Introduction

This protocol aims at guiding reasearcher how to 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 that provides single base pair resolution. Deconvolution here refers to creating two matrices (proportion matrix A and methylation pattern matrix T) from a single matrix of input DNA methylation data (dimension CpGs x samples). Non-negative matrix factorization can be employed for this task, and we will discuss some of the advantages and caveats of the methods.

Protocol

Data Retrival

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. The clinical metadata of the samples is available at https://portal.gdc.cancer.gov/projects/TCGA-LUAD and lists 585 samples. The discrepancy between the number comes from recent progress within TCGA. We used the Genomic Data Commons (GDC) data download tool (https://gdc.cancer.gov/access-data/gdc-data-transfer-tool) to download the intensity data (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 downstream analysis, the following code was employed.

```
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),functior
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)
anno.frame$healthy_cancer <- ifelse(grepl("11A",anno.frame$Comment..TCGA.Barcode.),"healthy"
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 Processing

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 performed 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.

```
suppressPackageStartupMessages(library(RnBeads))
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=T,
  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 =

## 2019-06-25 17:44:15     1.1  STATUS STARTED RnBeads Pipeline
## 2019-06-25 17:44:15     1.1    INFO     Initialized report index and saved to index.html
## 2019-06-25 17:44:15     1.1  STATUS     STARTED Loading Data
## 2019-06-25 17:44:15     1.1    INFO         Number of cores: 1
## 2019-06-25 17:44:16     1.1    INFO         Loading data of type "idat.dir"
## 2019-06-25 17:44:16     1.1  STATUS         STARTED Loading Data from IDAT Files
## 2019-06-25 17:44:17     1.1    INFO             Detected platform: HumanMethylation450
## 2019-06-25 18:04:40     1.5  STATUS         COMPLETED Loading Data from IDAT Files
## 2019-06-25 18:48:47     2.0  STATUS         Loaded data from idat/
## 2019-06-25 18:49:41     7.7  STATUS         Predicted sex for the loaded samples
## 2019-06-25 18:50:09     7.1  STATUS         Added data loading section to the report
## 2019-06-25 18:50:09     7.1  STATUS         Loaded 461 samples and 485577 sites
## 2019-06-25 18:50:09     7.1    INFO         Output object is of type RnBeadRawSet
## 2019-06-25 18:50:09     7.1  STATUS     COMPLETED Loading Data
## 2019-06-25 19:03:57     7.1    INFO     Initialized report index and saved to index.html
## 2019-06-25 19:03:58     7.1  STATUS     STARTED Quality Control
## 2019-06-25 19:03:58     7.1    INFO         Number of cores: 1
## 2019-06-25 19:03:58     7.1  STATUS         STARTED Quality Control Section

## 2019-06-25 19:04:30     2.0  STATUS             Added quality control box plots

## 2019-06-25 19:09:19     2.0  STATUS             Added quality control bar plots

## 2019-06-25 19:09:44     2.0  STATUS             Added negative control boxplots
## 2019-06-25 19:09:44     2.0  STATUS         COMPLETED Quality Control Section
## 2019-06-25 19:09:44     2.0  STATUS         STARTED Visualizing SNP Probe Data
## 2019-06-25 19:09:44     2.0  STATUS             STARTED Mixups Visualization Section

## 2019-06-25 19:10:19     5.4  STATUS                 Added SNP Heatmap

## Found more than one class "dist" in cache; using the first, from namespace 'BiocGenerics'

## Also defined by 'spam'

## Found more than one class "dist" in cache; using the first, from namespace 'BiocGenerics'
```

```
## Also defined by 'spam'

## Found more than one class "dist" in cache; using the first, from namespace 'BiocGenerics'

## Also defined by 'spam'

## 2019-06-25 19:10:20     5.4  STATUS                 Calculated Manhattan distances betwee

## 2019-06-25 19:10:23     5.4  STATUS                 Added SNP-based Distances
## 2019-06-25 19:10:23     5.4  STATUS             COMPLETED Mixups Visualization Section
## 2019-06-25 19:10:23     5.4  STATUS         COMPLETED Visualizing SNP Probe Data

## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff500b33eb3301.ff

## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff500b77a0b0ce.ff

## 2019-06-25 19:11:24     7.7  STATUS     COMPLETED Quality Control
## 2019-06-25 19:11:24     7.7    INFO     Initialized report index and saved to index.html
## 2019-06-25 19:11:25     7.7  STATUS     STARTED Saving RData
## 2019-06-25 19:11:25     7.7  STATUS     COMPLETED Saving RData
## 2019-06-25 19:11:25     7.7  STATUS COMPLETED RnBeads Pipeline
```

RnBeads creates an interactive HTML report, specifying the steps performed and the associated results. Data was of good quality such that is can be used for further analysis. (Include two screenshots from the RnBeads report)

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 individual function 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, respectively. 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. Accounting for potential confounding factor is crucial in epigenomic studies. Especially, the influence of donor sex and age on the DNA methylation pattern is well-studied and strong. Furthermore, genetic differences between groups of individuals due to different origins may influence the DNA methylation paterrn. We used Independent Component Analysis (ICA) to account for DNA

methylation differences that are due to these confounding factors. ICA detects components in the data accounting for most of the variance similar to PCA, but does not require orthogonality of the components but statistical independence. We used an external library (http://sablab.net/scripts/LibICA.r) for performing ICA to adjust for sex, age, race and ethnicity.

```
suppressPackageStartupMessages(library(DecompPipeline))
data.prep <- prepare_data(RNB_SET = rnb.set,
                          analysis.name = "TCGA_LUAD",
                          NORMALIZATION = "bmiq",
                          FILTER_BEADS = T,
                          MIN_N_BEADS = 3,
                          FILTER_INTENSITY = T,
                          MIN_INT_QUANT = 0.001,
                          MAX_INT_QUANT = 0.999,
                          FILTER_NA = T,
                          FILTER_CONTEXT = T,
                          FILTER_SNP = T,
                          FILTER_SOMATIC = T,
                          FILTER_CROSS_REACTIVE = T,
                          execute.lump=T,
                          remove.ICA=T,
                          conf.fact.ICA=c("age_at_diagnosis","race","gender","ethnicity"),
                          ica.setting=c("alpha.fact"=1e-5))
```

```
## 2019-06-25 19:22:19    17.9    INFO 163614 sites removed in bead count filtering.
## 2019-06-25 19:23:28    21.7    INFO 27623 sites removed in intensity filtering.
## 2019-06-25 19:23:54    22.2    INFO 0 sites removed in NA filtering
## 2019-06-25 19:23:54    22.2    INFO 44274 sites removed in SNP filtering
## 2019-06-25 19:23:54    22.2    INFO 6269 sites removed in somatic sites filtering
## 2019-06-25 19:23:55    22.2    INFO 1147 sites removed in CG context filtering
## 2019-06-25 19:23:55    22.2    INFO Removing 242927 sites, retaining  242650
```

```
## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff500b75dc6533.ff
```

```
## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff500b4f959343.ff
```

```
## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff500be127689.ff
```

```
## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff500b3e452644.ff
```

```
## opening ff /DEEP_fhgfs/projects/mscherer/data/450K/TCGA/LUAD/tmp/ff500b3395e5e8.ff
```

```
## 2019-06-25 20:28:48    11.6    INFO 12427 sites removed in cross-reactive filtering
## 2019-06-25 20:28:48    11.6  STATUS STARTED Removing confounding factors using ICA
## 2019-06-25 20:28:48    11.6    INFO     No imputation method selected, 'knn' method used.
## 2019-06-25 20:28:48    11.6  STATUS     STARTED Imputation procedure knn
## 2019-06-25 20:57:25    22.4  STATUS     COMPLETED Imputation procedure knn
## 2019-06-25 20:57:26    22.4  STATUS     STARTED Determining number of components
## 2019-06-25 20:57:59    21.7    INFO         Only sites with SD > 0 were kept: 230223 of 2

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## 2019-06-25 20:57:59    21.7    INFO         Assuming numeric data for pheno column ` age_

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## 2019-06-25 20:57:59    21.7    INFO         Assuming numeric data for pheno column ` days

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## 2019-06-25 20:57:59    21.7    INFO         Assuming numeric data for pheno column ` days

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## 2019-06-25 20:57:59    21.7    INFO         Assuming numeric data for pheno column ` days

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion
```

```
## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## Warning in getComponentNumber(rnb.set, conf.factor, nmin = nmin, nmax =
## nmax, : NAs introduced by coercion

## 2019-06-25 20:57:59    21.7    INFO         getMinCompNumber: working with 10 components

## Loading required package: fastICA

## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 10 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 20:58:02
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 20:59:15
## Time difference of 1.215448 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 20:59:17    26.1    INFO         0.102315904066909; 0.0640267167857304; 3.914
## 2019-06-25 20:59:17    26.1    INFO         getMinCompNumber: working with 11 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 11 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 20:59:17
## Execute one-core analysis, showing progress every 1 run(s)
```

```
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:00:27
## Time difference of 1.165935 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:00:29    26.2    INFO         0.00900900821212266; 0.0431124129468957; 4.61
## 2019-06-25 21:00:29    26.2    INFO         getMinCompNumber: working with 12 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 12 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:00:29
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:01:49
## Time difference of 1.330911 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:01:51    26.2    INFO         0.00596406508227194; 0.062667236618924; 8.079
## 2019-06-25 21:01:51    26.2    INFO         getMinCompNumber: working with 13 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 13 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:01:51
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:03:13
## Time difference of 1.360138 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:03:16    27.0    INFO         2.60157559255724e-05; 0.0178307618458239; 8.2
## 2019-06-25 21:03:16    27.0    INFO         getMinCompNumber: working with 14 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 14 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:03:16
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:04:41
## Time difference of 1.418273 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:04:43    26.2    INFO         0.0110307903871836; 0.0208455388292142; 4.797
## 2019-06-25 21:04:43    26.2    INFO         getMinCompNumber: working with 15 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 15 components, 1 runs, 230223 features, 461 samples.
```

```
## *** Start time: 2019-06-25 21:04:43
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:06:18
## Time difference of 1.57415 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:06:20    26.2    INFO         0.0129006790513419; 0.0174850449594158; 9.294
## 2019-06-25 21:06:20    26.2    INFO         getMinCompNumber: working with 16 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 16 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:06:20
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:08:06
## Time difference of 1.751105 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:08:08    25.5    INFO         0.000112649794832954; 0.0047641304482537; 0.0
## 2019-06-25 21:08:08    25.5    INFO         getMinCompNumber: working with 17 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 17 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:08:08
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:10:06
## Time difference of 1.967924 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:10:09    26.2    INFO         0.00533438349913006; 0.000922447247656091; 4.
## 2019-06-25 21:10:09    26.2    INFO         getMinCompNumber: working with 18 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 18 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:10:09
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:12:12
## Time difference of 2.048571 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:12:16    26.2    INFO         0.00227287128313991; 0.000404422856278843; 2.
## 2019-06-25 21:12:16    26.2    INFO         getMinCompNumber: working with 19 components
## *** Starting  calculation on 1 core(s)...
```

```
## *** System: unix
## *** 19 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:12:16
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:14:24
## Time difference of 2.133444 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:14:27    27.0    INFO         0.00033098648617749; 5.33313959678689e-06; 9.
## 2019-06-25 21:14:27    27.0    INFO         getMinCompNumber: working with 20 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 20 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:14:27
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:16:47
## Time difference of 2.330915 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:16:50    29.3    INFO         0.0276182355878157; 2.05690715209953e-05; 2.2
## 2019-06-25 21:16:50    29.3    INFO         getMinCompNumber: working with 21 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 21 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:16:50
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:19:42
## Time difference of 2.854681 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:19:45    26.8    INFO         0.00950085058058992; 2.97202018270671e-05; 1.
## 2019-06-25 21:19:45    26.8    INFO         getMinCompNumber: working with 22 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 22 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:19:45
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:22:32
## Time difference of 2.775942 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:22:35    26.8    INFO         0.000499604904914473; 7.43158627623381e-05; 9
```

```
## 2019-06-25 21:22:35    26.8    INFO         getMinCompNumber: working with 23 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 23 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:22:35
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:25:52
## Time difference of 3.272885 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:25:56    29.1    INFO         0.000642506120713836; 9.35379398466788e-05; 3
## 2019-06-25 21:25:56    29.1    INFO         getMinCompNumber: working with 24 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 24 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:25:56
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:29:43
## Time difference of 3.797569 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:29:47    27.5    INFO         0.000238400412644002; 0.000929232807364356; 5
## 2019-06-25 21:29:47    27.5    INFO         getMinCompNumber: working with 25 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 25 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:29:47
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:32:32
## Time difference of 2.736203 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:32:36    29.1    INFO         0.000129157488119915; 0.00127641671101029; 2.
## 2019-06-25 21:32:36    29.1    INFO         getMinCompNumber: working with 26 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 26 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:32:36
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:36:17
## Time difference of 3.682293 mins
```

```
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:36:21    28.3    INFO         0.00103250914566519; 0.000287092199625268; 9.
## 2019-06-25 21:36:21    28.3    INFO         getMinCompNumber: working with 27 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 27 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:36:21
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:40:25
## Time difference of 4.068034 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:40:29    29.1    INFO         0.0061677677459012; 0.000770210460797161; 1.1
## 2019-06-25 21:40:29    29.1    INFO         getMinCompNumber: working with 28 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 28 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:40:29
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:45:18
## Time difference of 4.815393 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:45:22    29.9    INFO         0.000719051040292291; 0.00060566100595299; 1.
## 2019-06-25 21:45:22    29.9    INFO         getMinCompNumber: working with 29 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 29 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:45:22
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:49:37
## Time difference of 4.235621 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:49:41    26.1    INFO         0.000100764215522075; 0.00174598516800233; 5.
## 2019-06-25 21:49:41    26.1    INFO         getMinCompNumber: working with 30 components
## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 30 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:49:41
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
```

```
## *** End time: 2019-06-25 21:54:04
## Time difference of 4.385208 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:54:09    26.2    INFO         0.00160717186944378; 0.00267991924115455; 3.5
## 2019-06-25 21:54:09    26.2  STATUS     COMPLETED Determining number of components
## 2019-06-25 21:54:18    27.7  STATUS     STARTED Removing factor effect

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## 2019-06-25 21:54:27    26.9    INFO         Assuming numeric data for pheno column ` age_

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## 2019-06-25 21:54:27    26.9    INFO         Assuming numeric data for pheno column ` days

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## 2019-06-25 21:54:27    26.9    INFO         Assuming numeric data for pheno column ` days

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## 2019-06-25 21:54:27    26.9    INFO         Assuming numeric data for pheno column ` days

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
```

```
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in removeFactor(rnb.set, fact = conf.factor, ncomp = ncomp, ntry =
## ntry, : NAs introduced by coercion

## Warning in if (!fact %in% names(Var)) {: the condition has length > 1 and
## only the first element will be used

## *** Starting  calculation on 1 core(s)...
## *** System: unix
## *** 19 components, 1 runs, 230223 features, 461 samples.
## *** Start time: 2019-06-25 21:54:27
## Execute one-core analysis, showing progress every 1 run(s)
## try # 1 of 1
## *** Done!
## *** End time: 2019-06-25 21:56:34
## Time difference of 2.110113 mins
## Calculate ||X-SxM|| and r2 between component weights
## 2019-06-25 21:56:37    26.9    INFO         Component 6 is linked to gender factor, p-val
## 2019-06-25 21:56:37    26.9    INFO         Component 16 is linked to ethnicity factor, p
## 2019-06-25 21:56:55    28.5  STATUS     COMPLETED Removing factor effect

## No id variables; using all as measure variables

## Saving 7 x 7 in image

## 2019-06-25 22:27:34    36.0  STATUS COMPLETED Removing confounding factors using ICA
```

```
names(data.prep)
```

```
## [1] "quality.filter"   "annot.filter"     "total.filter"
## [4] "rnb.set.filtered" "info"
```

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 is typically employed in epigenomic studies: selecting the most variable sites across the samples. Since many sites are constant for all samples, focusing on the ones that show the highest variablity across the samples is sensible. We assume that we do not lose information by not considering those sites that do not vary at all. Here, we focus on the 5,000 most variable sites.

```
cg_subset <- prepare_CG_subsets(rnb.set=data.prep$rnb.set.filtered,
                                MARKER_SELECTION = "var",
                                N_MARKERS = 5000)
names(cg_subset)
```

```
## [1] "var"
```

Methylome Deconvolution

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 also supports RefFreeCellMix and EDec.

```
md.res <- start_medecom_analysis(
  rnb.set=data.prep$rnb.set.filtered,
  cg_groups = cg_subset,
  Ks=2:15,
  LAMBDA_GRID = c(0,10^-(2:5)),
  factorviz.outputs = T,
```

```
  analysis.name = "TCGA_LUAD",
  cores = 15
)
```

```
## 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
```

```
## The following object is masked from 'package:R.utils':
##
##     capture
```

```
## Loading required package: RUnit
```

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

```
## [1] "Did not write the variable dump: should only be executed from an environment with al
## [2019-06-25 22:28:18, Main:] checking inputs
## [2019-06-25 22:28:18, Main:] preparing data
## [2019-06-25 22:28:18, Main:] preparing jobs
## [2019-06-25 22:28:18, Main:] 3570 factorization runs in total
## [2019-06-27 13:54:19, Main:] finished all jobs. Creating the object
```

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 components represent. For visualization and downstream analysis, we use FactorViz. LOLA or GO enrichment analysis can be employed on sites that are specifically methylated/unmethylated in one of the LMCs.

```
suppressPackageStartupMessages(library(FactorViz))
startFactorViz(file.path(getwd(),"TCGA_LUAD","FactorViz_outputs"))
```

References