

SVAPLSseq: An R package to correct for hidden sources of variability in differential gene expression studies based on RNAseq data

Sutirtha Chakraborty^{1*}

¹National Institute of Biomedical Genomics *email: sc4@nibmg.ac.in

Modified: June 25, 2016 Compiled: April 24, 2017

Contents

1 Overview	1
2 Formatting the data for use in the package	2
3 Extracting the signatures of the hidden effects in the data	2
3.1 The Unsupervised SVAPLSseq	3
3.2 The Supervised SVAPLSseq	4
4 Using the estimated hidden effect signatures to detect the truly differentially expressed genes	4

1 Overview

The R package *SVAPLSseq* contains functions that are intended for the extraction and correction of different types of hidden biological and technical variables that could potentially generate latent heterogeneity in RNAseq data on gene expression. The complexity of the sequencing workflow creates a number of technical artefacts along with the inherent biological variability stemming from the unknown gene and sample profiles. The package aims to provide the users with a flexible and generalized framework to identify these hidden effects and adjust for them in order to re-estimate the primary signals of group-specific differential gene expression with higher power and accuracy. The underlying method operates by implementing a non-linear partial least squares regression algorithm on two multivariate random matrices constructed from the data. To that end two methodological variants are provided in this package: (1) Unsupervised SVAPLSseq and (2) Supervised SVAPLSseq. Both these variants yield a set of surrogate variables that are then tested for statistical significance in order to detect the important signatures of latent variability in the data. The package also provides an added functionality

in terms of incorporating these extracted signatures in a linear regression framework and estimating the group- specific differential expression effects. For this purpose two different options are provided: (a) Wald test that uses the R packages “edgeR” and “limma” and (b) Likelihood ratio test.

This document provides a tutorial to use the package for:

- Formatting the data for use in the package.
- Extracting the signatures of the hidden effects in the data.
- Using the estimated hidden effect signatures to detect the truly differentially expressed genes.

2 Formatting the data for use in the package

The starting step for using the package is to set up the RNAseq expression data in an appropriate format. The input data should be in the form of either a count matrix object or a 'SummarizedExperiment' or a 'DGEList' object. The object will contain a feature matrix that will list the features (genes/transcripts) along the rows and samples along the columns. This matrix will contain the read count values for the features corresponding to the different samples. In addition, a separate factor variable should be designed that will keep track of the group each sample belongs (e.g. “treated” and “untreated”, “Normal” and “Cancer”). This variable will enable the estimation of the primary signal for group-specific differential expression of the features.

```
> library(SummarizedExperiment)
> library(SVAPLSseq)
> library(edgeR)
> data(sim.dat)
> dat = SummarizedExperiment(assays = SimpleList(counts = sim.dat))
> dat = DGEList(counts = sim.dat)
> sim.dat[1:6, c(1:3, 11:13)]
```

	S1	S2	S3	S11	S12	S13
[1,]	180	180	183	594	555	585
[2,]	278	275	269	357	359	350
[3,]	182	196	205	439	454	444
[4,]	438	430	345	2818	2790	2888
[5,]	238	226	215	1413	1452	1471
[6,]	156	158	180	486	527	455

3 Extracting the signatures of the hidden effects in the data

The package contains a function `svplsSurr` that extracts the signatures of latent variability (surrogate variables) in the data by using a multivariate non-linear partial least squares (NPLS) algorithm

(Boulesteix and Strimmer 2007). The function takes the original read count matrix of feature expression values along with a factor variable indicating the group of each sample as input. Moreover, it allows the user to specify a certain number of surrogate variables (`max.surrs`) that will be extracted from the data. These variables are further tested for statistical significance to generate an optimal set of significant surrogate variables capturing the latent variation in the data. The function returns a matrix with these variables along the columns and a vector containing the proportions of the total variance in the data space that are explained by them.

The function provides the user with two methodological variants: (1) The Unsupervised SVAPLSseq and (2) The Supervised SVAPLSseq. Details on these two variants and their usage on an RNAseq gene expression data are provided below:

3.1 The Unsupervised SVAPLSseq

This version of the method regresses the primary signal corrected residual matrix on the original gene expression data matrix via NPLS. The estimated scores in the data space are considered as the surrogate variables that are further tested for statistical significance. Setting the `controls` argument of the function to `NULL` starts this version.

```
> data(sim.dat)
> group = as.factor(c(rep(1, 10), rep(-1, 10)))
> sim.dat.se = SummarizedExperiment(assays = SimpleList(counts = sim.dat))
> sim.dat.dg = DGEList(counts = sim.dat)
> sv <- svplsSurr(dat = sim.dat, group = group, max.surrs = 3, controls = NULL)
> sv <- svplsSurr(dat = sim.dat.se, group = group, max.surrs = 3, controls = NULL)
> sv <- svplsSurr(dat = sim.dat.dg, group = group, max.surrs = 3, controls = NULL)
> slotNames(sv)

[1] "surr"      "prop.vars"

> head(surr(sv))

      Comp 1      Comp 2      Comp 3
1 -40.926156 -0.8084875  4.224341
2 -40.055960 -0.4950606  3.898370
3 -42.367566 -1.5554843  5.060546
4 -42.233553 -1.3974685  4.856386
5 -41.106307 -0.9151236  4.329619
6   7.970834 17.1226899 -1.939348

> head(prop.vars(sv))

      Comp 1      Comp 2      Comp 3
0.87480779 0.10710424 0.01808797
```

3.2 The Supervised SVAPLSseq

In this variant a separate expression matrix is first created corresponding to a set of available control genes that do not have any differential effect between the two groups. This matrix is only expected to contain the signatures of the hidden factors in the data. It is then regressed on the original data to extract the surrogate variables for the underlying latent variation. This variant is called by setting the `controls` argument of the function to a vector of indices for the control genes.

```
> data(sim.dat)
> controls = c(1:nrow(sim.dat)) > 400
> group = as.factor(c(rep(1, 10), rep(-1, 10)))
> sim.dat.se = SummarizedExperiment(assays = SimpleList(counts = sim.dat))
> sim.dat.dg = DGEList(counts = sim.dat)
> sv <- svplsSurr(dat = sim.dat, group = group, max.surrs = 3, controls = controls)
> sv <- svplsSurr(dat = sim.dat.se, group = group, max.surrs = 3, controls = controls)
> sv <- svplsSurr(dat = sim.dat.dg, group = group, max.surrs = 3, controls = controls)
> slotNames(sv)

[1] "surr"      "prop.vars"

> head(surr(sv))

      Comp 1      Comp 2      Comp 3
S1 -31.926628 -3.208735 -0.03536976
S2 -31.272906 -2.904747 -0.14619953
S3 -32.891642 -3.941770  0.28247933
S4 -32.853971 -3.871881  0.26042255
S5 -31.989481 -3.260728  0.03723828
S6   5.474165  3.098831  0.45612324

> head(prop.vars(sv))

      Comp 1      Comp 2      Comp 3
0.9800041257 0.0196787967 0.0003170775
```

4 Using the estimated hidden effect signatures to detect the truly differentially expressed genes

The package contains another function `svplsTest` that incorporates the significant surrogate variables estimated by the function `svplsSurr` inside a regression framework in order to test for the genes that are truly differentially expressed between the two groups. The function provides the user with two testing options: (1) Wald test based on the regression coefficients of the primary signal effects (group effects) after incorporating the surrogate variables in a linear model and (2) Likelihood ratio test (LRT) comparing two different regression models: one containing primary signal effects as well as the surrogate variables and the other including only the surrogate variables. A list is returned as

the output that contains the genes detected to be differentially expressed between the two groups (`sig.genes`), the uncorrected pvalues from the test (`pvs.unadj`) and the corresponding FDR adjusted pvalues (`pvs.adj`).

```
> data(sim.dat)
> group = as.factor(c(rep(1, 10), rep(-1, 10)))
> sv = svplsSurr(dat = sim.dat, group = group)
> surr = surr(sv)
> sim.dat.se = SummarizedExperiment(assays = SimpleList(counts = sim.dat))
> sim.dat.dg = DGEList(counts = sim.dat)
> fit = svplsTest(dat = sim.dat, group = group, surr = surr, test = "Wald")
> fit = svplsTest(dat = sim.dat.se, group = group, surr = surr, test = "Wald")
> fit = svplsTest(dat = sim.dat.dg, group = group, surr = surr, test = "Wald")
> head(sig.genes(fit))

[1] "13"  "73"  "210" "211" "246" "267"

> head(pvs.unadj(fit))

      1      2      3      4      5      6
0.30142607 0.47960254 0.56954329 0.57583142 0.04261951 0.62919697

> head(pvs.adj(fit))

      1      2      3      4      5      6
0.8744981 0.9554200 0.9611413 0.9611413 0.3777608 0.9714824
```

References

- [1] Boulesteix, A. L. and Strimmer, K. (2007) Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics* **8**(1), 32–44.