Travaux pratiques : analyses bioinformatiques d'expériences de puces à ADN

Ce TP a pour objectif de vous familiariser avec un certain nombre de méthodes bioinformatiques, classiquement utilisées lors de l'analyse d'expériences de puces à ADN. Deux parties seront successivement abordées :

Partie 1 (jeudi 13 novembre) : Correction des biais expérimentaux.
Cette première partie du TP consiste à analyser un fichier GPR (résultat brut d'une expérience de puce à ADN) afin de corriger les éventuels biais expérimentaux et ainsi obtenir pour chaque gène, une mesure fiable de son expression.

Partie 2 (vendredi 14 novembre) : Recherche des gènes différentiellement exprimés et classification des profils d'expression des gènes.
Une fois les biais expérimentaux corrigés, la deuxième partie de ce TP consiste à exploiter les mesures d'expression obtenues, afin de répondre à deux grandes classes de questions biologiques : (*i*) quels sont les gènes différentiellement exprimés dans une condition par rapport à une autre ; et (*ii*) comment classer les gènes relativement à leurs mesures d'expression dans plusieurs expériences.

Partie 1 : Correction des biais expérimentaux

A l'issue de l'analyse d'image, le logiciel GENEPIX génère un fichier texte (fichier « GPR ») contenant pour chaque dépôt (ou « spot ») présent sur la puce et pour chaque fluorochromes (Cy5 et Cy3), le nom du gène correspondant, l'intensité de fluorescence, l'intensité du bruit de fond, ainsi que de nombreuses statistiques relatives à la taille du spot, l'homogénéité des dépôts, etc.

1. Utilisation de la librairie « marray »

Un grand nombre de biais expérimentaux peuvent être détectés par des méthodes simples. Par exemple, l'utilisation de représentations graphiques des données permet de faire un premier diagnostic de la qualité et de la cohérence des données. Pour cela, la librairie « *marray* » (logiciel R) propose différentes fonctions permettant de :

- Lire un fichier GPR ;
- Réaliser un diagnostic rapide de la qualité des données (bruit de fond, spots non conformes, ratios des intensités Cy5/Cy3 (Rouge/Vert) ;
- Corriger les biais expérimentaux (normalisation).

i) Lecture du fichier GPR

Nom du fichier à télécharger : « 20min_Beno_Cy5_DMSO_Cy3.gpr »
Fonction à utiliser : read.GenePix( )

La fonction « *read.GenePix* » permet la lecture des fichiers générés par le logiciel d'analyse d'image GENEPIX (extension des fichiers .gpr).

```
> rawdata <- read.GenePix(GprFile, …)
```

Questions *: Pour chacun des fluorochromes, extraire les valeurs des intensités globales (arguments « name.Rf » et « name.Gf »), des intensités du bruit de fond (arguments « name.Rb » et « name.Gb ») ainsi que les valeurs de « Flags » (argument « name.W »).*

L'objet R obtenu est de la classe « *marrayRaw* ». Il est composé de plusieurs vecteurs contenant les différentes informations initialement présentes dans le fichier GPR. Par exemple, le vecteur des intensités dans le rouge (ci-dessous) :

```
> rawdata@maRf
```

Note : Ce vecteur se manipule comme un vecteur classique sous R. Vous pouvez ainsi regarder la distribution des intensités dans le Rouge et dans Vert, en utilisant des fonctions telles « summary( ) » ou « hist( ) ».

ii) Diagnostic de la qualité des données

- ***Identification et élimination des « spots saturants »***

Les spots saturants sont des spots pour lesquels les intensités mesurées sont supérieures à 50000 (intensité maximale lue par le scanner). Ces spots ne sont pas corrects pour l'analyse, ils doivent être éliminés.

Une manière d'éliminer un spot consiste à changer sa valeur de « Flag ». Les valeurs de Flags de l'ensemble des spots sont regroupées dans le vecteur suivant :

```
> rawdata@maW
```

***Questions** : Y a-t-il des spots saturants dans votre fichier ? Si oui, combien ? Pour les exclure de l'analyse, changez les valeurs de Flags des spots saturants (valeur -10 par exemple).*

- ***Répartition spatiale du bruit de fond sur la lame***

Pour rechercher des biais expérimentaux, il est possible de visualiser la répartition spatiale des données sur la lame. Par exemple, la représentation des intensités du bruit de fond dans chacun des canaux R et G permet de détecter certains biais spatiaux (zones où le bruit de fond n'est pas homogène).

Fonction à utiliser : image( ) (argument « xvar = »)

```
> image(rawdata, xvar = « maRb », …)
```

***Questions** : Regardez la répartition sur la lame du bruit de fond associé au fluorochrome Rouge (« maRb »), puis celle du bruit de fond associé au fluorochrome Vert (« maGb »). Que remarquez-vous ? Refaites les deux mêmes représentations, mais cette fois en éliminant les spots avec une valeur de Flag inférieure à 0. Que remarquez-vous ?*

- ***Répartition spatiale des différentes valeurs de « Flag » sur la lame***

Nous avons vu plus haut que les spots non conformes sont identifiés *via* une valeur de Flag négative. Un spot pouvant être qualifié de « non conforme » pour différentes raisons (taille trop petite, intensité trop faible ou saturante, décision de l'expérimentateur, etc.), différentes valeurs sont contenues dans le vecteur de Flags « rawdata@maW ».
Par exemple :

- Une valeur de Flag à -100 signifie que le spot a été annoté comme non conforme par la personne qui a fait l'analyse d'image.
- Une valeur de Flag à -50 signifie « NOT FOUND », les intensités mesurées au niveau de la zone d'hybridation sont trop faibles.
- Une valeur de Flag à -75 correspond à des spots vides.

***Questions** : Identifiez toutes les valeurs de Flag possible, et décomptez le nombre de spots alloués à chacune d'entre elles. Dans un deuxième temps, regardez la répartition de ces spots sur la lame. Est-ce qu'un tel résultat vous semble cohérent ?*

La partie suivante consiste à corriger les biais expérimentaux. **Avant d'aborder les questions suivantes, veillez à bien retirer de l'analyse tous les spots non conformes (Flags < 0)**. Pour cela, il suffit de remplacer toutes les mesures d'intensité des spots non-conformes par le symbole « NA » :

```
> rawdata@maRf[rawdata@maW < 0] <- NA
> rawdata@maGf[rawdata@maW < 0] <- NA
```

iii) Correction des biais expérimentaux

○ *Correction du bruit de fond*

La correction du bruit de fond est souvent réalisée par une soustraction des intensités de bruit de fond (« *rawdata@maRb* » et « *rawdata@maGb* ») aux intensités globales (« *rawdata@maRf* » et « *rawdata@maGf* »). Toutefois cette méthode de correction est de plus en plus contestée principalement à cause du risque de générer des valeurs de ratios R/G très importantes dans le cas des intensités faibles (division par une valeur proche de 0). **Aussi, les analyses suivantes seront réalisées sans soustraction du bruit de fond**.

Attention, par défaut les fonctions de normalisation implémentées dans la librairie « *marray* » réalisent une soustraction du bruit de fond. Pour éviter toute mauvaise surprise, les intensités de bruit de fond seront préalablement mises à 0 :

```
> rawdata@maRb[] <- 0
> rawdata@maGb[] <- 0
```

○ *Rapport relatif entre les intensités mesurées dans le Rouge et dans le Vert*

La représentation « MA plot » consiste à représenter les données issues d'une expérience de puce à ADN sous la forme d'un nuage où chaque spot est représenté par un point dont l'abscisse est sa mesure moyenne des intensités mesurées dans le Rouge et le Vert (A = (log2(G) + log2(R)) / 2) et l'ordonnée le logarithme du ratio (M = log2(R/G)). Ce type de représentation permet de mettre en évidence l'existence d'éventuels biais systématiques entre les mesures des intensités R et G.

Fonction à utiliser : plot( ) (arguments legend.func = NULL, lines.func = NULL)

```
> plot(rawdata, …)
```

Attention, la fonction « plot( ) » est ici utilisée dans l'environnement de la librairie « marray ». Le graphique obtenu est ainsi différent de celui obtenu classiquement sous R.

***Questions** : Réalisez la représentation MA plot. Qu'observez-vous ? Existe-t-il un biais systématique entre les intensités mesurées dans le Rouge et dans le Vert ?*

Une autre représentation couramment utilisée est la représentation « BoxPlot » (ou boîte à moustache) qui permet de comparer des distributions des log2(R/G) associées aux différents blocs de la puce.

Fonction à utiliser : boxplot( ) (arguments legend.func = NULL, lines.func = NULL)

```
> boxplot(rawdata, …)
```

***Questions** : Réalisez la représentation BoxPlot. Que remarquez-vous ? Faut-il appliquer une correction entre les intensités R et G ? Quelles méthodes proposez-vous ?*

- ***Normalisation inter-canaux***

Avant d'interpréter le rapport R/G, il faut réaliser une normalisation entre les deux intensités de fluorescence. Actuellement, de très nombreuses méthodes de normalisation sont proposées dans la littérature. Trois méthodes différentes seront appliquées ici :

- Une méthode globale : médiane (« median »);
- Une méthode intensité dépendante : loess global (« loess ») ;
- Une méthode qui tient compte du biais spatial : loess par bloc (« printTipLoess »).

Fonction à utiliser : maNorm( ) (argument « norm »)

```
> maNorm(rawdata, norm = « loess », …)
```

***Questions** : Normalisez les intensités R et G en utilisant les trois méthodes (« median », « loess » et « printTipLoess »). Regarder la distribution des log2(R/G) avant normalisation et après normalisation. Que remarquez-vous ? Représentez les « MA plots » et « BoxPlot » correspondants après normalisation. Comparez-les à ceux que vous aviez obtenus avant normalisation. Quelles différences observez-vous entre les différentes méthodes de normalisation ?*

2. Travail personnel

Pour votre compte rendu de TP, réalisez la même analyse sur le fichier GPR suivant : « *40min_Beno_Cy5_DMSO_Cy3.gpr* » (à télécharger sur la page Web du cours). **Commentez vos résultats avec soin**.

Partie 2 : Recherche des gènes différentiellement exprimés et classification des profils d'expression des gènes

Une fois les expériences de puces à ADN réalisées et les biais expérimentaux corrigés, les valeurs d'expression obtenues pour chacun des gènes sont représentatives des mesures d'expression différentielle entre les deux conditions comparées.
Les méthodes d'analyse proposées dans cette deuxième partie de TP ont pour objectif de répondre aux questions biologiques pour lesquelles les expériences ont été réalisées. Deux grandes questions seront abordées : (*i*) quels sont les gènes différentiellement exprimés dans une condition par rapport à une autre ; et (*ii*) comment classer les gènes relativement à leurs mesures d'expression dans plusieurs expériences.

1. Recherche des gènes différentiellement exprimés

i) Répétitions de la même expérience

Dans cette première partie, nous allons rechercher les gènes dont l'expression est significativement différente entre les deux conditions hybridées sur la puce. D'un point de vue statistique, cela revient à comparer la valeur d'une variable d'intérêt entre deux échantillons, il est donc nécessaire de posséder plusieurs répétitions de la même expérience.

Nom du fichier à télécharger : Replicats_Benomyl_20min.txt

Ce fichier contient les mesures de log2(Ratio) (après corrections des biais expérimentaux) correspondant à 4 répétitions d'une même expérience de puce à ADN. Il s'agit du temps 20 minutes de la cinétique « Bénomyl » (voir cours).

ii) Utilisation de la librairie « samr »

Dans la littérature, il existe différentes méthodes statistiques qui permettent de rechercher des gènes différentiellement exprimés. L'une des plus populaire est la méthode SAM (Significance Analysis of Microarrays) qui permet : (1) d'assigner à chaque gène un score qui quantifie la différence d'expression entre les deux conditions et (2) de définir une règle de décision permettant, à partir du score précédent, de déclarer un gène différentiellement exprimé ou non. Dans ce TP, le programme SAM est utilisé *via* la librairie R « *samr* », en mode « *One class* » : pour chaque gènes on regarde si les mesures de log2(ratio) sont significativement différentes de 0.

- ***Lecture du fichier de données et création d'une liste au format « samr »***

Fonctions à utiliser : read.table( ) et list( )

```
> data <- read.table(« Replicats_Benomyl_20min.txt », …)

>  samdata <-  list(x        = data,        # Les données à traiter
                    y        = c(1,1,1,1),  # Vecteur d'identification
                    geneid   = ??,          # Unique ID
                    genenames = ??,         # Nom des gènes
                    logged2   = TRUE)       # Données en log2 ?
```

***Questions** : Importez le fichier de données sous R, puis créez une liste au format « samr » (ci-dessus). En consultant la documentation, expliquez le vecteur d'identification « c(1, 1, 1, 1) » ?*

- ***Lancement des permutations***

La méthode SAM effectue un ensemble de permutation de signe, pour chaque valeur de log2(Ratio) présentes dans le fichier de données initiales. Ces permutations permettent de définir une règle de décision (limite à partir de laquelle les expressions des gènes sont considérées comme différentes de 0) ainsi que d'estimer un taux de faux positifs (FDR = « *False Discovery Rate* »).

Fonction à utiliser : samr( ) (arguments « resp.type = », « nperms = »)

***Questions** : Lancer la procédure de permutation. Combien de permutations sont réalisées ? Pourquoi ce nombre ?*

- ***Choix d'une valeur de delta***

La méthode SAM repose sur le choix d'une valeur de « *delta* » (voir cours). Cette valeur définit la limite à partir de laquelle l'expression d'un gène est considérée comme différente de 0. Pour choisir une valeur de *delta* pertinente, il faut regarder l'estimation du taux de faux positif (FDR) associé : plus la valeur de *delta* est grande, plus le taux de FDR est faible.

Fonction à utiliser : samr.compute.delta.table( )

***Questions** : Déterminez la valeur delta pour laquelle le taux de faux positifs (FDR) de l'ordre de 1%. Combien de gènes sont alors sélectionnés comme ayant une expression différente de 0 ?*

- ***Sélection des gènes dont l'expression est significativement différente de 0***

Les permutations réalisées et la valeur de *delta* choisie, la recherche des gènes différentiellement exprimés peut-être lancée.

Fonctions à utiliser : samr.compute.siggenes.table( ) et samr.plot( )

***Questions** : Identifiez les gènes dont l'expression est significativement > 0 (gènes induits) et significativement < 0 (gènes réprimés). Représentez graphiquement ce résultat (« Scores attendus » contre « Scores observés »).*

iii) Analyse fonctionnelle des résultats

Toute analyse bioinformatique de données d'expression requiert une validation rigoureuse des résultats. Une manière d'appréhender la cohérence biologique de groupes de gènes tels que ceux que vous avez identifié précédemment consiste à réaliser une analyse fonctionnelle des résultats. Pour cela il existe différents outils sur Internet tel que FUNSPEC, accessible à l'adresse suivante : http://funspec.med.utoronto.ca/.

***Question** : Regardez le fonctionnement de FUNSPEC. A quoi correspondent les « p-values » associées à chaque fonction ? Commentez la cohérence biologique des résultats que vous avez obtenu à l'issue de l'analyse SAM.*

2. Classification des profils d'expression des gènes

Les méthodes de classification ont pour objectif de regrouper les gènes sur la base d'une ressemblance entre leurs profils d'expression. L'analyse des profils d'expression des gènes repose sur le principe suivant : les gènes dont les profils d'expression se ressemblent sont souvent de bons candidats pour être régulés par les mêmes facteurs ou intervenir dans les mêmes processus biologiques.

Nom du fichier à télécharger : Benomyl_AllTimes.txt

Ce fichier de données contient pour chaque gène sa mesure d'expression aux différents temps de la cinétique « Bénomyl » (voir cours).

i) Gestion des données manquantes

Lors de la réalisation d'une expérience de puce à ADN, il n'est pas rare que les mesures de log2(R/G) obtenues pour certains spots ne puissent être exploitées (spots saturants, etc.). Cela conduit à la présence de valeurs manquantes dans les fichiers de données finaux. Ces valeurs sont notées par le sympbole « NA » dans le fichier « *Benomyl_AllTimes.txt* ». Avant d'utiliser les méthodes de classification, il est nécessaire d'éliminer des valeurs manquantes.

Outil WEB à utiliser : GEPAS http://www.transcriptome.ens.fr/cgi-bin/gepas/preprocess

***Questions** : En utilisant GEPAS, filtrez les profils d'expression de gènes qui ont plus d'une valeur manquantes (sur les 6 points de mesure réalisés au cours de la cinétique). Combien de gènes vous reste-t-il ? Dans un deuxième temps, complétez les valeurs manquantes restantes par la méthode KNNimpute (choisir 30 voisins).*

ii) Analyse en Composantes Principales (ACP)

L'Analyse en Composantes Principales (ou ACP) est une approche statistique qui permet d'isoler les variables (ou les combinaisons de variables) qui expliquent au mieux les différences entre les expressions des gènes (voir cours). Au final, l'examen du plan factoriel permet de visualiser les associations entre les variables temps de la cinétique et d'identifier des groupes de gènes ayant pris les mêmes valeurs d'expression pour les mêmes variables.

Fonctions à utiliser : princomp( ) ou prcomp( )

***Questions** : Quel est le pourcentage de variabilité observé sur chacun des axes principaux ? Qu'est-ce que cela signifie ? Représentez les deux premiers axes. Comment est le nuage de points obtenu ? Où sont les gènes activés et réprimés ? Quels sont les avantages et inconvénients de l'ACP ?*

iii) Filtrage des profils d'expression invariants

Le filtrage des profils d'expression invariants a pour but de supprimer les gènes dont l'expression ne varie pas, ou peu, au cours de la cinétique. Ces gènes ne sont *a priori* pas impliqués dans le processus biologique testé, ils ne représentent donc que peu d'intérêt pour l'analyse.

Aussi, vous réaliserez les analyses suivantes uniquement sur les profils d'expression des gènes pour lesquelles l'analyse SAM au temps 20 minutes de la cinétique a été concluante (gènes induits).

iv) Calcul des distances entre les profils d'expression

Le calcul d'une distance entre les profils d'expression des gènes permet de quantifier la ressemblance entre ces profils. Plus la distance est petite, plus les profils sont proches. Différents types de distance peuvent être utilisées : distance euclidienne, distance dérivée de la corrélation.

Fonction à utiliser : dist( ) (argument « method = »)

***Questions** : Calculer la matrice de distance euclidienne entre tous les profils d'expression des gènes. Tracer la distribution des valeurs obtenues.*

v) Classification des profils d'expression

○ Classification hiérarchique

Le principe de la classification hiérarchique consiste à créer, à chaque étape, une partition obtenue en agrégeant deux à deux les gènes dont les profils d'expression sont les plus proches (voir cours). Au final, cette méthode de classification fournit une hiérarchie de partition, se présentant sous la forme d'arbres (également appelés dendogrammes).

Fonction à utiliser : hclust( ) (argument « method = ») et cutree( )

***Questions** : Quels sont les avantages et les inconvénients de cette méthode de classification ? Quelle est l'importance de la méthode d'agrégation (« average », « single » et « complete ») sur les résultats ?*

○ k-means

La méthode des k-means est une méthode de partitionnement qui permet de répartir un ensemble d'éléments (ici des gènes) en *k* classes homogènes (voir cours).

Fonction à utiliser : kmeans( )

***Questions** : Quels sont les avantages et les inconvénients de cette méthode de classification ? Comment choisir le nombre k de groupes ? Pour chaque groupe ainsi obtenu, représentez les profils d'expression des gènes leur appartenant.*

vi) Analyse fonctionnelle des groupes de gènes issus de la classification

Une nouvelle fois, aidez-vous du site FUNSPEC : http://funspec.med.utoronto.ca/.

Références :

- Données « bénomyl » :
Lucau-Danila A, Lelandais G, Kozovska Z, Tanty V, Delaveau T, Devaux F, Jacq C.
Early expression of yeast genes affected by chemical stress.
Mol Cell Biol. 2005 Mar;25(5):1860-8.
*- **Site Bioconductor :***
http://www.bioconductor.org/
*- **Méthodes de normalisation :***
Quackenbush J.
Microarray data normalization and transformation.
Nat Genet. 2002 Dec;32 Suppl:496-501. Review.
*- **Méthode SAM :***
Tusher VG, Tibshirani R, Chu G.
Significance analysis of microarrays applied to the ionizing radiation response.
Proc Natl Acad Sci U S A. 2001 Apr 24;98(9):5116-21. Epub 2001 Apr 17. Erratum in: Proc Natl Acad Sci U S A 2001 Aug 28;98(18):10515.
*- **Méthodes de classification :***
Quackenbush J.
Computational analysis of microarray data.
Nat Rev Genet. 2001 Jun;2(6):418-27. Review.
*- **Documentation librairie « samr » :***
http://lib.stat.cmu.edu/R/CRAN/doc/packages/samr.pdf