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/nzk7280n61jd5qrjhqm5cwph0000gn/T/RtmpOZuhi3/devtools14d0ed7a9f/
\
##  --library='/Library/Frameworks/R.framework/Versions/3.4/Resources/library'
\
##  --install-tests
##

library(CompSign)
```

```
############################
####### 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 0.1987315 0.5638199 0.91801670 0.6796169 0.8894069 0.3519896
## sam2 0.8128805 0.6593104 0.18514303 0.1053678 0.5611413 0.2590321
## sam3 0.6050001 0.2540923 0.25940428 0.8213403 0.5969080 0.9449074
## sam4 0.9104714 0.3458921 0.04514874 0.2078522 0.3835074 0.9169413
##             s7        s8        s9       s10       s11       s12       s13
## sam1 0.8305910 0.1972703 0.2954425 0.1842939 0.2053968 0.1337796 0.5980837
## sam2 0.5058662 0.2439515 0.1381540 0.4998761 0.9886769 0.3826237 0.9692546
## sam3 0.1657199 0.5982735 0.9320994 0.1410417 0.5063486 0.9906361 0.3989435
## sam4 0.8693266 0.3740830 0.1440873 0.6437422 0.7980912 0.3529153 0.3447031
##            s14       s15        s16        s17       s18       s19
## sam1 0.7414813 0.6337050 0.48060115 0.84909583 0.9185874 0.8105136
## sam2 0.3634842 0.4203783 0.07598475 0.86363441 0.4137671 0.8931221
## sam3 0.3259484 0.9021382 0.45760978 0.26811933 0.9363666 0.6860977
## sam4 0.5211474 0.9044891 0.31324170 0.02960415 0.8244890 0.8680906
##            s20        s21       s22        s23         s24       s25
## sam1 0.3247066 0.79162196 0.2435041 0.25040330 0.043362997 0.5839870
## sam2 0.6031947 0.65683798 0.1394422 0.86029437 0.652102866 0.3367025
## sam3 0.1124012 0.27429140 0.5805026 0.08558455 0.762270931 0.9938002
## sam4 0.5557001 0.03076965 0.3851658 0.99501856 0.001275652 0.9788973
##
## Slot "modified":
## [1] TRUE
```

```
results_sumarise <- summarise(add_together_matrix(sign_dummy))
```

```
## Error in switch(robust, pearson = {:  EXPR must be a length 1 vector
```

```
results_sumarise$General
```

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

2 Linear model for numerical predictors

```
tmp_merged_compositional <- new("merged_compositional",
                                id='adas',
                                id_samples=c("sam1", "sam2", "sam3"),
                                id_signatures= c('s1', 's2', 's3', 's4'), ## signature names
                                count_matrix=MCMCpack::rdirichlet(3, c(1,1,1,1)),
                                df=data.frame(a=c(3,4,1), b=c(10, 10, 10)))
comp_lm(tmp_merged_compositional)

## [[1]]
## Response Y1 :
##
## Call:
## lm(formula = Y1 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##       1       2       3
##  0.5052 -0.3368 -0.1684
##
## Coefficients: (1 not defined because of singularities)
##                                          Estimate Std. Error t value
## (Intercept)                               0.25050    0.85873   0.292
## as.matrix((x@df)[, indices_predictor])a -0.09151    0.29170  -0.314
## as.matrix((x@df)[, indices_predictor])b       NA         NA      NA
##                                          Pr(>|t|)
## (Intercept)                                 0.819
## as.matrix((x@df)[, indices_predictor])a    0.806
## as.matrix((x@df)[, indices_predictor])b       NA
##
## Residual standard error: 0.6301 on 1 degrees of freedom
## Multiple R-squared:  0.0896,Adjusted R-squared:  -0.8208
## F-statistic: 0.09842 on 1 and 1 DF,  p-value: 0.8065
##
##
## Response Y2 :
##
## Call:
## lm(formula = Y2 ~ as.matrix((x@df)[, indices_predictor]))
##
```

```
## Residuals:
##       1       2       3
##  0.9943 -0.6629 -0.3314
##
## Coefficients: (1 not defined because of singularities)
##                                         Estimate Std. Error t value
## (Intercept)                              0.06221    1.68995   0.037
## as.matrix((x@df)[, indices_predictor])a  0.16497    0.57405   0.287
## as.matrix((x@df)[, indices_predictor])b       NA         NA      NA
##                                         Pr(>|t|)
## (Intercept)                                0.977
## as.matrix((x@df)[, indices_predictor])a    0.822
## as.matrix((x@df)[, indices_predictor])b       NA
##
## Residual standard error: 1.24 on 1 degrees of freedom
## Multiple R-squared:  0.07629,Adjusted R-squared:  -0.8474
## F-statistic: 0.08259 on 1 and 1 DF,  p-value: 0.8218
##
##
## Response Y3 :
##
## Call:
## lm(formula = Y3 ~ as.matrix((x@df)[, indices_predictor]))
##
## Residuals:
##       1       2       3
##  1.0857 -0.7238 -0.3619
##
## Coefficients: (1 not defined because of singularities)
##                                         Estimate Std. Error t value
## (Intercept)                             -1.3904     1.8453  -0.753
## as.matrix((x@df)[, indices_predictor])a  0.4634     0.6268   0.739
## as.matrix((x@df)[, indices_predictor])b       NA         NA      NA
##                                         Pr(>|t|)
## (Intercept)                                0.589
## as.matrix((x@df)[, indices_predictor])a    0.595
## as.matrix((x@df)[, indices_predictor])b       NA
##
## Residual standard error: 1.354 on 1 degrees of freedom
## Multiple R-squared:  0.3534,Adjusted R-squared:  -0.2933
## F-statistic: 0.5465 on 1 and 1 DF,  p-value: 0.5947
```