Package **CompSign**

Lena Morrill

2022

CompSign is a toolkit for differential abundance analysis of mutational signatures using a mixed effects Dirichlet-multinominal model (or simpler variations). The compositional nature of mutational signature exposures has often been overlooked but has important implications, as the analyses must be done in relative terms.

Contents

1 Installation

`CompSign` can be installed as usual from github:

```
library(devtools)
devtools::install_github("lm687/CompSign")
```

```
library(CompSign)

## Loading required package:  TMB
## Warning in checkMatrixPackageVersion():  Package version inconsistency detected.
## TMB was built with Matrix version 1.4.1
## Current Matrix version is 1.5.1
## Please re-install 'TMB' from source using install.packages('TMB', type = 'source') or ask
CRAN for a binary version of 'TMB' matching CRAN's 'Matrix' package
## Loading required package:  RcppEigen

library(gridExtra)
library(TMB)
# setwd(dirname(rstudioapi::getSourceEditorContext()$path))
```

2 Datasets

```
## if the folder data/ is not in github
for(i in list.files("../inst/extdata/", pattern = "*RDA", full.names = TRUE)){load(i)}
```

The package contains the following datasets of exposures of mutational signatures and metadata of the corresponding samples. These datasets are:

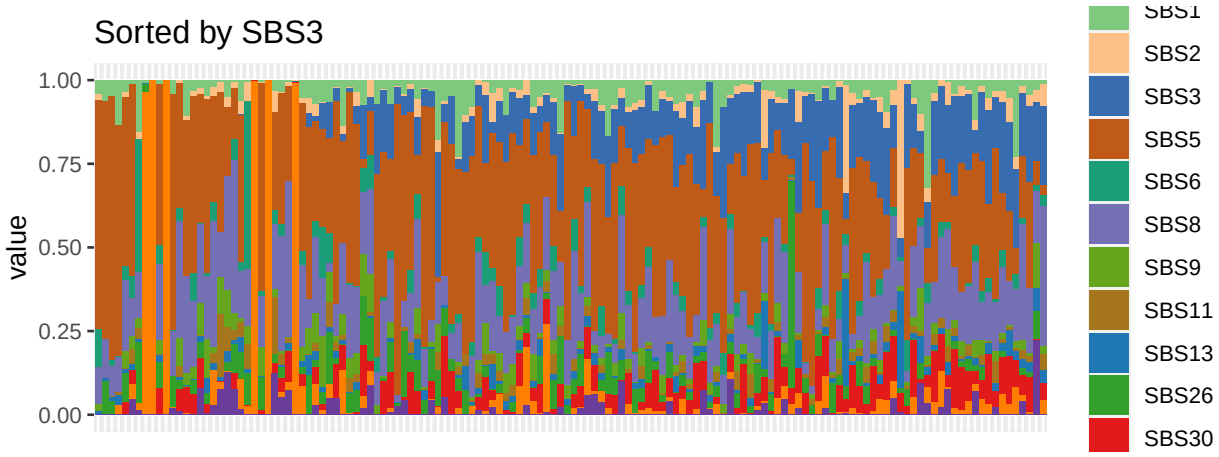
- `PancEndocrine_signaturesMSE`: Signature exposures for early and late mutations, in the PCAWG Panc-Endocrine cohort
- `ProstAdenoCA_chrom`: Signature exposures for each chromosome, in the PCAWG Prost-AdenoCA cohort

`PancEndocrine_signaturesMSE` is an object of class `sign`

```
PancEndocrine_signaturesMSE = load_PCAWG("../inst/extdata/roo/Panc-Endocrine_signaturesMSE_ROO.RDS",
                                         read_directly = T,
                                         typedata = "signaturesMSE", override_warning_X_Z = T)
PancEndocrine_signaturesMSE_v2 = load_PCAWG(ct = "Panc-Endocrine", typedata = "signaturesMSE", path_to_da
```

All samples - clonal and subclonal - sorted by increasing SBS3 exposure:

```
createBarplot(normalise_rw(non_duplicated_rows(PancEndocrine_signaturesMSE$Y)),
              order_labels = names(sort(non_duplicated_rows(PancEndocrine_signaturesMSE$Y)[,'SBS3'],
                                        decreasing = F)), remove_labels=T)+ggtitle('Sorted by SBS3')
```



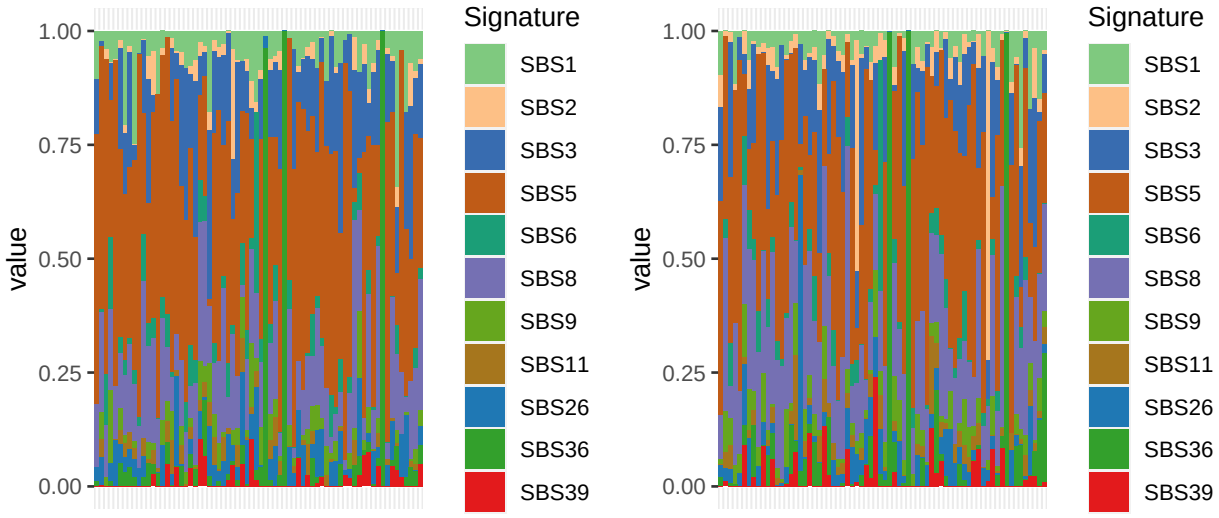
We create a simplified object containing exposures of fewer signatures (i.e. a subcomposition of the original signature vectors):

```
simplified_object <- give_subset_sigs_TMBobj(PancEndocrine_signaturesMSE,
                                             sigs_to_remove = c('SBS13', 'SBS17a', 'SBS17b', 'SBS30'))
```

The clonal and subclonal exposures are, respectively, the two barplots below:

```
do.call('grid.arrange', list(grobs=lapply(split_matrix_in_half(simplified_object$Y), function(i) createBa

## Creating plot... it might take some time if the data are large. Number of samples: 70
## Creating plot... it might take some time if the data are large. Number of samples: 70
```



3 Running the model for differential abundance

Running the model `diagREDMsinglelambda` with the dataset `simplified_object`.

```
diagDM_no_small_sigs <- wrapper_run_TMB(object = simplified_object,
                                        model = "diagREDMsinglelambda", use_nlminb=T, smart_init_vals=F)
```

These are the resulting object with the estimates and their standard deviations

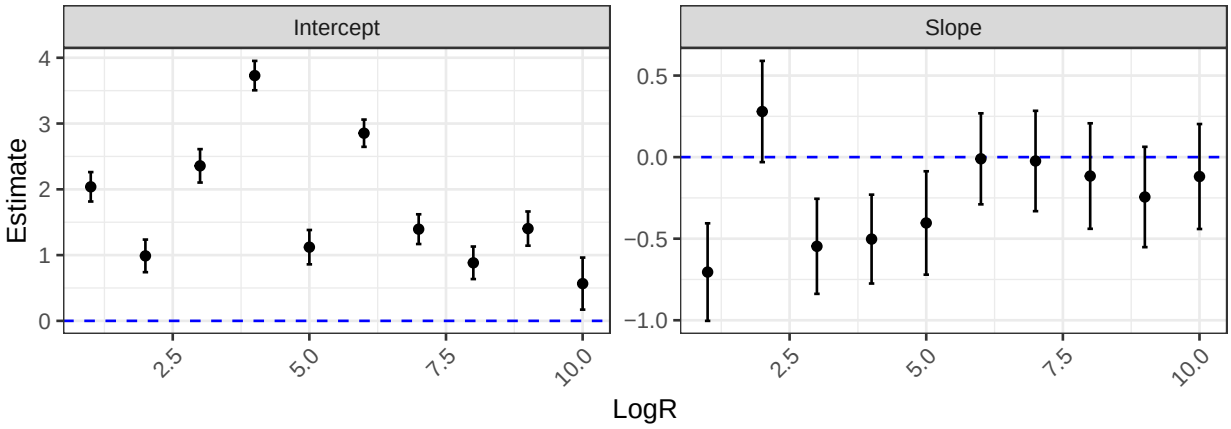
```
diagDM_no_small_sigs

## sdreport(.) result
##                Estimate   Std. Error
## beta         2.03812513   0.22363829
## beta        -0.70472748   0.29906045
## beta         0.98802357   0.24806637
## beta         0.27961604   0.31040576
## beta         2.35676410   0.25392933
## beta        -0.54677974   0.29139012
## beta         3.72918699   0.22353468
## beta        -0.50253512   0.27243310
## beta         1.12058231   0.26119873
## beta        -0.40364725   0.31694127
## beta         2.85330720   0.20725010
## beta        -0.01037487   0.27903239
## beta         1.39426509   0.22601114
## beta        -0.02361834   0.30764120
```

```
## beta         0.88267693   0.24737635
## beta        -0.11595383   0.32338020
## beta         1.40307627   0.26009578
## beta        -0.24433361   0.30748574
## beta         0.56649763   0.39571192
## beta        -0.11888545   0.32176697
## logs_sd_RE  -1.47068720   0.45938067
## logs_sd_RE  -0.88161995   0.33990127
## logs_sd_RE   0.16808990   0.25485304
## logs_sd_RE  -0.49538635   0.23244282
## logs_sd_RE  -0.19674461   0.30654573
## logs_sd_RE -11.55410912 369.39612166
## logs_sd_RE -11.65166157 253.15048729
## logs_sd_RE  -1.26242573   0.53884196
## logs_sd_RE   0.03004867   0.26234854
## logs_sd_RE   1.78120058   0.22931017
## log_lambda   2.85816323   0.05955504
## Maximum gradient component: 0.007163403
```

These are the betas for this model

```
plot_betas(diagDM_no_small_sigs)
```



and the p-value indicating differential abundance

```
wald_TMB_wrapper(diagDM_no_small_sigs)
```

```
##              [,1]
## [1,] 2.831261e-07
```

3.1 Other models

Other models can be run as follows:

```
res <- wrapper_run_TMB(object = simplified_object,
                       model = "diagREDMsinglelambda", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "diagRE_DM", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "diagRE_M", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "FEDMsinglelambda", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "FE_DM", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "fullREDMsinglelambda", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "fullRE_DMonefixedlambda", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "fullRE_DM", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "fullRE_M", use_nlminb=T, smart_init_vals=F)
res <- wrapper_run_TMB(object = simplified_object,
                       model = "singleRE_DM", use_nlminb=T, smart_init_vals=F)
```