Stefano Caruso (stefano.caruso@inserm.fr) **5 octobre 2018**
Eric Letouzé (eric.letouze@inserm.fr)

M2 BI, Module approches méthodologiques en bioinformatique

TP : Analyse génomique intégrée des adénomes hépatocellulaires

Les adénomes hépatocellulaires (AHC) sont des tumeurs rares du foie, généralement bénignes mais qui peuvent progresser en carcinomes hépatocellulaires malins. L'objectif de ce TP est d'utiliser différents types de données génomiques (transcriptome, séquençage exome) pour identifier des groupes moléculaires homogènes de ces tumeurs et les mutations génétiques causales.

Vous disposez pour ce TP de :
- données de puce expression pour 31 AHC et 3 échantillons de foie normal
- tables des mutations somatiques identifiées par séquençage exome dans 40 AHC
- annotations cliniques pour 65 échantillons analysés sur puce expression et/ou par séquençage exome.

La première partie du TP sera consacrée à l'identification de groupes moléculaires homogènes d'après les profils transcriptomiques des tumeurs via diverses approches de classification non supervisée. La 2ème partie du TP sera consacrée à l'identification des gènes « drivers » de chaque groupe à l'aide des données exome. Finalement, nous reviendrons aux données d'expression pour identifier les gènes et pathways dérégulés au sein d'un groupe moléculaire.

Ce TP nécessite la librairie R *ConsensusClusterPlus*. Si ce n'est pas déjà fait, installez cette librairie avec la fonction *install.packages()*.

1) Classification non supervisée des AHC sur la base du transcriptome

1. Chargez le fichier « expression_matrix_31T_3N_samples.RData », qui contient les données d'expression (puce Affymetrix U133Plus2.0) de 31 AHC et 3 échantillons de foie normal. Vérifiez la taille de la matrice -> combien de sondes la puce utilisée contient-elle ?
load(), dim()

2. Sélectionnez les 500 sondes les plus variables dans le jeu de données, sur la base de l'écart-type.
apply(), sd(), sort(), tail()

3. Préparez une table *mat* contenant les données d'expression réduites aux 500 sondes les plus variantes. Vérifiez les dimensions de cette table. Il faut maintenant centrer les données, c'est à dire retirer à chaque ligne la valeur moyenne d'expression de la sonde. L'objectif de cette étape est de « casser » les corrélations naturelles dues aux différences de gamme d'expression des gènes, pour mieux voir les variations fines inter-échantillons.
apply(), mean()

4. Réalisez un clustering hiérarchique des échantillons. Vous pouvez tester différentes distances et méthodes de clustering. Combien de groupes d'AHC définiriez-vous en se basant sur ce clustering ?
dist(), hclust(), plot()
distance conseillée : « euclidean » ; méthode de clustering conseillée : « ward.D »

5. Pour évaluer la stabilité des groupes et définir la meilleure classification, nous allons maintenant réaliser un clustering consensus. Chargez la librairie *ConsensusClusterPlus* et générez les classifications consensus (*clusterAlg="hc", distance="euclidean", innerLinkage="ward.D", finalLinkage="ward.D"*) de 2 à 10 groupes (*nmax=10*) en prenant à chaque fois 80% des sondes (*pFeature=0.8*) et 80% des échantillons (*pItem=0.8*). La fonction génère des graphes donc il faut ouvrir un pdf avant de la lancer. Vous pouvez stocker les résultats renvoyés par la fonction dans un objet *consclust*.
pdf(), ConsensusClusterPlus(), dev.off()

6. Sur la base du clustering consensus, quel nombre de groupes vous semble le plus approprié ? Chargez la table d'annotations (« Clinical_annotations.RData »). Ajoutez-y une colonne « expGroup » contenant, pour les 34 échantillons analysés sur puce expression, le groupe que vous souhaitez assigner sur la base du clustering consensus Vous pouvez récupérer le groupe de chaque échantillon dans l'objet *consclust*. Par exemple, les groupes de la classification en *k* classes sont dans *consclust[[k]]$consensusClass*.

7. La table d'annotations contient le groupe moléculaire que nous avons assigné à chaque échantillon au laboratoire sur la base d'une analyse génomique intégrée (colonne « Molecular.group »). Combien y en a-t-il ? Est-ce que votre classification est concordante avec la nôtre ?
table(), chisq.test()

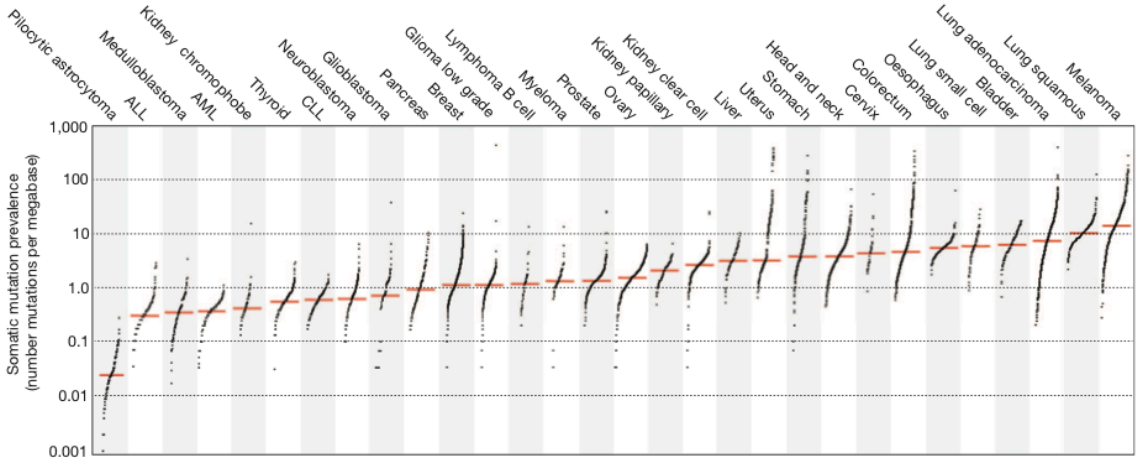
8. Réalisez une analyse en composantes principales et représentez les projections des échantillons sur les 2 premières composantes principales, avec un code couleur indiquant le groupe d'expression.
prcomp()

2) Identification du/des gène(s) driver(s) dans chaque groupe

Maintenant que nous avons défini des groupes moléculaires homogènes, nous allons utiliser les données de séquençage exome pour identifier les gènes drivers mutés de façon récurrente dans chaque groupe.

1. Le fichier « *WES_mutations.txt* » contient les mutations somatiques identifiées par séquençage de l'exome dans 40 AHC. Chargez cette table dans votre session.
read.delim()

2. Quel est le nombre médian, minimal et maximal de mutations par échantillon ? Sachant que le kit de capture exome permet de séquencer 50 Mb d'ADN, quel est le taux de moyen de mutations/Mb dans les AHC ? En comparaison avec les autres types de tumeurs humaines (ci-dessous), s'agit-il d'un type de tumeur avec beaucoup ou peu de mutations ?



3. Afin d'identifier les gènes drivers de chaque groupe, ajoutez une colonne « Group » à la table des mutations, indiquant à quel groupe moléculaire appartient l'échantillon portant la mutation (colonne « Sample.ID »). Vous pouvez pour cela utiliser les groupes fournis dans la table d'annotations (colonne « Molecular.Group »).
match()

4. Pouvez-vous maintenant définir le(s) gène(s) driver(s) impliqués dans chaque groupe ?
table(), sort(), tail()

5. Les mutations qui activent les oncogènes sont le plus souvent missense et affectent des hotspots fonctionnels. Les mutations qui inactivent les gènes suppresseurs de tumeurs sont souvent perte de fonction (missense, nonsense), distribués tout le long du gène, et affectent souvent les 2 copies du gènes (2 mutations dans la tumeur ou mutation + délétion). Dans quelles catégories classez-vous les gènes *CTNNB1* et *HNF1A* ? Oncogène ou suppresseur de tumeurs ?

3) Recherche du mécanisme oncogénique impliqué dans les UHCA
Le groupe UHCA (« unclassified HCA ») est le seul groupe pour lequel aucun gène driver muté de façon récurrente n'a pus être identifié par séquençage exome. Nous allons étudier les gènes différentiellement exprimés dans ces tumeurs pour tenter d'identifier le(s) pathway(s) oncogénique(s) impliqué(s).

1. Nous allons repartir de la matrice d'expression *exp* stockée dans l'objet « expression_matrix_31T_3N_samples.RData ». Chargez également la table «expression_array_annotations.RData » qui contient les annotations (gène cible) de chaque sonde de la puce expression. Pour limiter le nombre de tests à réaliser, réduisez ces tables aux gènes présentant un niveau d'expression suffisant (expression moyenne ≥3).

3. Utilisez la fonction *source()* pour lire le script « *Functions_for_supervised_analysis.R* » qui contient les fonctions *ExpCompare* et *enrichmentTest* nécessaires aux étapes suivantes.

4. Utilisez la fonction *ExpCompare* pour réalisez une analyse d'expression différentielle entre les 4 tumeurs du groupe UHCA (CHC603T, CHC605T, CHC950T et CHC1317T) et les échantillons de foie normal. La fonction reçoit en entrée la matrice d'expression (*exp*), les annotations de la puce (*featureTab*), un vecteur avec les noms des échantillons de référence (*refsamp*) et un autre avec les échantillons à tester (*gpsamp*).
ExpCompare()

5. Stockez les gènes significativement sur-exprimés (pval < 0.001, FC > 1) dans un vecteur *up* et les gènes significativement sous-exprimés (pval < 0.001, FC < 1) dans un vecteur *down*. Combien y a-t-il de gènes sur- et sous-exprimés ?
which(), length()

6. Pour mieux comprendre à quoi correspondent ces gènes, nous allons réaliser une analyse d'enrichissement. Il s'agit d'identifier des voies cellulaires enrichies parmi les gènes *up* et *down*. Chargez la liste des pathways cellulaires (gene sets) à tester (« gene_sets.Rdata »). Vous pouvez utiliser la fonction *enrichmentTest* pour identifier les pathways enrichis. Vous devez spécifier les gene sets à tester (argument *gene.*sets), les gènes d'intérêt (*up* ou *down*, argument *mylist*), et l'ensemble des gènes testés (c'est à dire l'ensemble des gènes représentés sur la puce, argument *possibleIds)*.
enrichmentTest()

7. Quel pathway est le plus significativement enrichi parmi les gènes surexprimés dans le sous-groupe de 4 tumeurs ? Quels gènes surexprimés appartiennent à ce pathway (colonne «corresponding elements in list ») ?

8. N'ayant pas trouvé d'altération récurrente dans les séquences codantes (séquençage exome), nous avons séquencé le génome complet de 3 tumeurs du groupe UHCA. Le séquençage génome complet permet notamment d'identifier des réarrangement structuraux (délétions, duplications, translocations chromosomiques ou inversions) non détectables par séquençage exome. La table ci-dessous indique les réarrangements identifiés dans les 3 UHCA. Quel gène est altéré dans les 3 tumeurs ? Est-ce cohérent avec votre analyse de pathway ?

Tumor ID	Rearrangement Type	Breakpoint 1 Chromosome	Breakpoint 1 Position	Breakpoint 1 Category	Breakpoint 1 Gene	Breakpoint 2 Chromosome	Breakpoint 2 Position	Breakpoint 2 Category	Breakpoint 2 Gene
CHC605T	Deletion	chr1	106563124	Intergenic Region		chr1	106651394	Intergenic Region	
CHC605T	Deletion	chr12	57849348	promoter	INHBE	chr12	57851031	promoter	GLI1
CHC950T	Deletion	chr12	57848774	Intron	INHBE	chr12	57850823	promoter	GLI1
CHC1317T	Deletion	chr1	51255587	Intron	FAF1	chr1	51449320	Intergenic Region	
CHC1317T	Deletion	chr6	21146116	Intron	CDKAL1	chr6	24665276	Intron	TDP2
CHC1317T	Deletion	chr10	133577232	Intergenic Region		chr10	133577358	Intergenic Region	
CHC1317T	Inversion	chr12	57845166	promote	INHBE	chr12	57853835	promoter	GLI1