DNA Methylation Deconvolution Protocol

Michael Scherer, Pavlo Lutsik, Petr Nazarov, Tony Kamoa

January 23, 2019

Introduction

This protocols aims at guiding reasearcher how to correcting employ deconvolution of methylomes obtained from complex tissue. It will start with data retrieval from a public resource, but is equally applicable to in-house generated data. We will furthermore focus on the Illumina BeadChip series as a data source, although the protocol is also compatible with bisulfite sequencing protocols that provide single base pair resolution.

Obtaining data from a public resource (duration ~5h)

We focus on DNA methylation data from cancer patients that has been generated in The Cancer Genome Atlas (TCGA) project. Since lung cancer has been shown to be a premier candidate for DNA methylation based deconvolution, we selected the lung adenocarcinoma dataset from the TCGA website (dataset TCGA-LUAD, https://portal.gdc.cancer.gov/legacy-archive/search/f). The dataset was generated using the Illumina Infinum 450k BeadChip and comprises 461 samples. Due to restrucuring of the TCGA data archive, the clinical metadata of the samples is available at https://portal.gdc.cancer.gov/projects/TCGA-LUAD and lists 585 samples. We used the GDC data download tool (https://gdc.cancer.gov/access-data/gdc-data-transfer-tool) to download the IDAT files listed in the manifest file and its associated metadata. This metadata also includes the mapping between each of the samples and the IDAT files. To create a final mapping and to prepare the files for RnBeads analyis, the following code was employed.

```
#knitr::opts_knit$set(root.dir="/DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/")
#setwd("/DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/")
clinical.data <- read.table("annotation/clinical.tsv",sep="\t",header=T)
idat.files <- list.files("idat",full.names = T)
meta.files <- list.files(idat.files[1],full.names = T)
untar(meta.files[3],exdir = idat.files[1])
meta.files <- untar(meta.files[3],list=T)
```

```
meta.info <- read.table(file.path(idat.files[1],meta.files[5]),sep="\t",header=T)
meta.info <- meta.info[match(unique(meta.info$Comment..TCGA.Barcode.),meta.info$Comment..TC
match.meta.clin <- match(clinical.data$submitter_id,substr(meta.info$Comment..TCGA.Barcode.
anno.frame <- na.omit(data.frame(clinical.data,meta.info[match.meta.clin,]))
anno.frame$barcode <- unlist(lapply(lapply(as.character(anno.frame$Array.Data.File),functio
anno.frame$Sentrix_ID <- unlist(lapply(lapply(as.character(anno.frame$Array.Data.File),funct
anno.frame$Sentrix_Position <- unlist(lapply(lapply(as.character(anno.frame$Array.Data.File)
write.table(anno.frame,"annotation/sample_annotation.tsv",quote=F,row.names = F,sep="\t")
anno.frame <- read.table("annotation/sample_annotation.tsv",quote=F,row.names = F,sep="\t")

#' write idat files to parent directory
lapply(idat.files,function(x){
  is.idat <- list.files(x,pattern = ".idat",full.names = T)
  file.copy(is.idat,"idat/")
  unlink(x,recursive = T)
})
```

Data Import and Quality Control in RnBeads (~3h)

After downloading the data, it has to be processed into a format that can be used by downstream software. We used RnBeads to convert the files into a data object and did basic quality control steps on the dataset. Most notably, analysis options need to be specified for RnBeads, either through an XML file or in the command line. We will follow the latter strategy here, and deactivate the preprocessing, exploratory, covariate inference and differential methylation modules. In the next step, we specify the input to RnBeads: the created sample annotation sheet, the folder in which the IDAT files are stored and a folder to which the HTML report is to be saved. We additionally recommend to specify a temporary directory for the analysis. Then we start the RnBeads analysis.

```
library(RnBeads)
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     Filter, Find, Map, Position, Reduce, anyDuplicated, append,
##     as.data.frame, basename, cbind, colMeans, colSums, colnames,
##     dirname, do.call, duplicated, eval, evalq, get, grep, grepl,
##     intersect, is.unsorted, lapply, lengths, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, rank, rbind,
##     rowMeans, rowSums, rownames, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min

## Loading required package: S4Vectors

## Warning: package 'S4Vectors' was built under R version 3.5.1

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: GenomicRanges

## Warning: package 'GenomicRanges' was built under R version 3.5.1

## Loading required package: IRanges

## Warning: package 'IRanges' was built under R version 3.5.1

## Loading required package: GenomeInfoDb

## Loading required package: MASS

## Loading required package: cluster

## Loading required package: ff
```

```
## Warning: package 'ff' was built under R version 3.5.1

## Loading required package: bit

## Warning: package 'bit' was built under R version 3.5.1

## Attaching package bit

## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL-2)

## creators: bit bitwhich

## coercion: as.logical as.integer as.bit as.bitwhich which

## operator: ! & | xor != ==

## querying: print length any all min max range sum summary

## bit access: length<- [ [<- [[ [[<-

## for more help type ?bit

##
## Attaching package: 'bit'

## The following object is masked from 'package:base':
##
##     xor

## Attaching package ff

## - getOption("fftempdir")=="/tmp/RtmpnLc80g"

## - getOption("ffextension")=="ff"

## - getOption("ffdrop")==TRUE

## - getOption("fffinonexit")==TRUE

## - getOption("ffpagesize")==65536
```

```
## - getOption("ffcaching")=="mmnoflush"  -- consider "ffeachflush" if your system stalls or

## - getOption("ffbatchbytes")==16777216 -- consider a different value for tuning your syste

## - getOption("ffmaxbytes")==536870912 -- consider a different value for tuning your system

##
## Attaching package: 'ff'

## The following objects are masked from 'package:bit':
##
##     clone, clone.default, clone.list

## The following objects are masked from 'package:utils':
##
##     write.csv, write.csv2

## The following objects are masked from 'package:base':
##
##     is.factor, is.ordered

## Loading required package: fields

## Loading required package: spam

## Loading required package: dotCall64

## Loading required package: grid

## Spam version 2.1-4 (2018-04-12) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following object is masked from 'package:stats4':
##
##     mle
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: maps

##
## Attaching package: 'maps'

## The following object is masked from 'package:cluster':
##
##     votes.repub

## See www.image.ucar.edu/~nychka/Fields for
##  a vignette and other supplements.

## Loading required package: ggplot2

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess

## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:BiocGenerics':
##
##     combine
```

```
## Loading required package: limma

## Warning: package 'limma' was built under R version 3.5.1

##
## Attaching package: 'limma'

## The following object is masked from 'package:BiocGenerics':
##
##     plotMA

## Loading required package: matrixStats

## Loading required package: illuminaio

## Loading required package: methylumi

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians

## Loading required package: scales

## Loading required package: reshape2

## Loading required package: FDb.InfiniumMethylation.hg19

## Loading required package: GenomicFeatures

## Loading required package: AnnotationDbi
```

```
##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:MASS':
##
##     select

## Loading required package: TxDb.Hsapiens.UCSC.hg19.knownGene

## Loading required package: org.Hs.eg.db

##

## Loading required package: minfi

## Loading required package: SummarizedExperiment

## Loading required package: DelayedArray

## Loading required package: BiocParallel

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

## The following objects are masked from 'package:base':
##
##     aperm, apply

## Loading required package: Biostrings

## Loading required package: XVector

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:DelayedArray':
##
##     type
```

```
## The following objects are masked from 'package:ff':
##
##     mismatch, pattern

## The following object is masked from 'package:base':
##
##     strsplit

## Loading required package: bumphunter

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: locfit

## locfit 1.5-9.1    2013-03-22

## Setting options('download.file.method.GEOquery'='auto')

## Setting options('GEOquery.inmemory.gpl'=FALSE)

## Loading required package: plyr

##
## Attaching package: 'plyr'

## The following object is masked from 'package:XVector':
##
##     compact

## The following object is masked from 'package:matrixStats':
##
##     count

## The following object is masked from 'package:maps':
##
##     ozone

## The following object is masked from 'package:IRanges':
##
##     desc
```

```
## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```
#knitr::opts_knit$set(root.dir="/DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/")
rnb.options(
  assembly="hg19",
  identifiers.column="submitter_id",
  import=T,
  import.default.data.type="idat.dir",
  import.table.separator="\t",
  import.sex.prediction=T,
  qc=F,
  preprocessing=F,
  exploratory=F,
  inference=F,
  differential=F,
  export.to.bed=F,
  export.to.trackhub=NULL,
  export.to.csv=F
)
sample.anno <- "annotation/sample_annotation.tsv"
idat.folder <- "idat/"
dir.report <- paste0("report",Sys.Date(),"/")
temp.dir <- "/DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp"
options(fftempdir=temp.dir)
rnb.set <- rnb.run.analysis(dir.reports = dir.report, sample.sheet = sample.anno, data.dir =
```

```
## 2019-01-30 09:20:52     1.1  STATUS STARTED RnBeads Pipeline
## 2019-01-30 09:20:52     1.1    INFO     Initialized report index and saved to index.html
## 2019-01-30 09:20:52     1.1  STATUS     STARTED Loading Data
## 2019-01-30 09:20:52     1.1    INFO         Number of cores: 1
## 2019-01-30 09:20:52     1.1    INFO         Loading data of type "idat.dir"
## 2019-01-30 09:20:52     1.1  STATUS         STARTED Loading Data from IDAT Files
## 2019-01-30 09:20:53     1.1    INFO             Detected platform: HumanMethylation450
```

```
## Warning: package 'RnBeads.hg19' was built under R version 3.5.1
```

```
## 2019-01-30 09:38:12     1.4  STATUS         COMPLETED Loading Data from IDAT Files
## 2019-01-30 10:27:29     2.1  STATUS         Loaded data from idat/
## 2019-01-30 10:28:07     7.7  STATUS         Predicted sex for the loaded samples
## 2019-01-30 10:28:27     7.1  STATUS         Added data loading section to the report
## 2019-01-30 10:28:27     7.1  STATUS         Loaded 461 samples and 485577 sites
## 2019-01-30 10:28:27     7.1    INFO         Output object is of type RnBeadRawSet
## 2019-01-30 10:28:27     7.1  STATUS     COMPLETED Loading Data
```

```
## 2019-01-30 10:42:13     7.1    INFO     Initialized report index and saved to index.html
## 2019-01-30 10:42:13     7.1  STATUS     STARTED Saving RData
## 2019-01-30 10:42:13     7.1  STATUS     COMPLETED Saving RData
## 2019-01-30 10:42:13     7.1  STATUS COMPLETED RnBeads Pipeline
```

Preprocessing and Filtering

For further analysis, we use the DecompPipeline package (https://github.com/lutsik/DecompPipeline), which provides a comprehensive workflow including crucial data preparation steps for methylome deconvolution experiments. The options are provided through the individuals function's parameters. We follow a stringent filtering strategy. First, all samples having fewer than 3 beads covered are filtered, as well as those probes that are in the 0.05 and 0.95 overall intensity quantiles. We then remove all probes containing missing values, outside of CpG context, that overlap with annotated SNPs, on the sex chromosomes and probes that have been shown to be cross-reactive on the chip. Then, BMIQ normalization [@bmiq] is employed to account for the chip's design bias.

```
library(DecompPipeline)
```

```
## Loading required package: MeDeCom
```

```
## Loading required package: Rcpp
```

```
## Loading required package: pracma
```

```
##
## Attaching package: 'pracma'
```

```
## The following object is masked from 'package:ff':
##
##     quad
```

```
## The following object is masked from 'package:bit':
##
##     is.sorted
```

```
## Loading required package: gtools
```

```
##
## Attaching package: 'gtools'
```

```
## The following object is masked from 'package:pracma':
##
##     logit

## Loading required package: RUnit

## Warning: replacing previous import 'gtools::logit' by 'pracma::logit' when
## loading 'MeDeCom'

## Loading required package: R.utils

## Loading required package: R.oo

## Loading required package: R.methodsS3

## R.methodsS3 v1.7.1 (2016-02-15) successfully loaded. See ?R.methodsS3 for help.

## R.oo v1.22.0 (2018-04-21) successfully loaded. See ?R.oo for help.

##
## Attaching package: 'R.oo'

## The following object is masked from 'package:methylumi':
##
##     getHistory

## The following object is masked from 'package:SummarizedExperiment':
##
##     trim

## The following objects are masked from 'package:ff':
##
##     clone, finalize

## The following object is masked from 'package:bit':
##
##     clone

## The following object is masked from 'package:GenomicRanges':
##
##     trim
```

```
## The following object is masked from 'package:IRanges':
##
##     trim

## The following objects are masked from 'package:methods':
##
##     getClasses, getMethods

## The following objects are masked from 'package:base':
##
##     attach, detach, gc, load, save

## R.utils v2.7.0 successfully loaded. See ?R.utils for help.

##
## Attaching package: 'R.utils'

## The following object is masked from 'package:gtools':
##
##     capture

## The following object is masked from 'package:RnBeads':
##
##     off

## The following object is masked from 'package:spam':
##
##     cleanup

## The following object is masked from 'package:utils':
##
##     timestamp

## The following objects are masked from 'package:base':
##
##     cat, commandArgs, getOption, inherits, isOpen, parse, warnings

data.prep <- prepare_data(RNB_SET = rnb.set,
                          analysis.name = "TCGA_Deconvolution",
                          NORMALIZATION = "bmiq",
                          FILTER_BEADS = T,
                          MIN_N_BEADS = 3,
                          FILTER_INTENSITY = T,
```

```
                            MIN_INT_QUANT = 0.05,
                            MAX_INT_QUANT = 0.95,
                            FILTER_NA = T,
                            FILTER_CONTEXT = T,
                            FILTER_SNP = T,
                            FILTER_SOMATIC = T,
                            FILTER_CROSS_REACTIVE = T)

## 2019-01-30 10:52:13    18.8    INFO 163614 sites removed in bead count filtering.
## 2019-01-30 10:53:23    21.4    INFO 249310 sites removed in intensity filtering.
## 2019-01-30 10:53:44    13.8    INFO 0 sites removed in NA filtering
## 2019-01-30 10:53:45    13.8    INFO 8866 sites removed in SNP filtering
## 2019-01-30 10:53:45    13.8    INFO 611 sites removed in somatic sites filtering
## 2019-01-30 10:53:45    13.8    INFO 236 sites removed in CG context filtering
## 2019-01-30 10:53:45    13.8    INFO Removing 422637 sites, retaining  62940

## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff674237573b8b.ff

## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff674236a92ae3.ff

## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff67427ac840ca.ff

## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff67423c650c75.ff

## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff67422bd2cc9d.ff

## 2019-01-30 11:25:33    13.3    INFO 2646 sites removed in cross-reactive filtering
```

Selecting informative features (CpGs)

The next, crucial, step is selecting a subset of sites that are informative about the cell type composition of your sample. This can be done in various ways, and DecompPipeline provides a list of them through the `prepare_CG_subsets` function. However, we focus on a single option, which we found to work well in many scenarios: feature selection through Indepedent Component Analysis (ICA).

Performing Deconvolution

In this step, the actual deconvolution experiment is performed. There are different approaches, which are conceptually similar, yet different in their performance, running time and robustness. Among others, EDec, RefFreeCellMix from the

RefFreeEWAS package and MeDeCom can be used to execute non-negative matrix factorization on your data. This will lead to two matrices, the proportions matrix of potential cell types (here referred to as LMCs) and the matrix of those pure profiles. We here focus on MeDeCom as the Deconvolution tool, although DecompPipeline equally support RefFreeCellMix and EDec.

Downstream analysis

After performing deconvolution, results need to be visualized and interpreted. Most notably, the contribution matrix can be linked to phenotypic information about the samples to indicate different cellular compositions of the groups and the LMC matrix can be used to determine what the component represent. For visualization and downstream analysis, we use FactorViz. Enrichment analysis can be employed on sites that are specifically methylated/unmethylated in one of the LMCs.