

Package **CompSign**

Lena Morrill

October 2017

```
## knitr preferences
## no chache
```

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

## Downloading GitHub repo lm687/CompSign@master
## from URL https://api.github.com/repos/lm687/CompSign/zipball/master
## Installing CompSign
## '/Library/Frameworks/R.framework/Resources/bin/R' --no-site-file
\
## --no-environ --no-save --no-restore --quiet CMD INSTALL \
## '/private/var/folders/22/nzk7280n61jd5qrjhm5cwph0000gn/T/RtmpSMQhvs/devtools84327ba40b
\
## --library='/Library/Frameworks/R.framework/Versions/3.4/Resources/library'
\
## --install-tests
##

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
```

```
#####
##### 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

```
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 0.04858211 0.8317766 0.05256749 0.3409959 0.5204520 0.7224081
## sam2 0.50588321 0.5770588 0.92585154 0.6151534 0.6980990 0.4591961
## sam3 0.54916562 0.8884311 0.18074831 0.4806562 0.0623873 0.1348595
## sam4 0.98579154 0.5938418 0.64554445 0.2983224 0.6354810 0.2453148
##           s7      s8      s9      s10      s11      s12
## sam1 0.1330572 0.74518787 0.2781470 0.7011675 0.03245983 0.64248301
```

```
## sam2 0.2722230 0.07501031 0.1886666 0.9067959 0.31114464 0.17275723
## sam3 0.5477834 0.66165313 0.1308051 0.8027359 0.07929365 0.07382778
## sam4 0.3757639 0.50000232 0.1606468 0.1952131 0.01551060 0.80239360
##          s13          s14          s15          s16          s17          s18
## sam1 0.64639791 0.5083813 0.6308606 0.7773304 0.65004224 0.7755878
## sam2 0.87854826 0.1266760 0.9119379 0.5181562 0.94692289 0.1243538
## sam3 0.05928518 0.1794574 0.8778418 0.9441995 0.22104930 0.1547693
## sam4 0.43216915 0.4294558 0.4622663 0.2422564 0.09119018 0.6248315
##          s19          s20          s21          s22          s23          s24
## sam1 0.7065150 0.6640117 0.35042408 0.6759364 0.5638160 0.03573475
## sam2 0.9014284 0.6012351 0.70919785 0.3100722 0.1514624 0.66245107
## sam3 0.6548523 0.2860429 0.03763259 0.5321743 0.6405682 0.54845881
## sam4 0.8955548 0.9146842 0.94123500 0.5437059 0.9597871 0.20470684
##          s25
## sam1 0.7183756
## sam2 0.5841280
## sam3 0.7387282
## sam4 0.3237947
##
## 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

```
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]))
##
```

```

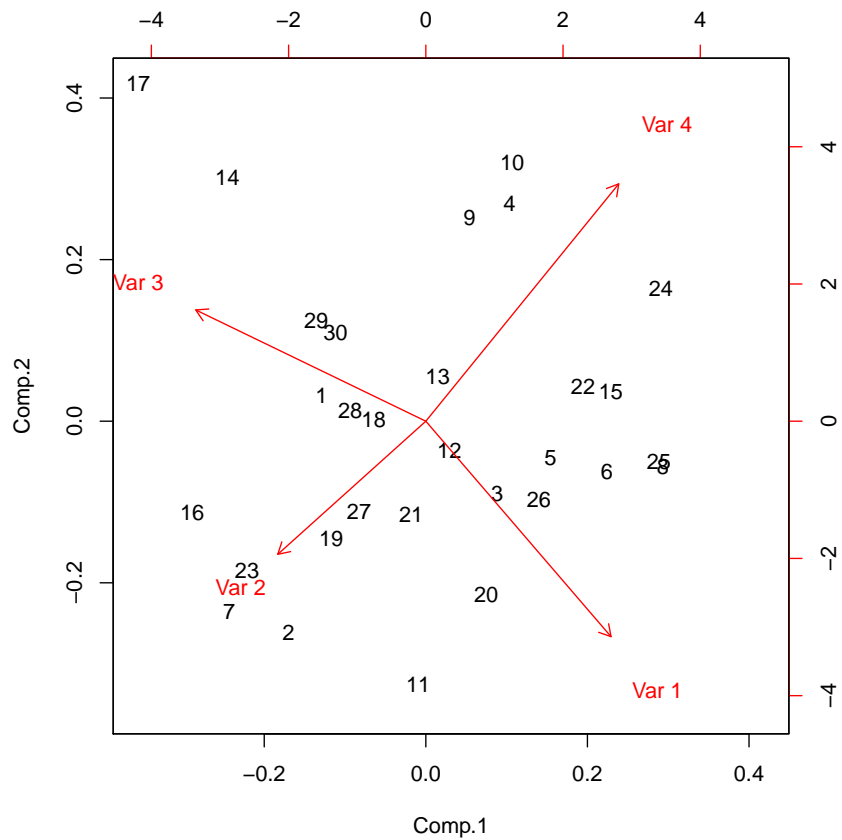
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9902 -0.8037 -0.2422  0.7749  2.2555
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error t value
## (Intercept)                   -4.108e-02  4.505e-01  -0.091
## as.matrix((x@df)[, indices_predictor])a  7.397e-06  7.329e-05   0.101
## as.matrix((x@df)[, indices_predictor])b           NA           NA           NA
##                                Pr(>|t|)
## (Intercept)                   0.928
## as.matrix((x@df)[, indices_predictor])a    0.920
## as.matrix((x@df)[, indices_predictor])b           NA
##
## Residual standard error: 1.188 on 28 degrees of freedom
## Multiple R-squared:  0.0003636, Adjusted R-squared:  -0.03534
## F-statistic: 0.01018 on 1 and 28 DF,  p-value: 0.9203
##
##
## Response Y2 :
##
## Call:
## lm(formula = Y2 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.1849 -0.9617  0.0916  0.8785  2.9278
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error t value
## (Intercept)                   2.286e-01  5.084e-01   0.450
## as.matrix((x@df)[, indices_predictor])a -4.409e-05  8.272e-05  -0.533
## as.matrix((x@df)[, indices_predictor])b           NA           NA           NA
##                                Pr(>|t|)
## (Intercept)                   0.656
## as.matrix((x@df)[, indices_predictor])a    0.598
## as.matrix((x@df)[, indices_predictor])b           NA
##
## Residual standard error: 1.341 on 28 degrees of freedom
## Multiple R-squared:  0.01005, Adjusted R-squared:  -0.02531
## F-statistic: 0.2841 on 1 and 28 DF,  p-value: 0.5982
##
##
## Response Y3 :
##

```

```
## Call:
## lm(formula = Y3 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.83013 -0.89995  0.06103  0.73338  2.73146
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error t value
## (Intercept)                   2.544e-01  4.954e-01   0.514
## as.matrix((x@df)[, indices_predictor])a -2.524e-05  8.060e-05  -0.313
## as.matrix((x@df)[, indices_predictor])b          NA          NA          NA
##                                Pr(>|t|)
## (Intercept)                   0.612
## as.matrix((x@df)[, indices_predictor])a    0.756
## as.matrix((x@df)[, indices_predictor])b          NA
##
## Residual standard error: 1.306 on 28 degrees of freedom
## Multiple R-squared:  0.003491, Adjusted R-squared:  -0.0321
## F-statistic: 0.09809 on 1 and 28 DF,  p-value: 0.7565
```

3 Importing data

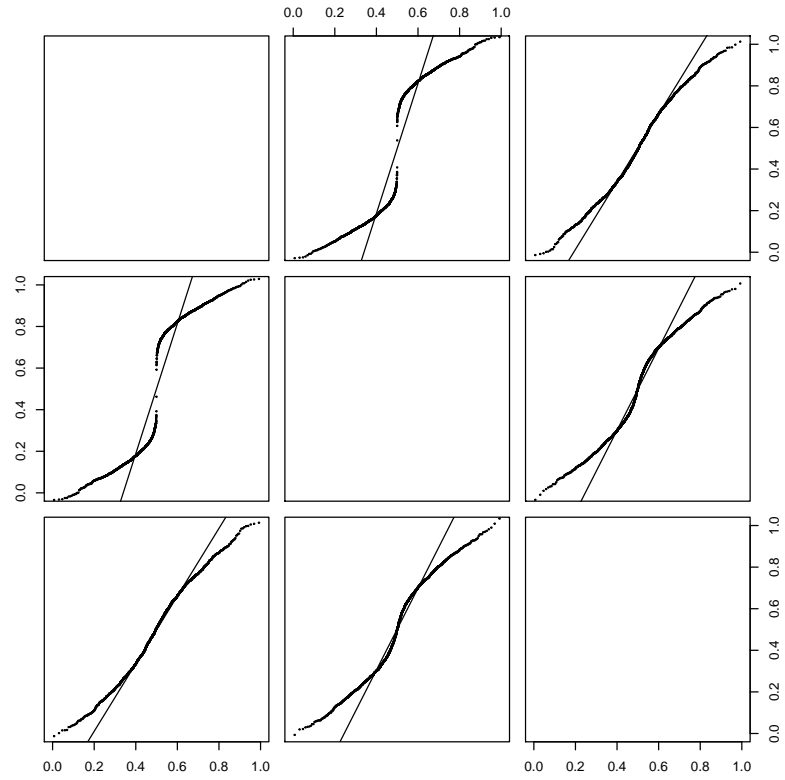
```
biplot(princomp(acompc(MCMCpack::rdirichlet(30, rep(1, 4)))))
```



4 Other

1. Test for normality as follows:

```
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)
```

