TP2

Justine Guégan - j.guegan-ihu@icm-institute.org
Guillaume Meurice - guillaume.meurice@gustaveroussy.fr

24 novembre 2016

Les données nécessaires à cette séance se trouvent sur le site web :

https://pf-bb.github.io/CentraleSupelec-R-genomics/TP2.html.

Objectif

L'objectif du TP est d'étudier la modulation de l'expression des gènes entre des échantillons de cancer du sein et des échantillons non tumoraux. Pour cela, des expériences de RNA-seq ont été réalisées.

Répondez aux questions dans un document Rmarkdown produisant un fichier **PDF** ou **HTML**.

Données

Les ARNm de 17 échantillons de tumeurs du sein de 3 types, HER2 positif (HER2), triple négatif (TNBC), non triple négatif (Non-TNBC), et 3 échantillons de sein normal (épithelium) ont été séquencés par Illumina HiSeq2000. Les données brutes sont disponibles sur le site **Sequence Reads Archive** (SRA), hebergé au NCBI, sous la référence SRP032789.

Ces données ont été utilisées dans les 2 articles suivants :

- J.Eswaran et al. Transcriptomic landscape of breast cancers through mRNA sequencing. Scientific Report (2012) article
- J.Eswaran et al. RNA sequencing of cancer reveals novel splicing alterations. Scientific Report (2013) article

Afin d'obetnir la table d'expression utilisée dans ce TP, les données brutes (fastq) ont préalablement été nettoyées, et alignées sur le génome de référence humain hg19 avec l'outil STAR. Puis la quantification a été faite avec l'outil `feature-count`. Cet outil génère un tableau avec en ligne les gènes, et en colonnes les individus. On y retrouve l'annotation des gènes ainsi que les valeurs d'expression brutes, aussi appelées "comptages".

Pré-requis

Ce TP nécessite le chargement de 2 packages :

- package pheatmap

```
install.packages("pheatmap")
```

- package RColorBrewser

```r
install.packages("RColorBrewser")
```

Analyse des données

1. Lecture des fichiers de données

> **Question** : Chargez en mémoire la table de comptage (fichier `counts.txt`) et la description des échantillons (fichier `annot_sample.txt`).
> Expliquez chacune des options utilisée.

> **Question** : Quelle est la classe des objets chargés et quelles en sont les dimensions ? Affichez un extrait de chaque objet.

2. Exploration des données

> **Question** : Indiquez le nombre d'échantillons par `condition`. Représenter cette répartition sous forme de pie chart.

> **Question** : A partir du fichier de comptages, créer un tableau contenant uniquement les données d'annotations des gènes et un tableau contenant uniquement les valeurs de comptages.
> Sur le tableau de comptage, modifier les noms des lignes pour qu'ils correspondent aux noms des gènes, et les noms de colonnes pour qu'ils correspondent aux noms d'échantillons (`sampleName`).

> **Question** : Pour chaque échantillon, indiquez les statistiques simples sur les valeurs de comptages (moyenne, médiane, min, max, 1er et 3ème quartiles). Quelle fonction permet de retourner très simplement toutes ces statistiques ? Que remarquez-vous ?

3. Normalisation

Un biais important en RNA-seq est la profondeur de séquençage de chaque échantillon, aussi appelée taille de la librairie.

> **Question** : Représentez en barplot la taille de librairie de chaque échantillon. Adaptez la couleur des barplot en fonction de la `condition` (1 couleur par `condition`). Existe-t-il un biais dans cette expérience ? Lequel ?

Pour corriger ce biais, nous allons normaliser les données. La taille de la librairie est souvent appelée, en RNA-seq, le *sizefactor*. Si les comptages des gènes non différentiellement exprimés sont, en moyenne, 2 fois plus grands dans un échantillon que dans un autre (car la librairie été séquencée 2 fois plus profond), le *sizefactor* du premier échantillon devrait être 2 fois plus grand que l'autre échantillon.

> **Question** : Chargez en mémoire le fichier de comptages normalisés `counts_normalized.txt`.

> **Question** : Afin d'observer l'effet de la normalisation sur les données, représenter sous forme de boxplot les données avant et après normalisation.
> Représentez ces 2 graphiques sur une même fenêtre.

> **Question** : Afin d'avoir une meilleure représentation des données, transformez les matrices de comptages non normalisés et normalisés en log2 (fonction `log2`). Représentez une nouvelle fois les 2 boxplots, après transformation des valeurs en log2.

En RNA-seq, il est courant d'avoir des valeurs de comptages à 0, ce qui correspond à des gènes non exprimés.

> **Question** : A partir de la matrice de comptage normalisés, affichez un graphique représentant le nombre de gènes ayant des comptages nuls en fonction des échantillons (utlisez le meme code couleur por les echantillons que precedemment) . Que remarquez-vous ?

> **Question** : Combien de gènes ne sont jamais exprimés chez tous les échantillons ? Créez une nouvelle matrice ne contenant pas ces gènes. Affichez les dimensions de cette nouvelle matrice. Transformez cette matrice en log2. Quelle est la valeur minimale par échantillon ? Pourquoi ? Afin d'éviter cette valeur, gênante pour la suite des analyses, nous allons ajouter un pseudocount de 1 avant de passer en log2. Crée cette nouvelle matrice.

Pour la suite du TP, nous travaillerons uniquement sur cette nouvelle matrice de comptages, c'est à dire la matrice de comptages normalisés, en log2, sans les gènes non exprimés chez tous les échantillons.

4. Analyse non supervisée

Nous allons visualiser la proximité relative des observations, grâce à une **Analyse en Composantes Principales**. Il s'agit d'une méthode d'analyse multivariée par réduction de dimension. Les composantes principales sont des combinaisons linéaires des variables. Elles ont pour contraintes de maximiser la variance entre les observations et d'être orthogonales entre elles. Le nombre de composantes est égal au rang de la matrice des données. On utilise la fonction `prcomp` de `R base`.

> **Question** :
>
> - Transposez votre matice de comptage grâce à la fonction `t()`.
> - Calculez les composantes grâce à la fonction `prcomp` avec les options `scale=TRUE` et `center=TRUE`. Combien y a-t-il de composantes ?
> - Représentez graphiquement les observations, c'est à dire les échantillons, en fonction des deux premières composantes, colorez les points en fonction de la colonne "condition" et changez la taille des points (paramètre `cex` de la fonction `plot`) en fonction du nombre de gènes non exprimés du tableau de données.
> - Ajoutez le noms des échantillons sur le graphique généré (fonction `text()`)
> - Ajoutez les lignes x=0 et y=0 en pointillé.
> - Interprétez cette ACP.

Il est possible de visualiser la proximité relative des observations, grâce à une autre méthode : **le clustering**. Cette methode consiste à calculer une distance entre les profils transcriptomique, puis à les regrouper de proche en proche. Il existe de nombreuses méthodes pour calculer une distance entre deux profils (distance Euclidienne, disante de Manahatan . . .) et de nombreuses méthode pour agréger les profils les plus semblables entre eux.

> **Question** :
>
> - Chargez les library `pheatmap` et `RColorBrewer` si ce n'est pas déjà fait.
> - Appliquez la fonction `dist` (avec les paramètres par défaut) à la matrice de comptage transposée. Quelle est la classe de l'objet généré ?
> - Créer un vector de couleurs de dégradé de bleu grâce à la commande suivante : `colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)`.
> - Générez une heatmap sample-to-sample grâce à la fonction `pheatmap`, et y ajouter une ligne d'annotations correspondant à l'annotation `condition`.
> - Interprétez ce clustering.