covEB package

1 Introduction

In Bioinformatics, one commonly used tool in differential expression analysis has been an empirical Bayes approach to estimating variances. This method was shown to reduce the false positive rates, particularly at low sample sizes. Small sample sizes are by monetary necessity, common place in experiments. This method, combines information across genes, shrinking the gene-specific variances to a common variance across all genes. Because this approach is used in linear regressions, we hypothesised that a similar methodology could be used with correlations as these are linear regressions between two variables. Correlation matrices are important in inferring relationships and networks between regulatory or signalling elements. As the sample sizes for experiments are small, these correlations can be difficult to estimate and can exhibit high false positive rates. This package is designed to reduce these false positive rates and therefore direct the researcher to higher value relationships that are more likely to be validated experimentally. At a genome-wide scale estimation of correlation matrices can also be computationally demanding. This package provides an empirical Bayes approach to improve covariance estimates for gene expression, where we assume the covariance matrix has block diagonal form. These covariance matrices can be estimates from either microarray or RNA-seq data.

1.1 A Simple Example

We show a simple example of how to run the empirical Bayes estimation, these are trivial examples but serve to illustrate the syntax and parameters of the function. We use the package mvtnorm to simulate data from a multivariate normal distribution.

```
> library(covEB)
>          sigma <- matrix(c(4,2,2,3), ncol=2)
>         x <- rmvnorm(n=500, mean=c(1,2), sigma=sigma)
>          samplecov<-cov(x)
>          test<-EBsingle(samplecov,startlambda=0.4,n=500)
```

In this example we pass the sample covariance matrix (samplecov) to the function EBsingle. In addition we give the number of samples used to calculate the covariance matrix, in this case 500. The third parameter startlambda is a thresholding parameter that is used to determine the block diagonal structure of the matrix. Once the block diagonal structure is known, the average of the correlations within each block is used to create the block diagonal prior that has a flat correlation structure within each block. An alternative approach is shown below in test2 where the groupings of the variables is assumed to be known, this information is then passed to the function as a list of elements in each block.

```
>          sigma <- matrix(c(4,2,0.5,0.5,2,3,0.5,0.5,0.5,0.5,3,2.5,0.5,0.5,2.5,4), ncol=4)
>         x <- rmvnorm(n=500, mean=c(1,2,1.5,2.5), sigma=sigma)
>          samplecov<-cov(x)
>         vnames<-paste("a",1:4,sep="")
>         rownames(samplecov)<-vnames
>         colnames(samplecov)<-vnames
>         test2<-EBsingle(samplecov,groups=list(c("a1","a2"),c("a3","a4")),n=500)
```

1.2 Example with biological data

Here we use a data set available from bioconductor to demonstrate how the covEB package can be used in the pipeline analysis of gene expression data. We load the data package curatedBladderData that contains gene expression from bladder cancer patients in the R object type expression set. We get the gene expression data matrix, this contains around 5,000 probes from microarrays with 40 samples and store this in the matrix Edata.

```
> library(curatedBladderData)
> data(package="curatedBladderData")
> data(GSE89_eset)
> Edata<-exprs(GSE89_eset)
```

We filter the data to include those that are 'expressed' as defined as being in the top 20th percentile according to variance across samples. This gives us just over a thousand genes, we then calculate the covariance matrix between the genes. This covariance matrix is our input into the covEB function.

```
> variances<-apply(Edata,1,var)
> edata<-Edata[which(variances>quantile(variances,0.8)),]
> covmat<-cov(t(edata))
> cormat<-cov2cor(covmat)
> #we are now able to use covmat as input into covEB:
> out<-EBsingle(covmat,startlambda=0.5,n=40)
```

We now provide an example of how the output may be used. We can visualise the correlations between genes using functions available in R and its associated packages. First we use simple thresholding to define significant correlations, we create adjacency matrices for graphs, setting correlations below 0.5 to zero.

```
> outmat<-out
> outmat[abs(out)<0.5]<-0
> outmat[abs(out)>=0.5]<-1
```

We find connected subgraphs in the adjacency matrix using the clusters function and then select one of the subgraphs (number 6) that has 12 genes in it. Finally, for visualisation purposes, we remove edges between nodes and themselves (i.e. the diagonal)

```
> clusth<-clusters(graph.adjacency(outmat))
> sel<-which(clusth$membership==6)
> subgraphEB<-outmat[sel,sel]
> subgraph<-cormat[sel,sel]
> subgraph[subgraph<0.5]<-0
> subgraph[subgraph>=0.5]<-1
> diag(subgraph)<-0
> diag(subgraphEB)<-0
```

We can now plot these graphs, there are 6 fewer edges after using covEB. On a larger network this would help the interpretability of the model further.

```
> plot(graph.adjacency(subgraph,mode="undirected"))
> plot(graph.adjacency(subgraphEB,mode="undirected"))
```

2 References

Champion, C. J. (2003). Empirical Bayesian estimation of normal variances and covariances. Journal of Multivariate Analysis, 87(1), 60-79. http://doi.org/10.1016/S0047-259X(02)00076-3

Benjamin Frederick Ganzfried, Markus Riester, Benjamin Haibe-Kains, Thomas Risch, Svitlana Tyekucheva, Ina Jazic, Xin Victoria Wang, Mahnaz Ahmadifar, Michael Birrer, Giovanni Parmigiani, Curtis Huttenhower, Levi Waldron. curatedOvarianData: Clinically Annotated Data for the Ovarian Cancer Transcriptome, Database 2013: bat013 doi:10.1093/database/bat013 published online April 2, 2013.