globalSeq: testing for association between RNA-Seq and high-dimensional data

A Rauschenberger, **MA Jonker**, **MA van de Wiel** and **RX Menezes**

October 17, 2016

This vignette explains how to use the R package **globalSeq**. The function omnibus tests for associations between an overdispersed count variable and a high-dimensional covariate set. The function proprius decomposes the test statistic to show the contributions of individual samples or covariates. And the function cursus performs genome-wide analyses.

1 Initialisation

Start with installing the R package **globalSeq** from Bioconductor:

```
source("https://bioconductor.org/biocLite.R")
BiocInstaller::biocLite("globalSeq")
```

Please type the following command to load and attach the package:

```
library(globalSeq)
```

If you want to reproduce the examples, you should attach the toy database:

```
attach(toydata)
```

The following commands access the R documentation:

```
utils::help(globalSeq)
utils::vignette("globalSeq")
```

2 Test of association

2.1 Data

Data is available for 10 individuals and 16 variables.

```
cbind(y,X)
```

```
##        y X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15
## ind1   7  0  1  0  0  0  0  1  1  1   0   1   0   0   1   0
## ind2   1  1  1  0 17 15 16 13 10 20   0   1   0   1   1   0
## ind3   1  0  1  0  0  0  1  1  1  0   0   0   0   0   0   1
## ind4  15 17 18 16 12 15 20 14 17 18  17  15   0   1   0  13
## ind5   6  1  1  0  0  0  1  1  1  1   1   1   0   1   1   1
## ind6  16 12 14 13 12 10 15 18 17 12  15  12   0   0   0  15
## ind7   5  0  0  1  1  0  0  1  0  0   0   0   1   1   1   1
## ind8   2  0  1  1  0  0  0  1  0  0   0   0   0   1   1   1
## ind9   2  1  1  1  1  0  0  0  1  1   0   0   0   0   1   1
## ind10  5  1  0  1  0  1  0  0  1  1   1   0   0   0   0   0
```

2.2 Minimal example

We are interested whether the response variable $\boldsymbol{y}$ is associated with the covariate matrix $\boldsymbol{X}$. Looking at the data might already lead to an answer.[1]
Because the number of covariates exceeds the sample size, classical tests cannot test their joint significance. But the function **omnibus** also works in high-dimensional settings:

```
set.seed(1)
omnibus(y,X)
```

```
##   pvalue teststat covs
## 1  0.024 651.6189   15
```

[1] Note that the response variable takes higher values for individuals 4 and 6 than for the other individuals. Looking at the covariate matrix, we observe that individuals 2, 4 and 6 are peculiar. We conclude: The data on individuals 4 and 6 speak for an association, but the data on individual 2 speaks against an association. Covariates 12, 13 and 14 are uninformative, and the role of other covariates is less clear.

2.3 Offset

Suppose that an offset is available. Relative to the offset, the response $\boldsymbol{y}$ is more or less constant across samples:

```
rbind(y,offset)

##        ind1 ind2 ind3 ind4 ind5 ind6 ind7 ind8 ind9 ind10
## y         7    1    1   15    6   16    5    2    2     5
## offset   15    1    2   28   10   28    9    2    5     7
```

If we account for this offset, there is no evidence for an association between the response $\boldsymbol{y}$ and the covariate matrix $\boldsymbol{X}$:

```
set.seed(1)
omnibus(y,X,offset=offset)

##   pvalue  teststat covs
## 1  0.977 -2881.224   15
```

2.4 Confounding variable

Suppose that each sample belongs either to group 1 or to group 2. We can observe that $\boldsymbol{y}$ tends to take small values in one group, and large values in the other:

```
rbind(y,group)

##       ind1 ind2 ind3 ind4 ind5 ind6 ind7 ind8 ind9 ind10
## y        7    1    1   15    6   16    5    2    2     5
## group    1    1    1    2    1    2    1    1    1     1
```

We suspect that the group membership explains some variation of the response $\boldsymbol{y}$ or the covariate matrix $\boldsymbol{X}$. Therefore we account for this confounding variable by using stratified permutations:

```
set.seed(1)
omnibus(y,X,group=group)

##      pvalue teststat covs
## 1 0.8947368 651.6189   15
```

2.5 Overdisperion

Setting the dispersion parameter of the negative binomial distribution equal to zero is equivalent to using the Poisson distribution:

```
set.seed(1)
omnibus(y,X,phi=0)

##   pvalue teststat covs
## 1  0.024 21877.03   15
```

2.6 Multiple covariate sets

Suppose that two covariate sets are available:

```
X1 <- X[,c(1:11,15)]
X2 <- X[,12:14]
```

We are interested in testing for associations between $\boldsymbol{y}$ on one hand, and $\boldsymbol{X1}$ or $\boldsymbol{X2}$ on the other:

```
set.seed(1)
omnibus(y,list(X1,X2))

##   joint teststat single.1 single.2 covs.1 covs.2
## 1 0.029 3.448796    0.024    0.284     12      3
```

The output includes the p-value and the test statistic for the joint test, the p-values for the individual tests, and the numbers of tested covariates.

2.7 P-values

When testing single covariate sets, the user can choose between three different types of p-values. By default p-values are calculated by permutation without repetitions (kind = 1). Alternatively, permutation can be interrupted when it becomes impossible to reach a predefined significance level (0 < kind < 1), or the method of control variables can be used (kind = 0).

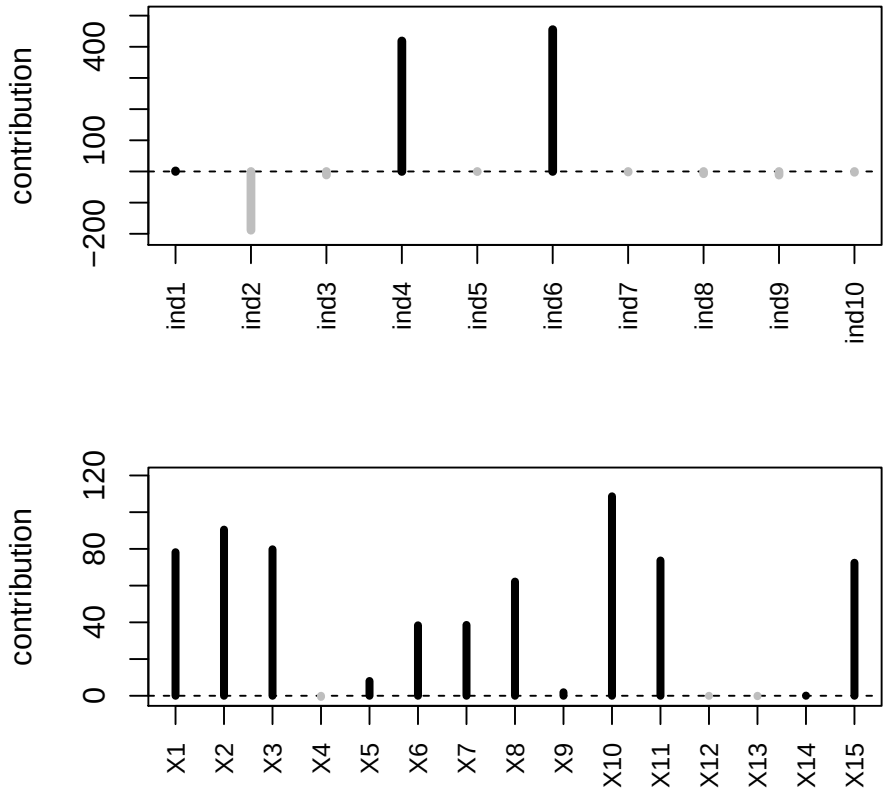
```
omnibus(y,X,kind=1) # crude permutation test
omnibus(y,X,kind=0.05) # interrupting permutation
omnibus(y,X,kind=0) # method of control variables
```

3 Decomposition

3.1 Minimal example

Even though a single hypothesis is tested on the covariate set, the function **proprius** can obtain the contributions of individual covariates or samples to the test statistic.

```
proprius(y,X,type="samples")
proprius(y,X,type="covariates")
```

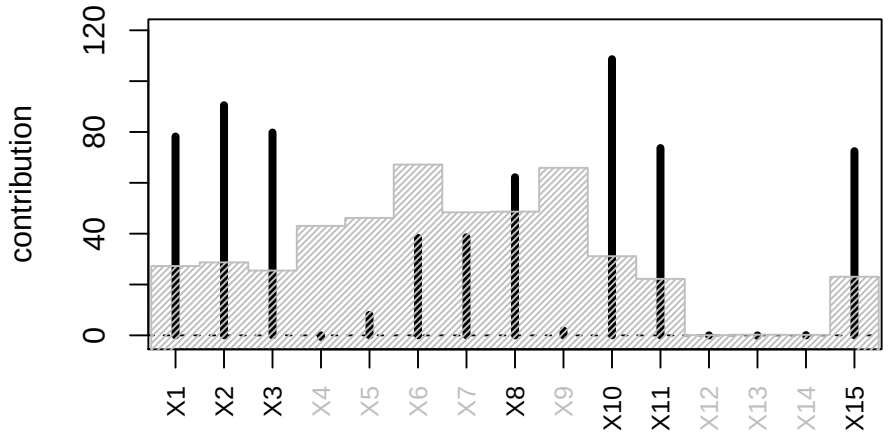


We observe that individual 2 contributes negatively to the test statistic, whereas the contributions of individuals 4 and 6 are positive. We also observe that several covariates have large positive contributions. Summing over the individual contributions gives back the test statistic.

3.2 Null distribution

If a significance level α is specified, then the $1-\alpha$ lower quantile under the null hypothesis is plotted:

```
proprius(y,X,type="covariates",alpha=0.05)
```



3.3 Further arguments

Offsets are included as in section 2.3, confounding variables are taken into account as in section 2.4, and overdispersion is treated as in section 2.5. The decompositions have not been implemented for multiple covariate sets.

4 Genome-wide analysis

4.1 Data

Suppose the matrix $\boldsymbol{Y}$ contains expression levels of 7 genes and 5 individuals, and the matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ represent two different molecular profiles. Suppose that we furthermore know the locations of the genes ($\boldsymbol{Yloc}$), the locations of the genetic or epigenetic alterations ($\boldsymbol{Vloc}$ and $\boldsymbol{Wloc}$), and the chromosome indicators $\boldsymbol{Ychr}$, $\boldsymbol{Vchr}$ and $\boldsymbol{Wchr}$.

```
cbind(Yloc,Ychr,Y)
```

```
##       Yloc Ychr ind1 ind2 ind3 ind4 ind5
## gene1    9    1    9   55    7   32   25
## gene2   32    1   10    6   13   11   13
## gene3   39    1    5   15   30   29   44
## gene4   70    1   11   51    4    9   40
## gene5   75    1   18   20   50   20    5
## gene6   77    1   16    1    4   37   33
## gene7   96    1    1   30   25    6    2
```

```
cbind(Vloc,Vchr,V)
```

```
##     Vloc Vchr ind1 ind2 ind3 ind4 ind5
## V1    27    1  1.1  0.8 -1.4  0.0 -0.9
## V2    38    1  2.8  1.7  0.1 -0.1 -1.5
## V3    39    1 -0.7  0.3 -0.3  1.1  0.3
## V4    47    1  1.3  0.1  0.7 -0.5  0.3
## V5    65    1 -0.2  0.2  0.9  1.3  0.3
## V6    66    1  0.9 -0.5  0.1  0.2  0.9
## V7    68    1  0.1  0.9  0.5 -0.4  2.5
## V8    79    1 -0.6  0.1  0.2  0.2 -1.5
## V9    81    1 -0.6 -1.0  0.6 -0.2 -0.4
## V10   93    1  0.2  0.1 -0.8  1.0 -0.8
```

```
cbind(Wloc,Wchr,W)
```

```
##    Wloc Wchr ind1 ind2 ind3 ind4 ind5
## W1    2    1    0    2    1    0    2
## W2   23    1    2    1    0    1    1
## W3   35    1    1    0    1    1    1
## W4   45    1    1    0    2    1    1
## W5   71    1    1    1    1    0    0
## W6   86    1    0    1    2    0    2
## W7   92    1    0    2    1    1    2
## W8  100    1    2    1    1    1    1
```

4.2 Minimal example

We are interested in testing the expression of each gene for associations with *local* genetic or epigenetic alterations.
Setting the argument window to 5 entails that each gene is tested for association with all covariates that are located no further than 5 units from the gene. In this example the covariates of interest are:

```
## gene1:
## gene2:
## gene3: V2 V3
## gene4: V6 V7
## gene5: V8
## gene6: V8 V9
## gene7: V10
```

The chromosome-wide analysis will not return any p-value for gene 4, because no covariates are in its vicinity.

```
set.seed(1)
cursus(Y,Yloc,V,Vloc,window=5)
```

4.3 Multiple chromosomes

If multiple chromosomes are analysed, it is important to include the chromosome indicators. They make sure that genes are mapped to covariates on the same chromosome:

```
set.seed(1)
cursus(Y,Yloc,V,Vloc,window=5,Ychr,Vchr)
```

4.4 Different library sizes

To account for different sequencing depths, an offset can be included:

```
offset <- colSums(Y) # library sizes
set.seed(1)
cursus(Y,Yloc,V,Vloc,window=5,offset=offset)
```

Otherwise the offset is calculated based on $\boldsymbol{Y}$.

4.5 Multiple molecular profiles

For the simultaneous analysis of multiple molecular profiles, the function **cursus** expects one covariate matrix, one location vector and one window size for each profile.

```
set.seed(1)
cursus(Y,Yloc,list(V,W),list(Vloc,Wloc),list(5,50))

## Analysing a single chromosome.
##           joint   teststat   single.1  single.2 covs.1 covs.2
## gene1        NA        NaN         NA 0.2000000      0      4
## gene2 0.3916667  0.3162150 0.78333333 0.1250000      1      5
## gene3 0.0500000  2.3485357 0.03333333 0.5166667      2      6
## gene4 0.6000000 -0.5383582 0.55000000 0.6750000      3      7
## gene5 0.3666667  0.3000674 0.30000000 0.5250000      1      6
## gene6 0.4333333 -0.1737118 0.74166667 0.3583333      2      6
## gene7 0.7583333 -1.2972658 0.70000000 0.8166667      1      4
```

4.6 Further arguments

Confounding variables are taken into account as in section 2.4. By setting `phi=rep(0,q)` where `q` is the number of genes, the user can restrict the negative binomial distribution to the Poisson distribution. By default, offsets and dispersion parameters are calculated internally, but they can also be inserted. The following example uses the R package **edgeR** from Bioconductor:

```
list <- edgeR::DGEList(counts=Y)
list <- edgeR::calcNormFactors(list)
list <- edgeR::estimateDisp(list)
offset <- list$samples$norm.factors
phi <- list$tagwise.dispersion

cursus(Y,Yloc,V,Vloc,window=5,offset=offset,phi=phi)
```

References

The R package **globalSeq** is based on Rauschenberger et al. [5], where detailed references to previous work are given. If you use **globalSeq** for publications, please cite Rauschenberger et al. [5].

Based on calculations from le Cessie and van Houwelingen [2] and using the generalised linear modelling framework from McCullagh et al. [3], Goeman et al. [1] showed how to test for association between a response variable from the exponential family of distributions and a high-dimensional covariate set, and derived the contributions of samples and covariates to the test statistic. Menezes et al. [4] extended this test to multiple covariate sets. Rauschenberger et al. [5] adapted the methods from Goeman et al. [1] and Menezes et al. [4] to the negative binomial distribution with an unknown dispersion parameter. This permutation test was made computationally efficient using ideas from Wieringen et al. [7] and Senchaudhuri et al. [6].