BayesPrism/TED Package

TED packages contains three major functions, 1) the “run.Ted” function which implements a fully Bayesian inference of tumor microenvironment composition and gene expression, 2) the “learn.embedding.Kcls” and 3) “learn.embedding.withPhiTum” functions which uses Expectation-maximization (EM) to approximate the tumor expression using a linear combination of tumor pathways while conditional on the inferred expression and fraction of non-tumor cells estimated by the deconvolution module. The “learn.embedding.Kcls” function initializes the embedding by running hierarchical clustering over the tumor expression inferred by run.Ted, while “learn.embedding.withPhiTum” initializes using the embedding provided by the user.

In this example we will be deconvolving 169 TCGA-GBM bulk RNA-seq samples using the scRNA-seq dataset from 8 high grade glioma patients as the reference. We will also demonstrate the embedding learning of tumor pathways from TCGA-GBM samples. The "tcga.gbm.example.rdata" contains the pre-prepared normalized scRNA-seq gene expression profile (GEP) matrix and bulk sample matrix over protein-coding genes. In practice, users may also use unnormalized GEP or raw count matrix of individual cells. Genes need not to be aligned between the reference matrix and the bulk matrix. run.Ted will automatically collapse, align the genes on the common subset between reference and bulk, and normalized the scRNA-seq reference.

1) Deconvolve bulk RNA-seq and infer tumor expression

```
#load TED package
    > library(TED)

# load pre-prepared reference scRNA-seq raw data and the bulk sample matrix
# please specify the gbm.rdata from the clone git repository

    > load(".../tutorial.dat/gbm.rdata")

# Remove ribosomal and mitochondrial genes from the reference matrix, may also remove genes on the sex chromosomes if samples are collected on different genders.
# We curated from several commonly used annotations. use ?cleanup.genes to find out more details.
# Users are also recommended to manually do this step using their matched annotation file / curated genes.
    > ref.dat.filtered <- cleanup.genes(ref.dat= ref.dat,
                                        species="hs",
                                        gene.type=c("RB","chrM","chrX","chrY"),
                                        input.type="scRNA",
                                        exp.cells=5)
```

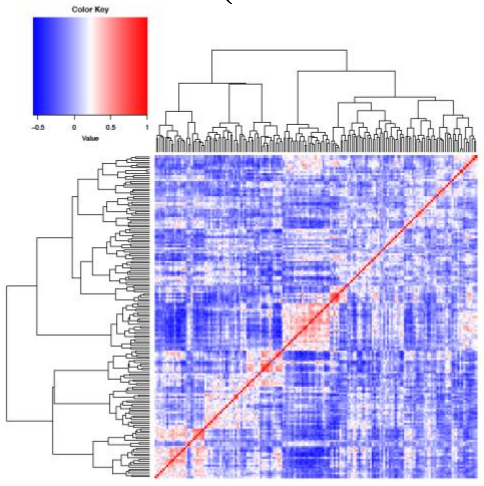
```
#console output:
       [1] "EMSEMBLE IDs detected. Cleaning up genes based on EMSEMBLE IDs."
       A total of  2959  genes from RB chrM chrX chrY  have been excluded
       A total of  10856  lowly expressed genes have been excluded

# To save memory, we can clean up data that are not used for BayesPrism.
# This step can help prevent memory overflow for large datasets, in particular Visium data
```

# All other files in the workspace be removed, make sure to save them if needed.

```
       > rm(list=setdiff(ls(), c("ref.dat.filtered", "X","meta")))
       > gc()

#run BayesPrism
      > tcga.ted <-  run.Ted  (ref.dat= ref.dat.filtered,
                                              X=X,
                                              cell.type.labels= meta$cell.type,
                                              cell.subtype.labels= meta$cell.subtype,
                                              tum.key="tumor",
                                              input.type="scRNA",
                                              n.cores=20,
                                              pdf.name="tcga.tumor")

# To run BayesPrism on collapsed gene expression profiles:
      > ref.gep.filtered <- cleanup.genes(ref.dat= ref.gep,
                                                    species="hs",
                                                    gene.type=c("RB","chrM","chrX","chrY"),
                                                    input.type="GEP")
#console output:
      A total of  4218  genes from RB chrM chrX chrY  have been excluded
      A total of  13806  lowly expressed genes have been excluded

#assign cell type/subtype labels
      > cell.subtype.labels <- rownames(ref.gep.filtered)
      > cell.type.labels <- cell.subtype.labels
      > cell.type.labels[grepl("tumor", cell.type.labels)] <- "tumor"

#run BayesPrism
      > tcga.ted <-  run.Ted  (ref.dat= ref.gep.filtered,
                                              X=X,
                                              cell.type.labels= cell.type.labels,
                                              cell.subtype.labels= cell.subtype.labels,
                                              tum.key="tumor",
                                              input.type="GEP",
                                              n.cores=20,
                                              pdf.name="tcga.tumor")
```

```
#to extract output from tcga.ted

#access cell type fractions
> tcga.ted$res$first.gibbs.res$gibbs.theta  #Initial estimates of fraction for all cell subtypes in
each bulk sample.
> tcga.ted$res$first.gibbs.res$ theta.merged  #Initial estimates of fraction for all cell types in
each bulk sample.
> tcga.ted$res$ final.gibbs.theta  # theta: the updated estimates of cell type fraction


#access gene expression (raw reads scale)
> tcga.ted$res$first.gibbs.res$Znkg # estimates of the mean of posterior read count for each cell
subtype in each bulk sample.
> tcga.ted$res$first.gibbs.res$Znkg.merged # estimates of the mean of posterior read count for
each cell type (merged across subtypes) in each bulk sample.
> tcga.ted$res$first.gibbs.res$ Z.tum  # the mean count of tumor expression in each bulk sample

#access gene expression (normalized reads)
> tcga.ted$res$first.gibbs.res $ Zkg.tum.norm # the depth-normalized count of tumor expression
in each bulk sample (the zero count genes adjusted to the same small value for each sample)
> tcga.ted$res$first.gibbs.res $ Zkg.tum.vst # the variance stabilizing transformed count of tumor
expression in each bulk sample (by the vst function in DESeq2)
> tcga.ted$res$phi.env #the batch corrected non-malignant cell expression

#other information
> tcga.ted$res$cor.mat #the correlation heatmap of tumor expressions across bulk samples
```

#The correlation (between 169 TCGA-GBM samples) heatmap generated by run.Ted:



2) Learning embeddings of tumor expression. We usually recommend users to provide phi.tum and choose an optimum K using NMF (as used in the paper).

Note that embedding learning can be computationally intensive, which may take ~10 hours to complete. Initializing phi.tum using NMF may speed up the inference.

```
library(NMF)
```

```
#extract the normalized inferred tumor expression profile
```

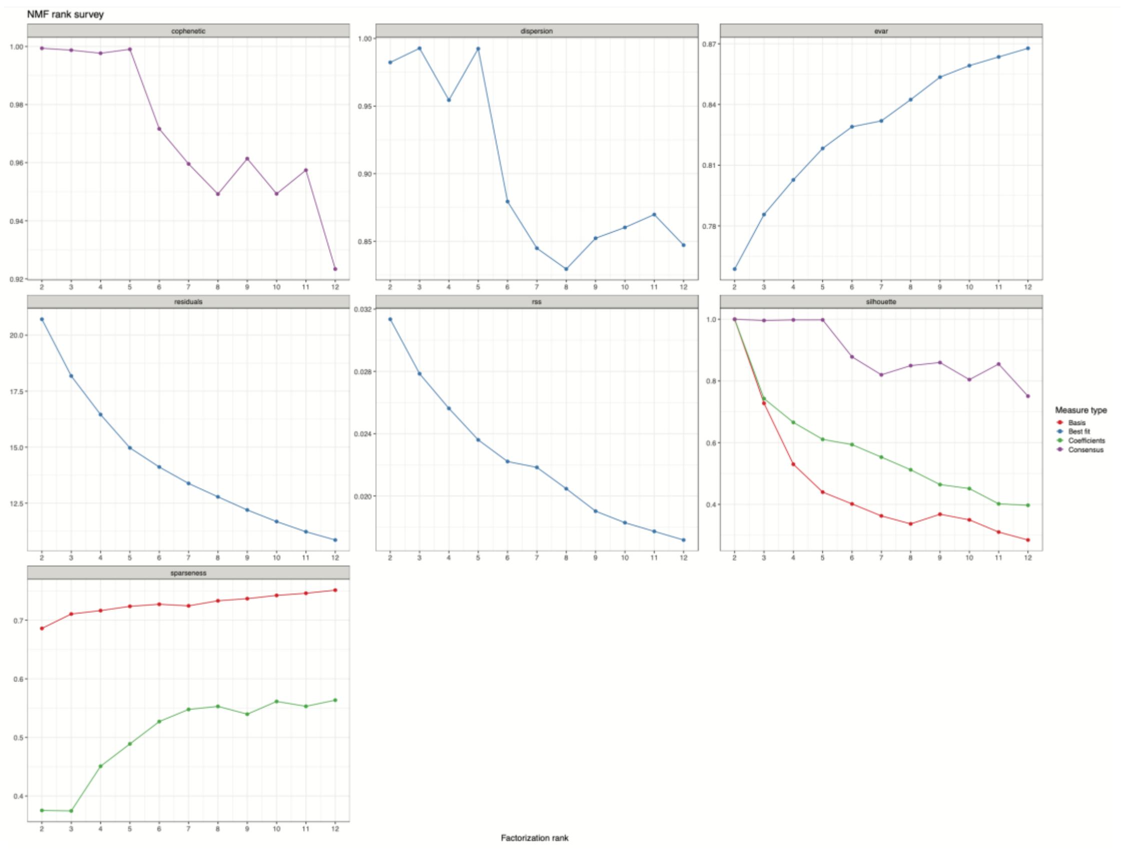
```
Z.tum.norm <- t(tcga.ted$res$first.gibbs.res$Zkg.tum.norm)
```

```
#scanning through multiple Ks using NMF, and determine the optimum K.
```

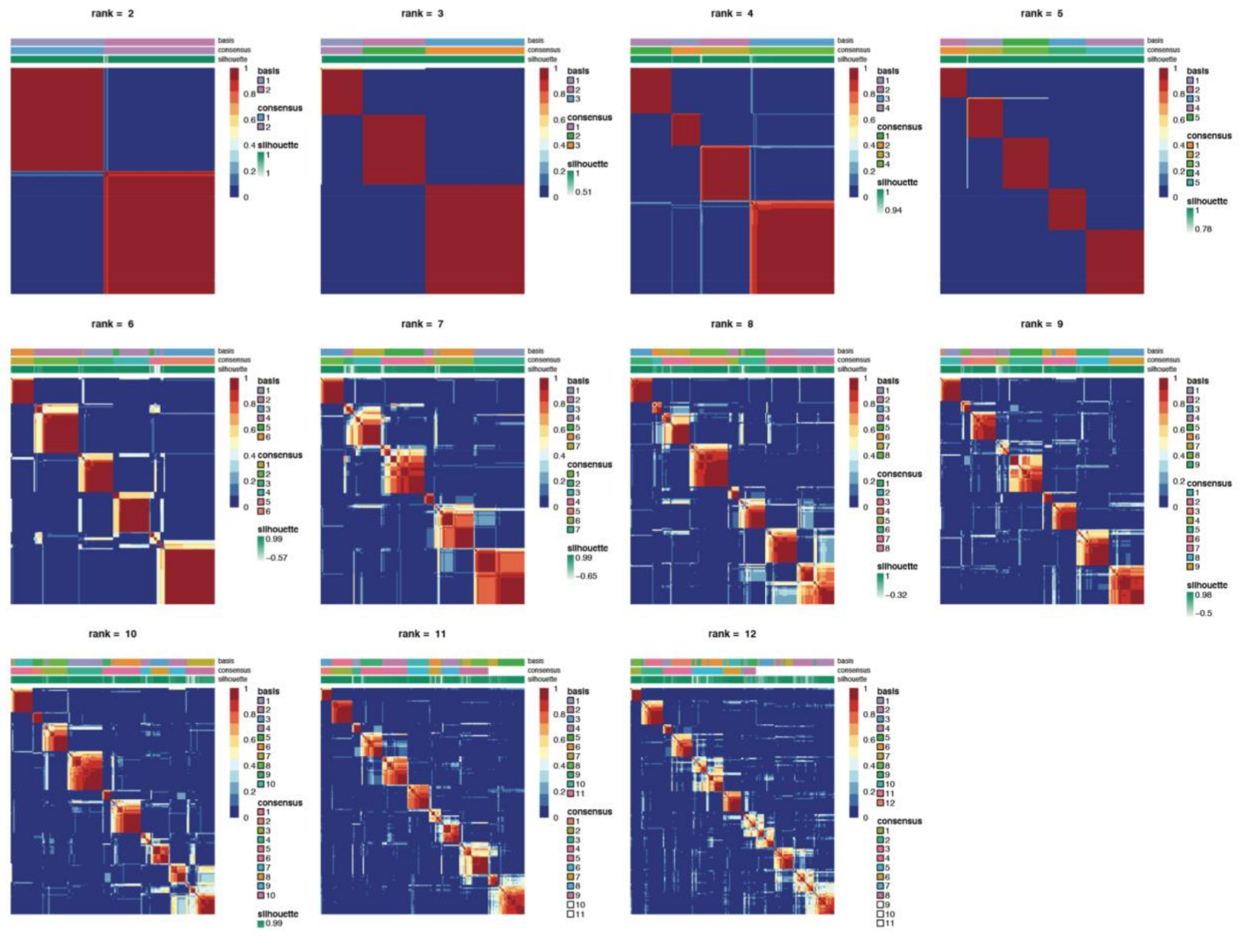
```
estim.Z.tum.norm <- nmf(Z.tum.norm, 2:12, nrun=30, seed=123456)
```

```
#plot the metrics of different Ks. Note that figures below were computed using the deconvolution setup from the BayesPrism paper, which may differ from the output of this tutorial. We are working on updating the vignette soon.
```

```
plot(estim.Z.tum.norm)
```



```
consensusmap(estim.Z.tum.norm, labCol=NA, labRow=NA,cexRow=2,cexCol=2)
```



```
#The optimum K=5 is determined from the metrics above. Now let’s perform a final run at K=5,
and optimize using 200 random initializations.

res5 <- nmf(Z.tum.norm, 5 , nrun=200, .opt='vp50', seed=123456)
phi.tum5 <- t(basis(res5))


# run embedding learning using the phi.tum supplied by NMF.
tcga.ebd.res.k5 <- learn.embedding.withPhiTum  (ted.res = tcga.gbm.ted,
                                                 phi.tum = phi.tum5,
                                                 EM.maxit=50,
                                                 n.cores =50,
                                                 compute.posterior=T)
```

#to extract output from tcga.ebd.res.k5

tcga.ebd.res.k5$ theta.all # The fractions associated with tumor bases (first K.tum columns) and stromal cells.

tcga.ebd.res.k5$ opt.phi.hat.tum # the inferred expression profile of each tumor pathways, referred to as eta in the TED paper

tcga.ebd.res.k5$ log.posterior #log posterior of each EM cycle. You may check convergence using this. EM typically converges within 50 cycles.

3) Learning embeddings of tumor expression at K=4, with pathways initialized by hierarchical clustering. This setup does not need NMF, but requires a user defined K.

```
#
>  tcga.ebd.res.k4 <- learn.embedding.Kcls  (ted.res = tcga.ted,
                                              K.vec = 4,
                                              EM.maxit=50,
                                              n.cores =20)
```

#to extract output from tcga.ebd.res.k4

tcga.ebd.res.k4$ theta.all # The fractions associated with tumor bases (first K.tum columns) and stromal cells.

tcga.ebd.res.k4$ opt.phi.hat.tum # the inferred expression profile of each tumor pathways, referred to as eta in the TED paper

tcga.ebd.res.k4$ log.posterior #log posterior of each EM cycle, if compute.posterior=T (default is not computed)