, double convergence,
            int nbIterations) {

            Matrice erreursLignes;
            Matrice erreursColonnes;
```



```

            double medianneCalculee;

            Matrice reponse = Medpolish(matrice, convergence, nbIterations,
                out erreursLignes, out erreursColonnes, out medianneCalculee);

            return reponse;
        }

        /// <summary>
        /// Calcul de median polish (Tukey, 1977)
        /// Permet de corriger des erreurs systématiques sur des lignes et
        /// des colonnes de la matrice.
        /// Méthode complète donnant aussi les vecteurs des erreurs trouvés
        /// </summary>
        /// <remarks>
        /// Code adapté de la version medpolish proposée par le logiciel statistique R
        /// </remarks>
        /// <param name="matrice">Matrice à corriger</param>
        /// <param name="convergence">
        /// Limite de convergence pour arrêter les itérations
        /// </param>
        /// <param name="nbIterations">Nombre maximal d'itérations</param>
        /// <returns>La matrice corrigée</returns>
        /// <param name="erreursLignes">
        /// Vecteur des erreurs systématiques détectées sur les lignes
        /// </param>
        /// <param name="erreursColonnes">
        /// Vecteur des erreurs systématiques détectées sur les colonnes
        /// </param>
        /// <param name="medianneCalculee">Médiane calculée par l'algorithme</param>
        /// <returns>La matrice corrigée</returns>
        public static Matrice Medpolish(Matrice matrice, double convergence,
            int nbIterations, out Matrice erreursLignes, out Matrice erreursColonnes,
            out double medianneCalculee) {
            // TODO : Voir si on peut simplifier et refactoriser

            Matrice residus = matrice.Copy();

            List<double> listeTemp = new List<double>();

            Matrice vLignes = new Matrice(1, matrice.NbLignes);
            Matrice vColonnes = new Matrice(1, matrice.NbColonnes);
            Matrice rDelta = new Matrice(1, matrice.NbLignes);
            Matrice cDelta = new Matrice(1, matrice.NbColonnes);

            double oldSum = 0;
            double t = 0;
            double delta;
            double newSum;
            bool converge = false;

            int compteurIterations = 0;
            do {

                for (int l = 0; l < residus.NbLignes; l++) {

                    // Extraction des lignes
                    listeTemp.Clear();
                    for (int c = 0; c < residus.NbColonnes; c++) {
                        listeTemp.Add(residus[l, c]);
                    }

                    rDelta[0, l] = Mediane(listeTemp);
                }

                for (int l = 0; l < residus.NbLignes; l++) {
                    for (int c = 0; c < residus.NbColonnes; c++) {
                        residus[l, c] -= rDelta[0, l];
```

```
266                 }
267             }
268
269             for (int l = 0; l < residus.NbLignes; l++) {
270                 vLignes[0, l] += rDelta[0, l];
271             }
272
273             delta = Mediane(vColonnes);
274
275             for (int c = 0; c < residus.NbColonnes; c++) {
276                 vColonnes[0, c] -= delta;
277             }
278
279             t += delta;
280
281             for (int c = 0; c < residus.NbColonnes; c++) {
282
283                 // Extraction des colonnes
284                 listeTemp.Clear();
285                 for (int l = 0; l < residus.NbLignes; l++) {
286                     listeTemp.Add(residus[l, c]);
287                 }
288
289                 cDelta[0, c] = Mediane(listeTemp);
290             }
291
292             for (int l = 0; l < residus.NbLignes; l++) {
293                 for (int c = 0; c < residus.NbColonnes; c++) {
294                     residus[l, c] -= cDelta[0, c];
295                 }
296             }
297
298             for (int c = 0; c < residus.NbColonnes; c++) {
299                 vColonnes[0, c] += cDelta[0, c];
300             }
301
302             delta = Mediane(vLignes);
303
304             for (int l = 0; l < residus.NbLignes; l++) {
305                 vLignes[0, l] -= delta;
306             }
307
308             t += delta;
309
310             newSum = 0;
311             for (int l = 0; l < residus.NbLignes; l++) {
312                 for (int c = 0; c < residus.NbColonnes; c++) {
313                     newSum += Math.Abs(residus[l, c]);
314                 }
315             }
316
317             converge = ((Math.Abs(newSum) < convergence) ||
318                 (Math.Abs(newSum - oldSum) < (convergence * newSum)));
319
320             oldSum = newSum;
321             compteurIterations++;
322
323         } while ((!converge) && (compteurIterations < nbIterations));
324
325         if (!converge) { // des fois marche pas !!
326             throw new HTS_StatistiquesException("Median Polish do not converge" +
327                     "after " + nbIterations + " iterations");
328         }
329
330         erreursLignes = vLignes;
331         erreursColonnes = vColonnes;
332         medianneCalculee = t;
333
334         return residus;
```



```
335     }
336
337     /// <summary>
338     /// Calcul de Median Absolue Deviation (MAD)
339     /// Voir Malo et al. (2006)
340     /// </summary>
341     /// <remarks>Version proposant une distribution normale</remarks>
342     /// <param name="matrice">Matrice à traiter</param>
343     /// <returns>La valeur du MAD (normalisé)</returns>
344     public static double MAD_Normalise(Matrice matrice) {
345         return MAD(matrice) * 1.4826; // * 1.4826 "Normalise" le résultat
346     }
347
348     /// <summary>
349     /// Calcul de Median Absolue Deviation (MAD)
350     /// Voir Malo et al. (2006)
351     /// </summary>
352     /// <remarks>
353     /// Code adapté de la version MAD proposée par le logiciel statistique R
354     /// </remarks>
355     /// <param name="matrice">Matrice à traiter</param>
356     /// <returns>La valeur du MAD (non normalisé)</returns>
357     public static double MAD(Matrice matrice) {
358
359         List<double> listeTemp = new List<double>();
360
361         double medianePlateaux;
362         double resultat;
363
364         // Copie des valeurs dans un vecteur
365         for (int l = 0; l < matrice.NbLignes; l++) {
366             for (int c = 0; c < matrice.NbColonnes; c++) {
367                 listeTemp.Add(matrice[l, c]);
368             }
369         }
370
371         medianePlateaux = Mediane(listeTemp);
372
373         int compteur = 0;
374         for (int l = 0; l < matrice.NbLignes; l++) {
375             for (int c = 0; c < matrice.NbColonnes; c++) {
376                 listeTemp[compteur] = Math.Abs(matrice[l, c] - medianePlateaux);
377                 compteur++;
378             }
379         }
380
381         resultat = Mediane(listeTemp);
382
383         return resultat;
384     }
385
386     /// <summary>
387     /// Calcul du B score. Voir Brideau et al. (2003) Malo et al. (2006)
388     /// </summary>
389     /// <remarks>
390     /// Selon nos simulations et tests cette méthode n'est pas très robuste
391     /// si les erreurs sont présents sur plusieurs ligne et / ou colonnes
392     /// Voir Makarenkov et al. (2006)
393     /// </remarks>
394     /// <param name="listeMatrices">Liste de matrices à traiter</param>
395     /// <param name="convergence">
396     /// Limite de convergence pour arrêter les itérations de median polish
397     /// </param>
398     /// <param name="nbIterationsMedPolish">
399     /// Nombre maximal d'itérations pour median polish
400     /// </param>
401     /// <returns>La liste des matrices corrigées par B Score</returns>
402     public static Matrice[] BScore(Matrice[] listeMatrices, double convergence,
403         int nbIterationsMedPolish) {
```

```
404
405             Matrice[] liste = new Matrice[listeMatrices.Length];
406             Matrice[] resultat = Matrice.CopyListe(listeMatrices);
407
408             double mad;
409
410             for (int p = 0; p < resultat.Length; p++) {
411
412                 liste[p] = Medpolish(resultat[p], convergence, nbIterationsMedPolish);
413
414                 mad = MAD_Normalise(liste[p]);
415
416                 // Pour éviter les divisions par 0 ou très petit nombre
417                 if (mad < 0.0001) {
418                     mad = 0.0001;
419                 }
420
421                 for (int l = 0; l < resultat[0].NbLignes; l++) {
422                     for (int c = 0; c < resultat[0].NbColonnes; c++) {
423                         resultat[p][l, c] = liste[p][l, c] / mad;
424                     }
425                 }
426             }
427
428             return resultat;
429         }
430
431         #endregion
432
433         Méthodes non encore stables
571     }
572 }
```