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 Oct 24 09:51:46 2018 ------##

## install latest version
library(devtools)
devtools::install_github("lm687/CompSign")

## Skipping install of 'CompSign' from a github remote, the SHA1 (ae95502a)
has not changed since last install.
##  Use 'force = TRUE' to force installation

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 Oct 24 09:51:48 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 Oct 24 09:51:48 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 0.6764623 0.2729444 0.3436545 0.1002153 0.06561845 0.03094423
## sam2 0.4520095 0.4091145 0.1040464 0.3355779 0.68618118 0.44924840
## sam3 0.9363835 0.3928727 0.4027341 0.2753264 0.65121843 0.66822314
## sam4 0.5446427 0.1544315 0.2462713 0.3190422 0.43053817 0.77015991
##             s7        s8        s9       s10       s11       s12
## sam1 0.3532618 0.8480522 0.7084803 0.8895526 0.6296052 0.4025335
## sam2 0.8411673 0.7207060 0.8103149 0.1315620 0.8245883 0.5702180
## sam3 0.9850511 0.5855240 0.5503083 0.8914260 0.8225037 0.9071219
## sam4 0.7384766 0.2273273 0.7245602 0.1884538 0.2813367 0.4807784
##             s13        s14       s15       s16        s17       s18
## sam1 0.61211950 0.58389072 0.7192076 0.4294944 0.16047355 0.6105783
## sam2 0.01719359 0.16899129 0.7834863 0.1660177 0.45931066 0.1057023
## sam3 0.94706416 0.02446017 0.7819861 0.1789141 0.29158493 0.3087867
## sam4 0.51996204 0.42566369 0.6535919 0.7844009 0.02664156 0.6373981
##             s19       s20        s21        s22        s23       s24
## sam1 0.30225377 0.9639017 0.79162225 0.02198544 0.08396833 0.6812706
## sam2 0.07004629 0.7094089 0.70866945 0.43777464 0.78886873 0.6356862
## sam3 0.27966923 0.8772923 0.04130497 0.82702003 0.41161577 0.2505033
## sam4 0.65060962 0.9220265 0.16407478 0.60034012 0.54354101 0.4179587
##              s25
## sam1 0.193195837
## sam2 0.036536185
## sam3 0.607445674
## sam4 0.006969308
##
## 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 Oct 24 09:51:48 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
## -1.7305 -0.6656 -0.1989  0.6349  2.8453
##
## Coefficients: (1 not defined because of singularities)
##                                          Estimate Std. Error t value
## (Intercept)                             2.308e-01  3.747e-01   0.616
## as.matrix((x@df)[, indices_predictor])a -2.947e-05  7.314e-05  -0.403
## as.matrix((x@df)[, indices_predictor])b         NA         NA      NA
##                                         Pr(>|t|)
## (Intercept)                                0.543
## as.matrix((x@df)[, indices_predictor])a    0.690
## as.matrix((x@df)[, indices_predictor])b       NA
##
## Residual standard error: 1.134 on 28 degrees of freedom
## Multiple R-squared:  0.005766,Adjusted R-squared:  -0.02974
## F-statistic: 0.1624 on 1 and 28 DF,  p-value: 0.69
##
##
## Response Y2 :
##
## Call:
## lm(formula = Y2 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8634 -0.5298  0.2247  0.7899  1.9961
##
## Coefficients: (1 not defined because of singularities)
##                                          Estimate Std. Error t value
## (Intercept)                             1.438e-01  3.704e-01   0.388
## as.matrix((x@df)[, indices_predictor])a -3.753e-05  7.231e-05  -0.519
## as.matrix((x@df)[, indices_predictor])b         NA         NA      NA
##                                         Pr(>|t|)
## (Intercept)                                0.701
```

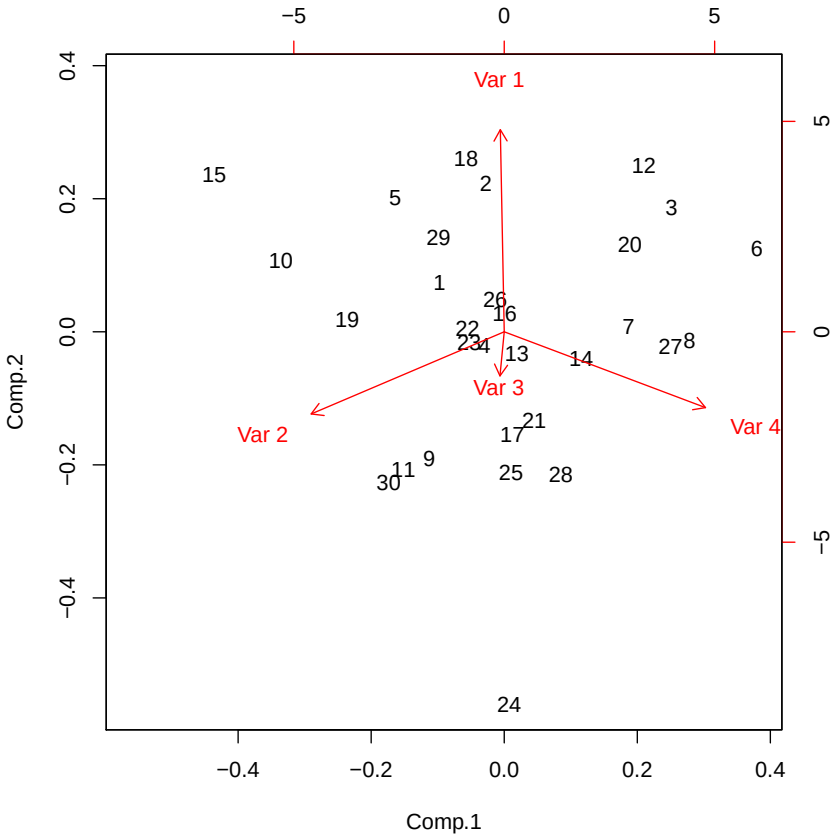
```
## as.matrix((x@df)[, indices_predictor])a    0.608
## as.matrix((x@df)[, indices_predictor])b       NA
##
## Residual standard error: 1.121 on 28 degrees of freedom
## Multiple R-squared:  0.00953,Adjusted R-squared:  -0.02584
## F-statistic: 0.2694 on 1 and 28 DF,  p-value: 0.6078
##
##
## Response Y3 :
##
## Call:
## lm(formula = Y3 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6174 -0.7001  0.2863  0.8532  2.6761
##
## Coefficients: (1 not defined because of singularities)
##                                         Estimate Std. Error t value
## (Intercept)                            -4.560e-01  4.850e-01  -0.940
## as.matrix((x@df)[, indices_predictor])a  8.054e-06  9.468e-05   0.085
## as.matrix((x@df)[, indices_predictor])b         NA         NA      NA
##                                        Pr(>|t|)
## (Intercept)                               0.355
## as.matrix((x@df)[, indices_predictor])a    0.933
## as.matrix((x@df)[, indices_predictor])b       NA
##
## Residual standard error: 1.468 on 28 degrees of freedom
## Multiple R-squared:  0.0002584,Adjusted R-squared:  -0.03545
## F-statistic: 0.007236 on 1 and 28 DF,  p-value: 0.9328
```

3 Importing data

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

## ##------ Wed Oct 24 09:51:49 2018 ------##

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



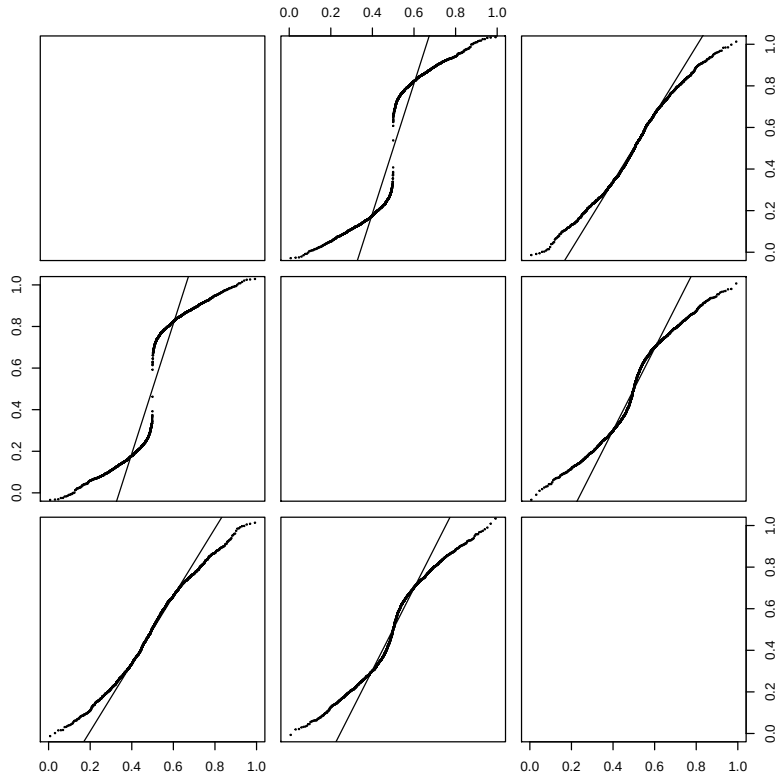
4 Other

1. Test for normality as follows:

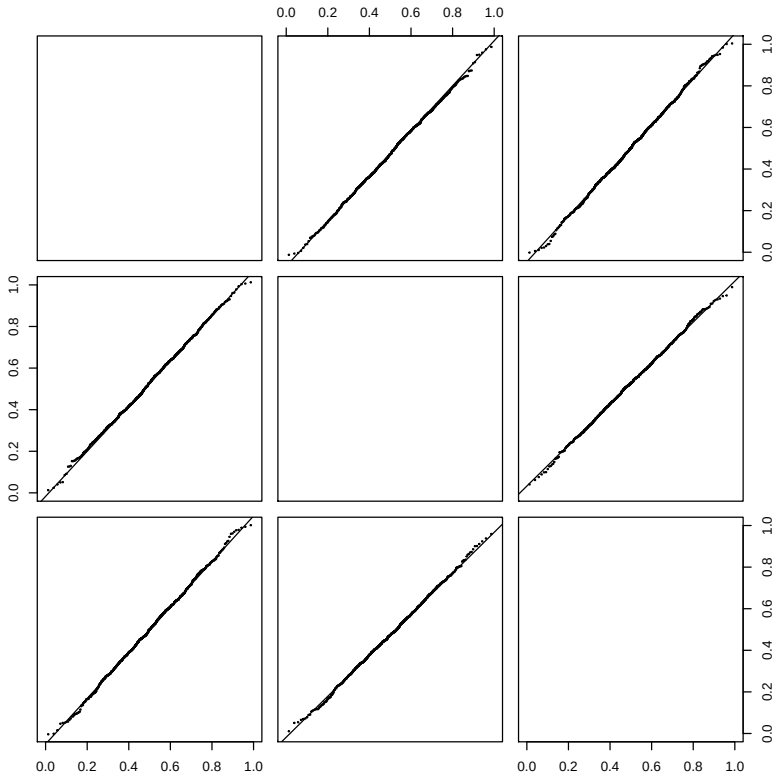
```
## This chunk was last ran in
timestamp()

## ##------ Wed Oct 24 09:51:49 2018 ------##

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




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


0.0 0.2 0.4 0.6 0.8 1.0

4.1 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 Oct 24 09:51:50 2018 ------##

##TODO!!
```

The next is whether the populations have a different center:

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

## ##------ Wed Oct 24 09:51:50 2018 ------##

## This dataset includes the two components above, as well as four others
## (a total of seven)
data("two_normal_pops_extended")
compare_populations(predictors = two_normal_pops_extended@count_matrix,
                    response = two_normal_pops_extended@df[,1])

## Loading required package:  Compositional
##
## Attaching package:  'Compositional'
## The following object is masked from 'package:compositions':
##
##    alr
## Error in Compositional::comp.test(x = predictors[, -1], ina = tmp_response,
:  object 'result' not found
```

5 Data for 560 breast cancer patients

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

```
##continue in save_560BRCA_rda.R
```