

Package **CompSign**

Lena Morrill

October 2017

CompSign is a package for yadayada... overlooked that mutational signatures are compositional in nature yadayada. The reference manual can be found [here](#).

```
knitr::opts_chunk$set(cache = FALSE)
```

```
## This chunk was last ran in  
timestamp()
```

```
## ##----- Wed Nov 7 09:53:08 2018 -----##
```

```
## install latest version
```

```
library(devtools)
```

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

```
## Downloading GitHub repo lm687/CompSign@master
```

```
##
```

```
##
```

```
checking for file '/private/var/folders/22/gyg0hl2s7f73m8f051b35080qz1b0n/T/RtmpQrj6E4/re
```

```
v checking for file '/private/var/folders/22/gyg0hl2s7f73m8f051b35080qz1b0n/T/RtmpQrj6E4/re
```

```
##
```

```
- preparing 'CompSign':
```

```
##
```

```
checking DESCRIPTION meta-information ...
```

```
v checking DESCRIPTION meta-information
```

```
##
```

```
- excluding invalid files
```

```

##
    Subdirectory 'man' contains invalid file names:
##      'summary_R00.txt'
##

- checking for LF line-endings in source and make files and shell scripts
##

- checking for empty or unneeded directories
##

- building 'CompSign_0.1.0.tar.gz'
##

Warning: invalid uid value replaced by that for user 'nobody'
##

##

library(CompSign)
library(compositions)

## Loading required package: tensorA
##
## Attaching package: 'tensorA'
## The following object is masked from 'package:base':
##
##      norm
## Loading required package: robustbase
## Loading required package: energy
## Loading required package: bayesm
## Welcome to compositions, a package for compositional data analysis.
## Find an intro with "? compositions"
##
## Attaching package: 'compositions'
## The following objects are masked from 'package:stats':
##
##      cor, cov, dist, var
## The following objects are masked from 'package:base':
##
##      %*%, scale, scale.default

```

```
## This chunk was last ran in
timestamp()

## ##----- Wed Nov 7 09:53:38 2018 -----##

#####
##### Dummy data #####
#####

### Example of matrix transformed into sign object
input_dummy <- matrix(runif(100), 4)
colnames(input_dummy) <- paste0('s', 1:25); rownames(input_dummy) <- paste0('sam', 1:4)
sign_dummy <- to_sign(input_dummy)
```

1 Summarise the signature matrix

```
## This chunk was last ran in
timestamp()

## ##----- Wed Nov 7 09:53:38 2018 -----##

add_together_matrix(sign_dummy)

## An object of class "sign"
## Slot "id":
## [1] "input_dummy"
##
## Slot "id_samples":
## [1] "sam1" "sam2" "sam3" "sam4"
##
## Slot "id_signatures":
## [1] "s1" "s2" "s3" "s4" "s5" "s6" "s7" "s8" "s9" "s10" "s11"
## [12] "s12" "s13" "s14" "s15" "s16" "s17" "s18" "s19" "s20" "s21" "s22"
## [23] "s23" "s24" "s25"
##
## Slot "count_matrix":
##          s1          s2          s3          s4          s5          s6
## sam1 0.6130077 0.6201001 0.02548431 0.9164084 0.2908617 0.07488885
## sam2 0.9900551 0.3453004 0.80846802 0.6432645 0.7316618 0.92458896
## sam3 0.2915941 0.4309228 0.75626895 0.5831739 0.8232874 0.15455563
## sam4 0.7403550 0.3917636 0.78327104 0.4309776 0.1752640 0.94668388
##          s7          s8          s9          s10          s11          s12
## sam1 0.7811775 0.5824201 0.7072237 0.86595023 0.3520402 0.4358718
## sam2 0.5869432 0.6389264 0.3450403 0.07703436 0.4722528 0.7467841
```

```
## sam3 0.8196181 0.3919323 0.7823582 0.74613238 0.5338816 0.0846173
## sam4 0.7044953 0.3257368 0.9367393 0.99475854 0.9464276 0.5732689
##          s13          s14          s15          s16          s17          s18
## sam1 0.9754523 0.6971351 0.5247873 0.5126980 0.5080613 0.8311513
## sam2 0.5771914 0.9870072 0.4881304 0.8546315 0.5936254 0.6873601
## sam3 0.7392869 0.5104545 0.1886795 0.3496769 0.6928111 0.6738743
## sam4 0.9027598 0.3239889 0.9622762 0.2661097 0.3346391 0.6533321
##          s19          s20          s21          s22          s23          s24
## sam1 0.09571556 0.1549136 0.2458252 0.3138197 0.8134497 0.6355442
## sam2 0.42322630 0.7672090 0.9687606 0.3592505 0.9477584 0.7051245
## sam3 0.08214040 0.9420649 0.8997512 0.2251877 0.4940255 0.9120137
## sam4 0.91560367 0.1443106 0.7358621 0.9465534 0.3667558 0.2882320
##          s25
## sam1 0.07286385
## sam2 0.26951424
## sam3 0.64187211
## sam4 0.90365832
##
## Slot "modified":
## [1] TRUE

results_sumarise <- summarise(add_together_matrix(sign_dummy))
results_sumarise$General

## [1] "Object of class sign"
```

2 Linear model for numerical predictors

```
## This chunk was last ran in
timestamp()

## ##----- Wed Nov 7 09:53:38 2018 -----##

tmp_merged_compositional <- new("merged_compositional",
                                id='adas',
                                id_samples=paste0("sam", 1:30),
                                id_signatures= c('s1', 's2', 's3', 's4'), ## signature names
                                count_matrix=MCMCpack::rdirichlet(30, c(1,1,1,1)),
                                df=data.frame(a=sample(1:1e4, 30), b=rep(10, 30)))
comp_lm(tmp_merged_compositional)

## [[1]]
## Response Y1 :
##
```

```
## Call:
## lm(formula = Y1 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8146 -0.7776  0.1236  0.6441  3.1480
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value
## (Intercept)      3.258e-01  4.733e-01   0.688
## as.matrix((x@df)[, indices_predictor])a -1.047e-04  8.822e-05  -1.187
## as.matrix((x@df)[, indices_predictor])b          NA          NA          NA
##              Pr(>|t|)
## (Intercept)          0.497
## as.matrix((x@df)[, indices_predictor])a    0.245
## as.matrix((x@df)[, indices_predictor])b          NA
##
## Residual standard error: 1.137 on 28 degrees of freedom
## Multiple R-squared:  0.04788, Adjusted R-squared:  0.01388
## F-statistic: 1.408 on 1 and 28 DF,  p-value: 0.2453
##
##
## Response Y2 :
##
## Call:
## lm(formula = Y2 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4838 -1.1343  0.1275  1.1752  2.0129
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value
## (Intercept)    -1.0767141  0.5460040  -1.972
## as.matrix((x@df)[, indices_predictor])a  0.0002004  0.0001018   1.969
## as.matrix((x@df)[, indices_predictor])b          NA          NA          NA
##              Pr(>|t|)
## (Intercept)      0.0586 .
## as.matrix((x@df)[, indices_predictor])a  0.0589 .
## as.matrix((x@df)[, indices_predictor])b          NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.312 on 28 degrees of freedom
## Multiple R-squared:  0.1216, Adjusted R-squared:  0.09027
```

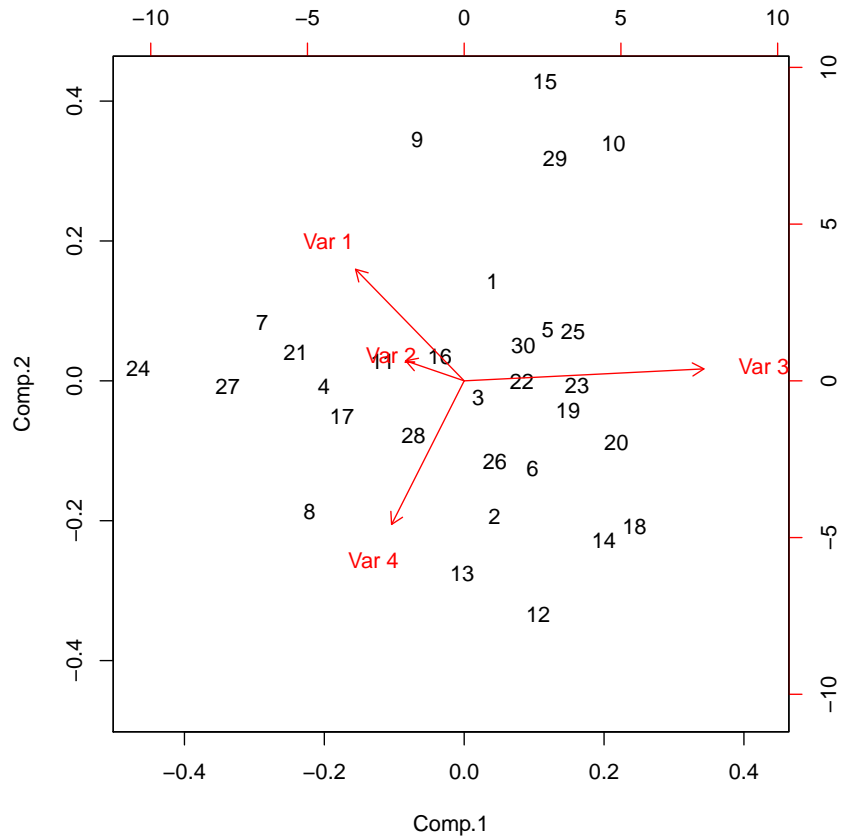
```
## F-statistic: 3.878 on 1 and 28 DF, p-value: 0.0589
##
##
## Response Y3 :
##
## Call:
## lm(formula = Y3 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22997 -0.43382  0.07819  0.67836  1.85067
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value
## (Intercept)    7.110e-01  3.814e-01   1.864
## as.matrix((x@df)[, indices_predictor])a -1.294e-04  7.109e-05  -1.821
## as.matrix((x@df)[, indices_predictor])b          NA          NA          NA
##              Pr(>|t|)
## (Intercept)    0.0728 .
## as.matrix((x@df)[, indices_predictor])a    0.0794 .
## as.matrix((x@df)[, indices_predictor])b          NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9165 on 28 degrees of freedom
## Multiple R-squared:  0.1058, Adjusted R-squared:  0.07391
## F-statistic: 3.315 on 1 and 28 DF, p-value: 0.07937
```

3 Importing data

```
## This chunk was last ran in
timestamp()

## ##----- Wed Nov  7 09:53:39 2018 -----##

biplot(princomp(acomp(MCMCpack::rdirichlet(30, rep(1, 4)))))
```



4 Other

1. Test for normality as follows:

```
## This chunk was last ran in
timestamp()

## ##----- Wed Nov 7 09:53:39 2018 -----##

data(two_normal_pops)

## Warning in data(two_normal_pops): data set 'two_normal_pops'
not found

par(mfrow=c(1,2))
qqnorm.acomp(acomp(two_normal_pops@count_matrix), pch=19, cex=0.2)
```

```
## Error in acomp(two_normal_pops@count_matrix): object 'two_normal_pops'
not found

qqnorm.acomp(acomp(two_normal_pops@count_matrix[1:1000,]), pch=19, cex=0.2, plot.it=FALSE)

## Error in acomp(two_normal_pops@count_matrix[1:1000, ]): object
'two_normal_pops' not found
```

5 Clustering of samples

5.1 Testing hypotheses about two populations

We might have our samples split into two categories; e.g. sex. As in Aithison 1986[], I follow a hierarchy of alternative hypotheses, from least to most complex.

Our first question is whether two populations have the same covariance and structure and center (i.e. if there is any distributional difference)

```
## This chunk was last ran in
timestamp()

## ##----- Wed Nov 7 09:53:40 2018 -----##

##TODO!!
```

The next is whether the populations have a different center:

```
## This chunk was last ran in
timestamp()

## ##----- Wed Nov 7 09:53:40 2018 -----##

## This dataset includes the two components above, as well as four others
## (a total of seven)
data("two_normal_pops_extended")

## Warning in data("two_normal_pops_extended"): data set 'two_normal_pops_extended'
not found

## Data from the Landscape... paper
data("Breast560")

## Warning in data("Breast560"): data set 'Breast560' not found

wrapper_compare_populations <- function(predictors, response, ...){
  if(length(unique(response)) == 2){
    tmp <- compare_populations(predictors, response, ...)
    tmp <- tmp$info[1:2]
    tmp
  }
}

x <- do.call('rbind', lapply(1:ncol(metadata(Breast560)),
  function(k){
    wrapper_compare_populations(predictors = count_matrix(Breast560),
      response = metadata(Breast560)[,k])
  })
))

## Error in metadata(Breast560): object 'Breast560' not found
x

## Error in eval(expr, envir, enclos): object 'x' not found
```

6 Data for 560 breast cancer patients

Data from 560 breast cancer patients is available as part of the document as well:

```
data("Breast560")

## Warning in data("Breast560"): data set 'Breast560' not found

metadata(Breast560)[1:4,1:5]

## Error in metadata(Breast560): object 'Breast560' not found

count_matrix(Breast560)[1:4,1:5]

## Error in count_matrix(Breast560): object 'Breast560' not found
```

Not sure if this is correct

```
source("~/Documents/CantabPhD/CDA_in_Cancer/code/functions/various_functions.R")

## Warning in file(filename, "r", encoding = encoding): cannot open
file '/Users/morril01/Documents/CantabPhD/CDA_in_Cancer/code/functions/various_functions.R':
No such file or directory
## Error in file(filename, "r", encoding = encoding): cannot open
the connection

plotPCA(irlr(count_matrix(Breast560)), pch=4, col='blue')

## Error in plotPCA(irlr(count_matrix(Breast560)), pch = 4, col = "blue"):
could not find function "plotPCA"
```

6.1 (ongoing) test for equality

```
comp.test(x = count_matrix(Breast560),
          ina = as.numeric(as.factor(metadata(Breast560)$final.ER)),
          test = "james", R = 0)

## Error in comp.test(x = count_matrix(Breast560), ina = as.numeric(as.factor(metadata(Breas
: could not find function "comp.test"
```

7 Data for 12k TCGA samples, with ovarian cancer-derived CNA signatures

```
timestamp()

## ##----- Wed Nov 7 09:53:40 2018 -----##

data("CNA_12K_TCGA")

## Warning in data("CNA_12K_TCGA"): data set 'CNA_12K_TCGA' not found

dim(metadata(CNA_12K_TCGA))

## Error in metadata(CNA_12K_TCGA): object 'CNA_12K_TCGA' not found

dim(count_matrix(CNA_12K_TCGA))

## Error in count_matrix(CNA_12K_TCGA): object 'CNA_12K_TCGA' not found
```