Bioconductor's sva package

Jeffrey Leek and John Storey
Johns Hopkins School of Public Health
Princeton University
email: jleek@jhsph.edu, jstorey@princeton.edu

May 13, 2011

Contents

1 Overview

The sva package contains functions for the identifying and building surrogate variables for high-dimensional data sets. Surrogate variables are covariates constructed directly from high-dimensional data (e.g., gene expression data) that can be used in subsequent analyses (e.g., differential expression analysis). Surrogate variables are designed to overcome the ubiquitous problem of dependence in multiple testing. The use of surrogate variables in differential expression analysis has been shown to reduce dependence, stabilize error rate estimates, and improve reproducibility see [2, 3] for more detailed information.
This document provides a tutorial for using the sva package. The key functions in the sva package are: sva for identifying and building surrogate variables from a data set and a model matrix, num.sv for estimating the number of surrogate variables from the data matrix and a model matrix (based on a modification of the [1] algorithm), and the low level computational functions twostepsva.build and irwsva.build for computing the surrogate variables. As with any R package, detailed information on functions, their arguments and value, can be obtained in the help files. For instance, to view the help file for the function sva within R, type ? sva. Here we will demonstrate the use of sva to analyze a simulated expression experiment.

2 Simulated Example

We demonstrate the functionality of this package using simulated gene expression data that clearly illustrates the characteristics of the SVA approach. The data used in this analysis is included with the sva package as the dataset svadat. This data set consists of simulated data for 1000 genes (in rows) and 20 arrays (in columns). The first 10 arrays correspond to the first group and the second 10 correspond to the second. A second factor also affects the gene expression data and is

an indicator variable described below. Genes 1-300 are differentially expressed across groups and genes 200-500 are differentially expressed with respect to the second factor. The goal is to identify genes differentially expressed across groups. We show how surrogate variables are estimated and how this affects a differential expression analysis.
To load the data set type `data(svadat)`, and to view a description of this data type `? svadat`.

```
> library(sva)
> data(svadat)
> dim(svadat)

[1] 1000   20
```

3 The sva function

The `sva` function computes surrogate variables from the gene expression matrix and model matrix for a microarray experiment [3]. In the expression matrix, genes should be in rows and arrays in columns. It is assumed that the number of genes is much larger than the number of arrays. First we test for differential expression with respect to group, assign the p-values for each gene to the vector `pu` and plot a histogram of the p-values (see Figure 1).

```
> grp <- rep(c(0, 1), each = 10)
> mod <- cbind(rep(1, 20), grp)
> mod0 <- cbind(rep(1, 20))
> hfact <- rep(c(0, 1), 10)
> pu <- f.pvalue(svadat, mod, mod0)

> hist(pu, main = "", xlab = "P-value", col = "grey")
```

Then we calculate the surrogate variables using the `sva` and assign the result to the surrogate variable object `svaboj`. We can compare the estimated surrogate variable to the second factor and see if they match.

```
> svaobj <- sva(svadat, mod, mod0, method = "irw")

Number of significant surrogate variables is:  1
Iteration (out of 5 ):1  2  3  4  5

> cor(hfact, svaobj$sv[, 1])

[1] -0.9991339
```

We can also calculate the p-values adjusted for surrogate variables by including the surrogate variables as covariates in the linear model fits. A comparison of the null p-values from each analysis can be performed by examining the histograms (Figure 2), or with the Kolmogorov-Smirnov test of equality with the uniform distribution.

```
> mod.sv <- cbind(mod, svaobj$sv)
> mod0.sv <- cbind(mod0, svaobj$sv)
> pa <- f.pvalue(svadat, mod.sv, mod0.sv)
```

```
> par(mfrow = c(1, 2))
> hist(pu[301:1000], main = "Unajusted Null P-values", xlab = "P-value",
+     col = "grey")
> hist(pa[301:1000], main = "Adjusted Null P-values", xlab = "P-value",
+     col = "grey")

> ks.test(pu[301:1000], "punif")$p.value

[1] 9.552619e-06

> ks.test(pa[301:1000], "punif")$p.value

[1] 0.7095938
```

References

[1] Buja A. and Eyuboglu N. Remarks on parrallel analysis. *Multivariate Behavioral Research, 27(4), 509-540*, 1992.

[2] J.T. Leek and J.D. Storey. Capturing heterogeneity in gene expression studies by 'surrogate variable analysis'. *PLoS Genetics 3:e161*, 2007.

[3] J.T. Leek and J.D. Storey. A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences 105:18718-18723*, 2008.

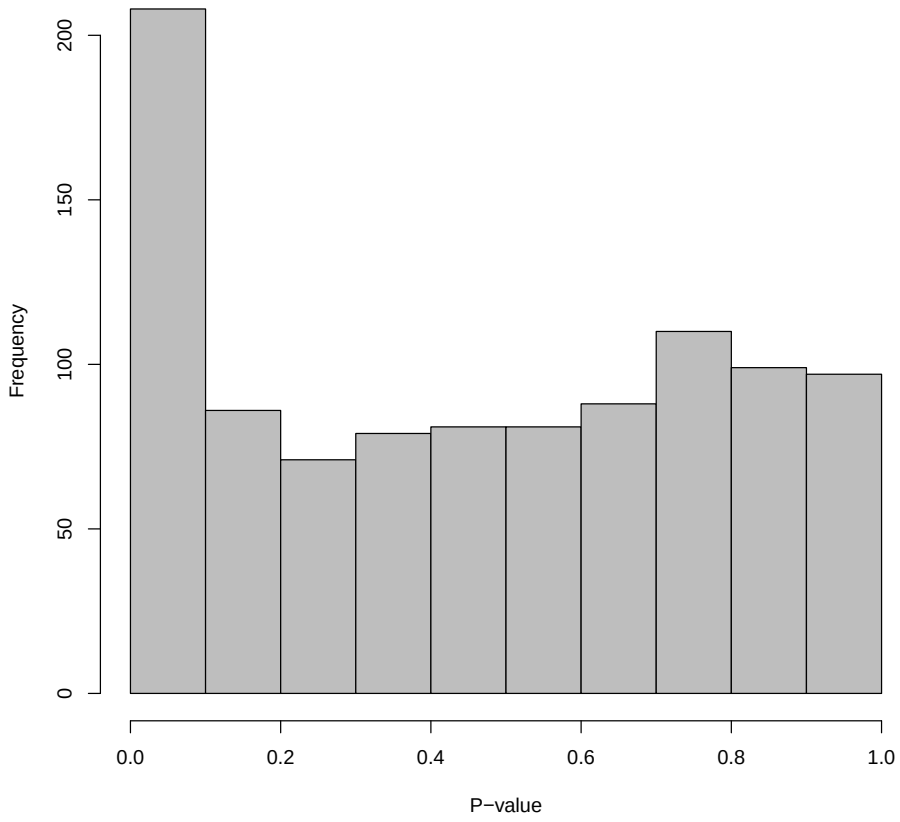

Figure 1: Histogram of unadjusted p-values.

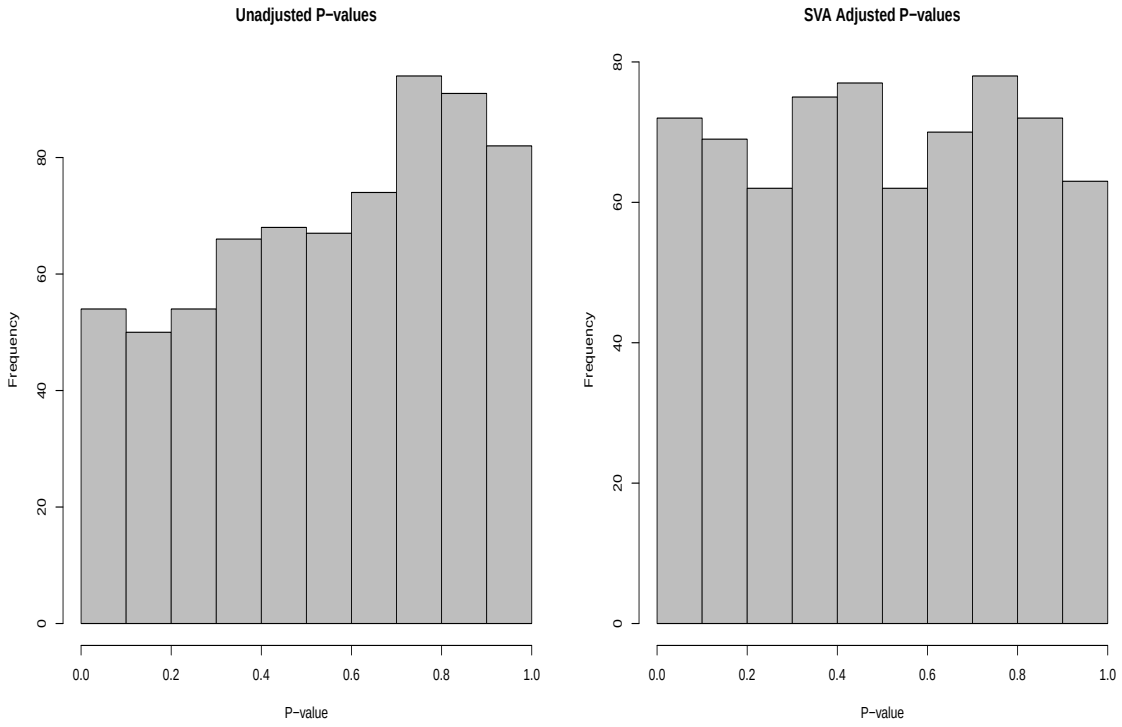

Figure 2: Histograms of the unadjusted and the SVA adjusted null p-values from differential expression analysis for group.