**Hans-on session 1: Computational analysis of experimentally delineated mutational signatures; subtracting the signatures of background mutagenesis and of experimental artefacts**

**Part of: Tutorial 1: Mutational signature analysis: pipelines, machine learning, and benchmarking on synthetic data**

Prof. Steven G. Rozen, Dr. Arnoud Boot and Dr. Ferran Muiños

**Aim:** you will perform mutational signature analysis of experimental data generated in our lab. We exposed human cell lines MCF10A (breast ductal epithelium) and HepG2 (hepatoblastoma) to 2 commonly used chemotherapeutics; cisplatin, carboplatin and oxaliplatin. You will look for differences between the different cell lines and compounds, and assess whether the experimental results reflect the mutational signatures extracted from publicly available data.

**Requirements:**

This entire practical will be done in R version 4.0 or later.

You will need following packages from CRAN:

* + ICAMS v2.1.2
  + philentropy
  + gplots
  + factoextra
  + nloptr

All these packages are available on CRAN, and can be installed with:

install.packages("package.name")

You should have already installed ICAMS. You will need the Bioconductor package BSgenome.Hsapiens.1000genomes.hs37d5 (which you also should have already installed, but if not):

install.packages("BiocManager")

BiocManager::install("BSgenome.Hsapiens.1000genomes.hs37d5")

There are also 2 packages we will need from github

remotes::install\_github("steverozen/mSigBG", ref = "1.0-branch")

remotes::install\_github("steverozen/PCAWG7")

Load the packages:

library(ICAMS)

library(mSigBG)

library(PCAWG7)

library(philentropy)

library(gplots)

library(factoextra)

library(nloptr)

**Download the data required for these analyses**

<https://github.com/steverozen/ISMB.mutsig.tutorial/raw/master/ISMB.mutational.sigatures.practicum.zip>

unzip the folder, inside are

* Vcfs of HepG2 (N=4) and MCF10A (N=6) clones treated with cisplatin
* Spectra for HepG2 and MCF10A clones treated with carboplatin and oxaliplatin
* Spectra for untreated HepG2 and MCF10A clones, to establish the background mutagenesis pattern for each of the cell lines

Set this folder as your working directory e.g.

setwd("C:/Users/gmsarno/Desktop/ISMB.mutational.sigatures.practicum")

**Now let’s get started by loading the data;**

We start by generating mutation spectra for the cisplatin treated cell line clones.

First we list the vcf files to process:

path <- "vcfs/HepG2\_Cis/"

files <- list.files(path)

Then we read those vcfs using the ReadAndSplitStrelkaSBSVCFs function like this:

vcfs <- ReadAndSplitStrelkaSBSVCFs(paste0(path,files),

names.of.VCFs = gsub(".vcf.gz","",files))

The vcfs object now contains multiple types of vcfs, and the ReadAndSplitStrelkaSBSVCFs function separates the SBSs from DBSs and other substitutions. Within the SBS.vcfs list, there are separate vcfs for each of the individual samples.

You can look inside the list objects by typing list.name$ and pressing tab.

The next step is to generate the mutation spectrum for each of the samples. This is done using the VCFsToSBSCatalogs function:

HepG2\_Cis <- VCFsToSBSCatalogs(list.of.SBS.vcfs = vcfs$SBS.vcfs,

ref.genome = "GRCh37",

trans.ranges = trans.ranges.GRCh37,

region = "genome")

Similar to the vcfs object, the HepG2\_Cis object is also a list, this one containing 3 types of spectra. We will only be working with the basic mutational signatures in this tutorial; the 96 channel spectra, so to simplify: HepG2\_Cis <- HepG2\_Cis$catSBS96

You can visualize the spectra either in numerical form (View(HepG2\_Cis)), or you can plot the spectra to pdf using:

PlotCatalog(catalog = HepG2\_Cis[,1,drop=F])

Alternatively, you can also plot to pdf:

PlotCatalogToPdf(catalog = HepG2\_Cis,

file = "HepG2\_Cis\_SBS96.pdf")

You can now do the same for the MCF10A cell line clones treated with cisplatin.

For the remaining data (HepG2 treated with carboplatin and oxaliplatin and MCF10A treated with carboplatin), we have already provided the 96-channel mutational spectra, which can be loaded with the ReadCatalog function:

HepG2\_Car <- ReadCatalog("spectra/HepG2\_Car.csv")

HepG2\_Car<-as.catalog(object = HepG2\_Car,

ref.genome = "GRCh37",

region = "genome",

catalog.type = "counts")

Also read the HepG2 oxaliplatin and MCF10A carboplatin data

Plot all the spectra

**Q1: Visually check all the spectra; are there differences between the mutational spectra of the different compounds? Are there cell line specific differences? Is the mutation load comparable between the clones, compounds and cell lines?**

**Now we examine the differences between the spectra by clustering**

Combine all the 96-channel mutation spectra

all.spectra <- cbind(HepG2\_Cis,

HepG2\_Car,

HepG2\_Oxa,

MCF10A\_Cis,

MCF10A\_Car)

Heatmap clustering of all samples

cosine.sim <- distance(t(all.spectra),

method = "cosine" ,

use.row.names = T)

colnames(cosine.sim) <- rownames(cosine.sim)

heatmap.2(x = cosine.sim,

dendrogram = "column",

margins = c(9, 9),

cex.axis = 0.5,

symm = TRUE,

trace = "none")

PCA of all samples

PCA should be done in proportion space, as otherwise the mutation load would dominate the effect, not the pattern of the signature

We can transform the spectra to signatures like this:

all.spectra.as.sigs <- TransformCatalog(all.spectra,

target.catalog.type = "counts.signature")

pc <- prcomp(t(all.spectra.as.sigs),

center = TRUE, scale = FALSE, retx = TRUE)

fviz\_pca\_ind(pc)

**Q2: Does the clustering reflect your observations at Task 1?**

**Now we are ready to separate out the background**

As discussed during the lecture, due to the prolonged exposure, we expect there is also accumulation of background mutagenesis in our data; mutations that are not associated with the platinum-compounds.

We start by loading the spectra of the untreated clones, which we will use as our background signature, and convert it to a background object as required by mSigBG.

HepG2\_BG <- ReadCatalog("spectra/Background\_HepG2.csv")

HepG2\_BG <- as.catalog(object = HepG2\_BG,

ref.genome = "GRCh37",

region = "genome",

catalog.type = "counts")

HepG2\_BG <- mSigBG::MakeBackgroundInfo(HepG2\_BG,"HepG2\_BG")

Also load the MCF10A background spectra

Now we go ahead and perform the background separation

HepG2\_Cis\_noBG<-SeparateSignatureAndSpectra(

spectra = HepG2\_Cis,

bg.sig.info = HepG2\_BG,

start.b.fraction = 0.5,

sig.name = "HepG2\_Cis")

Also do this for the other datasets (this should take ~1.5 minutes per dataset)

Now plot the extracted signatures:

par(mfrow = c(2, 1),mar = c(2, 5, 6, 1),cex = 0.8,cex.main = 1.4)

PlotSpectraAsSigsWithUncertainty(

HepG2\_Cis\_noBG$inferred.target.spectra)

We can also plot the original spectra, visualizing the proportion of background mutagenesis in each peak of the 96-channel spectrum

par(mfrow = c(2, 1),mar = c(2, 5, 6, 1),cex = 0.8,cex.main = 1.4)

Plot1StackedSpectrum(HepG2\_Cis\_noBG$inferred.bg.spectra[,1,drop=F],

HepG2\_Cis\_noBG$inferred.target.spectra[,1,drop=F])

**Q3: Are the background mutational patterns of HepG2 and MCF10A similar? Do differences in background mutagenesis in the different cell lines affect the signature very much, and why? Do all compounds and cell lines have comparable levels of background mutagenesis and/or signal to noise ratio?**

**Q4: Using the clustering functions used previously, assess whether background separation reduces the difference between the mutational signatures observed in different compounds and different cell lines.**

**Compare experimental data to mutational signatures from human tumors**

The Pan Cancer of Whole Genomes consortium working group 7 (PCAWG7) discovered 2 mutational signatures that have been attributed to platinum chemotherapy SBS31 and SBS35. However, by eye, we could see elements of both SBS31 and SBS35 back in our experimental cisplatin signature.

One of the hypotheses was that one was due to cisplatin, and the other due to oxaliplatin.

With this experimental data, we can test this.

To test whether SBS31 and SBS35 are a split of the cisplatin signature, we can use the following function for numerical optimization

one.opt <- function(sig, method = "cosine", invert = -1) {

SBS31<-PCAWG7::signature$genome$SBS96[,"SBS31"]

SBS35<-PCAWG7::signature$genome$SBS96[,"SBS35"]

my.obj.fn <- function(coef) {

recon <- coef[1]\*SBS31 + coef[2]\*SBS35

return(

suppressMessages(

invert \* philentropy::distance(

rbind(recon, as.vector(sig)), method = method)))

}

g\_ineq <- function(coef) { abs(1 - sum(coef)) }

retval <- nloptr::nloptr(

x0 = c(0.5, 0.5),

eval\_f = my.obj.fn,

eval\_g\_ineq = g\_ineq,

lb = c(0, 0),

ub = c(1, 1),

opt = list(algorithm = "NLOPT\_LN\_COBYLA",maxeval = 2000,

xtol\_rel = 1e-6, xtol\_abs = 1e-7))

retval <- list(similarity = invert \* retval$objective,

coef = retval$solution / sum(retval$solution))

return(retval)

}

We can apply this function to the experimental signatures like this:

one.opt(HepG2\_Cis\_noBG$inferred.target.sig.as.catalog)

The output contains the similarity (cosine similarity), as well as the proportion of SBS31 and SBS35 which gives that similarity.

**Q5: Are SBS31 and SBS35 different platinum compounds, or are they a split of the experimental signature? Do the signatures of all experimental compounds get reconstructed equally well? If not, does the mutation load of the experimental data explain which compounds get reconstructed best?**