**Amélie Roehrig (**[**amelie.roehrig@inserm.fr**](mailto:amelie.roehrig@inserm.fr)**) October 18th 2021**

**Théo Hirsch (**[**theo.hirsch@inserm.fr**](mailto:theo.hirsch@inserm.fr)**)**

**Integrative genomic analysis of hepatocellular adenomas**

Hepatocellular adenomas (HCA) are rare benign liver tumors, sometime at risk of malignant transformation. The aim of this training is to identify stable molecular subgroups and the main driver genes in these types of tumors using different genomic data (transcriptomic and whole exome sequencing data)

1. **Unsupervised transcriptomic classification of HCA.**

The first part of the practical will be focused on the identification of homogeneous molecular subgroups based on transcriptomic profiles using different unsupervised classification approaches.

For this purpose, you can use:

- expression data (Affymetrix) for 31 HCA and 3 normal liver tissue samples

- clinical annotations for 65 samples analyzed by Affimetrix and/or WES

This practical requires the R “ConsensusClusterPlus”, "ComplexHeatmap", “ggplot2” and fgsea libraries. If it is not already done, install this library with the install.packages() function or alternatively Biomanager using the following code:

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("ComplexHeatmap").

1. Load the file « expression\_matrix\_31T\_3N\_samples.RData » which contains expression data (Affymetrix U133Plus2.0) from 31 AHC and 3 normal liver samples. Check the size of the matrix -> how many probes does the chip contain?

*load(), dim()*

2. Select the 500 most variable probes in the data set, based on the standard deviation.

*apply(), sd()*

3. Prepare a table called “mat” containing the expression data reduced to the 500 most variant probes. Check the dimensions of this table. It is now necessary to center the data, i.e. remove the average expression value of the probe from each line. The aim of this step is to remove the differences in gene expression range between different probe sets, in order to highlight inter-sample variations.

*apply(), mean()*

4. Perform a hierarchical clustering of samples. You can test different distances and clustering linkage methods. How many groups of AHCs would you define based on this clustering?

*dist(), hclust(), plot()*

*recommended distance: "euclidean"; recommended clustering linkage method: "ward.D"*

5. To assess the stability of the groups and define the best classification, we will now perform a consensus clustering. Load the “ConsensusClusterPlus” library and generate consensus classifications (clusterAlg= "hc", distance= "euclidean", innerLinkage= "ward.D", finalLinkage= "ward.D") from 2 to 10 groups (maxk=10) taking 80% of the probes (pFeature=0.8) and 80% of the samples (pItem=0.8) each time. The function generates graphs so you have to open a pdf before launching it. You can save the results obtained by the function in a “consclust” object.

*pdf(), ConsensusClusterPlus(), dev.off()*

6. On the basis of consensus clustering, which number of groups do you think is most appropriate? Load the annotation table "Clinical\_annotations.RData". Add an "expGroup" column in the table containing (for the 34 samples with expression data) the group you want to assign based on consensus clustering. You can retrieve the group of each sample in the object consclust. For example, the groups of the classification in k classes are in *consclust[[k]]$consensusClass*.

*match()*

7. The annotation table contains the molecular group that we assigned to each sample in the laboratory based on an integrated genomic analysis ("Molecular.group" column). How many are there? Is your classification consistent with ours?

*table(), chisq.test()*

8. Create a heatmap using the “Heatmap” function from the R library “ComplexHeatmap”. To cluster the samples, use the dendogram you obtained above at point 5. Subsequently, create another heatmap by grouping the samples using the dendogram that comes out from the consensus cluster (*consclust[[k]]$consensusTree*). Save the 2 heatmaps as pdf files.

*Heatmap(cluster\_columns=…)*,

9. Now it’s time to add some annotation to your heatmap. For this purpose, starting from the annotation table you have to create a new data.frame for the samples in the heatmap (n=34) and containing the annotations you want to use (“Molecular.group”,“Pathological.diagnosis”).

What are your findings? Attention: sort samples in the new data frame as in the expression matrix.

<https://jokergoo.github.io/ComplexHeatmap-reference/book/heatmap-annotations.html>

*HeatmapAnnotation(), Heatmap(top\_annotation=..)*

10. Perform a principal component analysis (PCA) and represent the sample projections on the first 2 main components, with a color code indicating the expression group you defined at step 7 (“expGroup”). How much is the percentage of variance explained by the first component? Represents the percentage of variance explained by each component as barplot.

*prcomp(), summary(), barplot(),*

1. **Identification of the driver gene(s) in each group**

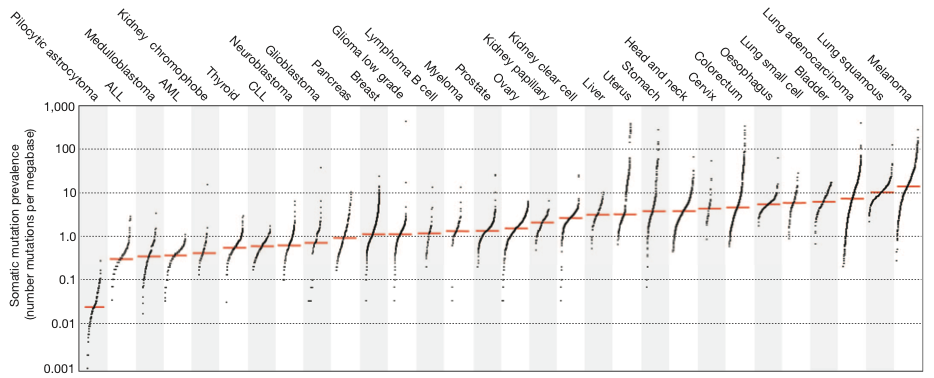
This part of the TP will be focused on identifying the "driver" genes of each group using the exome data.

In fact, now that we have defined homogeneous molecular groups, we will use the exome sequencing data to identify the mutated driver genes recurrently in each group.

1. The file "WES\_mutations.txt" contains the somatic mutations identified by exome sequencing in 40 HCA. Load this table into your session.

*read.delim()*

2. What is the median, minimum and maximum number of mutations per sample? Knowing that the exome capture kit allows to sequence 50 Mb of DNA, what is the median mutations/Mb rate in HCA? Compared to other human tumor types (below), is it a tumor type with many or few mutations?



3. To identify the driver genes of each group, add a "Group" column to the mutation table, indicating to which molecular group the sample with the mutation belongs ("Sample.ID" column). You can use the groups provided in the annotation table ("Molecular.Group" column).

*match()*

4. Can you now define the driver(s) involved in each group?

*table(), sort(), tail()*

5. The mutations that activate oncogenes are most often missense and affect functional hotspots. Mutations that inactivate tumor suppressor genes are often loss of function (missense, nonsense), distributed throughout the gene, and often affect both copies of the gene (2 mutations in the tumor or mutation + deletion). In which category do you classify the CTNNB1 and HNF1A genes? Oncogene or tumor suppressor?

**3) Analysis of the oncogenic mechanism involved in UHCA**

Now, we will return to expression data to identify deregulated genes and pathways within a molecular group.

The UHCA group ("unclassified HCA") is the only group for which no recurrently mutated driver gene could be identified by exome sequencing. We will study the differentially expressed genes in these tumors in order to identify potential oncogenic pathway(s) involved in this groups of tumors.

For this purpose you can use:

- expression data (Affymetrix) for 31 HCA and 3 normal liver tissue samples

- gene table containing information for each probe set (Affymetrix)

- clinical annotations for 65 samples analyzed by Affimetrix and/or WES

1. We will start from the expression matrix “exp” stored in the object "expression\_matrix\_31T\_31T\_3N\_samples.RData". To limit the number of tests to be performed, please reduce this matrix to genes with sufficient expression (mean expression ≥3).

Please reduce also the “gene\_table.RData” to the new selection of probe set.

2. Use the “limma” library to perform differential expression analysis between the tumors of the UHCA group and normal liver samples. How many samples do you have for each group? For differential gene expression analysis with limma you will need to create a data frame “df” (*data.frame()*) with the sample information in binary code (1= yes, 0 = no) as follows:

|  |  |  |
| --- | --- | --- |
|  | Normal | UHCA |
| Sample1 | 1 | 0 |
| Sample2 | 1 | 0 |
| Sample3 | 1 | 0 |
| Sample4 | 0 | 1 |
| Sample5 | 0 | 1 |
| Sample6 | 0 | 1 |
| Sample7 | 0 | 1 |

Then create a new expression matrix “exp.comp” containing only the selected samples with the same order as in the “df” data frame. Finally, fit the model by using the following code:

*fit <- lmFit(exp.comp, df)*

*cont.matrix <- makeContrasts(UHCA-Normal, levels=df)*

*fit2 <- contrasts.fit(fit, cont.matrix)*

*fit2 <- eBayes(fit2)*

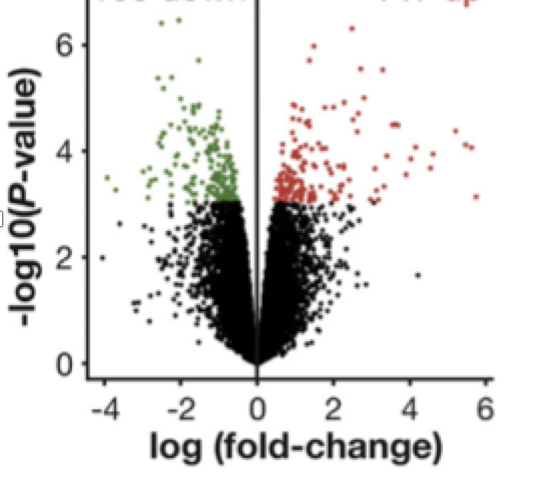
*options(digits=2)*

*RES=topTable(fit2, adjust="BH", n=nrow(exp.comp))*

Add a new column to the RES object corresponding to the Gene name (look at the gene\_table). Remove row without a Gene name and then save the new RES object as a tab delimited txt file.

*match(), write.table()*

Starting from the RES object, represent the differentially expressed genes as a volcano plot where in the axis of the abscissae is represented the logFC and in the ordinate the –log10(adjusted p-value). Try to color significant differentially expressed genes (adjusted p-value< 0.05) in red or green if are up- or down-regulated, respectively. Example:



*plot(), with(subset(), points())*

OPTIONAL: Try to represent your volcano plot using ggplot library. Little clue: you could need to create a column in your RES data.frame indicating if the gene is significant up, significant down or not significant

*ggplot(), geom\_point()*

How many probes are significant (adjusted p value <0.05)? How many genes?

3. Save significantly over-expressed genes (adjusted p value <0.05, logFC > 0) in a ”up” vector and significantly under-expressed genes (adjusted p value <0.05, logFC < 0) in a “down” vector. How many over- and under-expressed genes are there?

*which(), length()*

4. Use the *source()* function to read the script "Functions\_for\_supervised\_analysis.R" which contains the “enrichmentTest” function required in the following steps.

5. To better understand the significance of the deregulation of those genes, we will perform an enrichment analysis. The aim is to identify enrichment in cellular pathways among the up and down genes. Load the list of cellular pathways (gene sets) to be tested ("gene\_sets.Rdata"). You can use the EnrichmentTest function below to identify enriched pathways. You must specify the gene sets to be tested (argument *gene.sets*), the genes of interest (*up* or *down*, argument *mylist*), and all the genes tested (i.e. all the genes represented on the chip, argument *possibleIds*).

*enrichmentTest()*

6. Which pathway is most significantly enriched among the overexpressed genes in the UHCA subgroup? Which overexpressed genes belong to this pathway (column "corresponding elements in list")?

7. Load the list of cellular pathways (gene sets) to be tested ("gene\_sets.Rdata"). We are now going to try another type of enrichment analysis, namely GSEA. To do this we will use the “fgsea” library. This type of analysis requires that all the analyzed genes (significant and not significant) are ordered by fold change or p-value (in our case by fold change) from the highest to the lowest. Create a vector “rank” containing the sorted fold changes and do not forget to name each value of the vector with the corresponding gene name (*names()*).Use the fgsea function to perform the analysis and save the results in a new object.

For the fgsea function you should use the gene sets we used before (argument *pathway*), your vector “rank” with the sorted fold-changes (argument *stats*), the gene set should contain at least 30 genes (argument *minSize*), and maximum 500 genes (argument *maxSize*), with 1000 permutations (argument *nperm*).

*duplicated(), order(), fgsea()*

8. Which are the top pathways with an enrichment score (ES) higher than 0.7 (or 0.6) and p-value <0.05? Try to represent a GSEA plot of one of the enriched pathways using the function *plotEnrichment()* and save it as pdf.

*pdf(), plotEnrichment(), dev.off()*

9. As we did not find any recurrent alterations in the coding sequences (exome sequencing), we sequenced the complete genome of 3 tumours of the UHCA group. Complete genome sequencing makes it possible to identify structural rearrangements (deletions, duplications, chromosome translocations or inversions) that cannot be detected by exome sequencing. The table below shows the rearrangements identified in the 3 UHCAs. Which gene is altered in all 3 tumors? Is this consistent with your pathway analysis?

