Examples of using gimmR package (Gaussian Infinite Mixture Model in R)

Mario Medvedovic, Vinayak Kumar

August 30, 2010

1 Introduction

gimmR serves as the interface for running low level procedures for clustering gene expression data using the Bayesian infinite mixtures clustering procedures described in Medvedovic and Sivaganesan (2002), Medvedovic *et al.* (2004), Medvedovic and Guo (2004), Liu *et al.* (2006), and Freudenberg *et al.* (2010). For more details see **gimmRUserManual.pdf**.

2 Using the software

First, load the library with

```
library(gimmR)
```

```
<environment: R_GlobalEnv>
```

We will anlayze the Galactose dataset descried previously in Yeung *et al.* (2004) and Medvedovic *et al.* (2004). The dataste consists of gene expression measurements for 205 genes accross 20 experimental conditions with 4 replicated observations for each condition. To load "GalData" and print first 8 columns and rows (data for first two genes):

```
data(GalData)
head(GalData[,1:8], 8)
```

```
      ORF GeneName  wtRG1 gal1RG1 gal2RG1 gal3RG1 gal4RG1 gal5RG1
1 YAL038W    CDC19 -0.118  -0.729  -0.136  -0.202  -0.036   0.248
2 YAL038W    CDC19 -0.074  -0.723  -0.177  -0.214  -0.041   0.188
3 YAL038W    CDC19  0.082  -0.639   0.176  -0.134   0.406  -0.064
4 YAL038W    CDC19  0.053  -0.588   0.408  -0.154   0.428  -0.010
```

```
5 YBL021C     HAP3  0.072   0.157  -0.365   0.152  -0.479  -0.054
6 YBL021C     HAP3  0.013   0.165  -0.651   0.049  -0.208  -0.071
7 YBL021C     HAP3  0.000  -0.009   0.045  -0.019  -0.204  -0.171
8 YBL021C     HAP3 -0.022  -0.009   0.051   0.043  -0.227  -0.010
```

For the generic structure of the dataframe to be used as input for ***gimmR*** procedures see **gimmRUserManual.pdf**. We will first cluster all 820 gene expression profiles without taking into account the replication structure. The following command will take the "**GalData**" dataset, perform 100 Gibbs sampler iterations (specified by nIter=100), discard the first 50 as "burn in" (burnIn=50), calculate posterior pairwise probabilities of co-expression, and contsruct the hierarchical clustering structure using the average linkage principle (default). The hierarchical clustering structure and the information needed to display and anlyze clustering results are storied in the ***gimmR*** object *galGimm*.

```
galGimm<-runGimmNPosthoc(GalData, M=20, T=820, nIter=100, burnIn=50,
              nreplicates=1, verbose=FALSE, intFiles=FALSE)
```

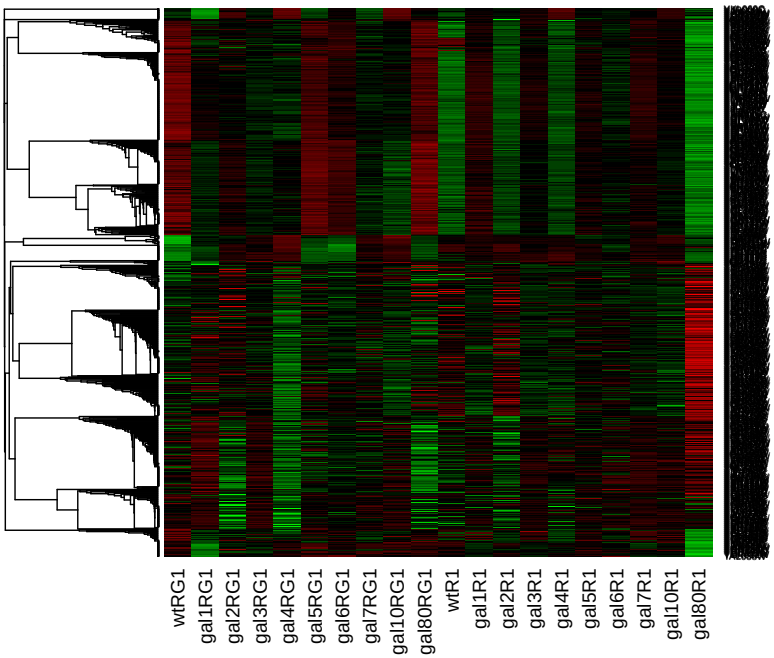
```
Running GIMM Executable ..........
Running posthoc Executable .....
```

The definitions of the parameters used here are:

- **M**: - Number of samples, i.e. dimensionality of the gene expression vectors to be clustered.
- **T**: - Number of genes to be clustered.
- **nIter**: - Number of iterations to be generated by the Gibbs sampler.
- **nreplicates**: - Number of experimental replicates within a single sample.
- **burnIn**: - Number of Gibbs sampler iterations to be discarded as "burn-in".
- **verbose**: - If TRUE, all the internal comments of the executable will be displayed on the console.
- **intFiles**: - If TRUE, the internal files generated by executables gimm and posthoc will not be deleted.

The heatmap displying the clustering can be produced as

```
drawHeatmap(galGimm, color = "red-green")
```



Please note that the default stack size may be too low while using *gimmR* just like with most other hierarchical clustering procedures. To to remedy this, R should be started with a command like the following:

```
R --max-ppsize=50000
```

The Gibbs sampling can be time-consuming. In order to monitor progress, you can set the variable verbose to TRUE, which will print the current iteration counter and the current number of clusters:

```
galGimm<-runGimmNPosthoc(GalData, M=20, T=820, nIter=100, burnIn=50,
           nreplicates=1, verbose=TRUE, intFiles=FALSE}
```

In addition to creating the list of data frames that can be used to generate heatmaps within R, as a side-effect, runGimmNPosthocesults produces **Results.cdt** and **Results.gtr** files in the current working directory. The hierarchical clustering defined in these two files can be viewed and analyzed using **treeview** software (Eisen *et al.* (1998)). The java version of **treeview** for Linux can be downloaded from http://sourceforge.net/project/showfiles.php?group_id=84593. The Windows version can be downloaded from Michael Eisen's web page

http://rana.lbl.gov/EisenSoftware.htm. An extended version facilitating the functional annotation of clustering results is available at http://eh3.uc.edu/clean/ and http://eh3.uc.edu/ftreeview/.

The run-time of the Gibbs sampler depends on the size of the data and the number of iterations specified, but also on the number of mixtures underlying the data. Therefore, the Gibbs sampler can be time-consuming! For this reason, it is recommended to run the sampler in the verbose mode in order to monitor the progress of the sampling procedure.

3 Data format for the simple model

The required format for the input data set can be inferred from the structure of the *GalData* data frame. In the simple model each row in the data frame represents the gene expression profile for a single gene. The first two columns are assumed to contain gene annotations and the remaining columns contain expression levels of genes under different experimental conditions. In the *GalData* datasets, each gene is represented by 4 rows of data obtained from 4 independent microarray hybridizations, but in this simple analysis the replicate measurements are treated as different expression profiles. One simple assessment of the results can be in terms of the proportion of experimental replicates for the same gene that cluster together. For a thorough description of the dataset see Yeung *et al.* (2003). For the proper treatment of this dataset and experimental replicates it contains using the replicated data IMM model see the next section.

4 Analyzing data with replicates

The following code will cluster the *GalData* dataset using the replicated data IMM model as described in Medvedovic *et al.* (2004). The only difference from the code in Section 2 is that the *nReplicates* parameter is set to 4 and the T parameter specifying the number of profiles is set to 205 in the call to the *runGimmNPosthoc* function.

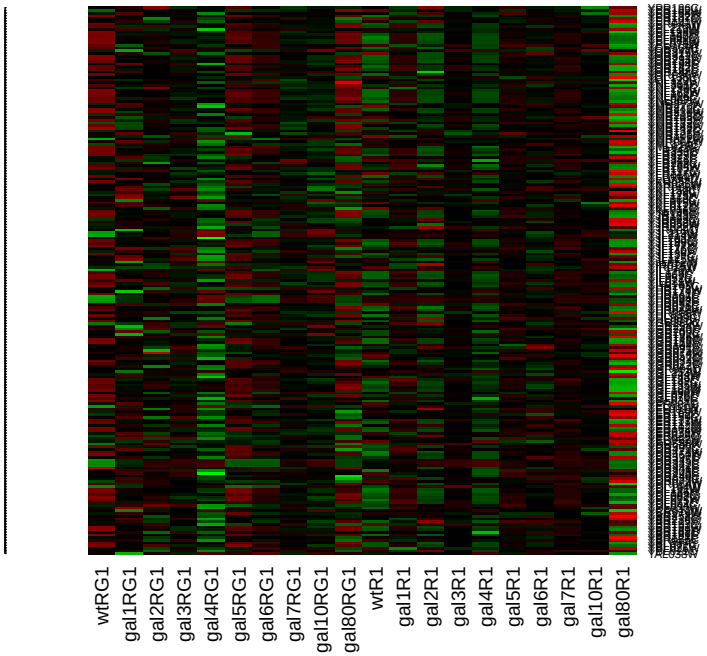
```
data(GalData)

galGimm <- runGimmNPosthoc(GalData, M=20, T=205, nIter=100, burnIn=50,
                verbose=FALSE, nreplicates=4, intFiles=FALSE)

Running GIMM Executable ..........
Running posthoc Executable .....
```

```
drawHeatmap(galGimm, color = "red-green")
```

The following figure shows the heatmap for the gimm object.



5 Using the Context-Specific Infinite Mixture Model (CSIMM)

In the context-specific model, the experimental conditions are further organized into contexts (http://eh3.uc.edu/gimm/csimm). The following code will re-create the analysis of the sporulation and cell cycle data described in Liu *et al.* (2006). **It should be noted that this code takes about three hours to run on the OpenMP-enabled system with dual 3.6GHz Xeons. On a single CPU Windows machine it can take 24 hours to complete. To run just a few test-iterations of the algorithm reduce the nIter and burnIn numbers.**

Successfully running this code may require starting R with

```
R --max-ppsize=50000
```

and setting the *expressions* option with

```
options(expressions=100000)
```

As it can be seen from the code, in order to run the context-specific clustering, the *contextSpecific* parameter is set to "y", *nContexts* is set to 4, and *contextLengths* is set to the vector c(8,7,9,7) defining number experiments in each of the 4 contexts. Note that the sum of *contextLengths* must be equal to the total number of experiments M. Furthermore, the context-specific model currently does not support replicated observations.

6 Using Differential Co-expression Infinite Mixtures (DCIM)

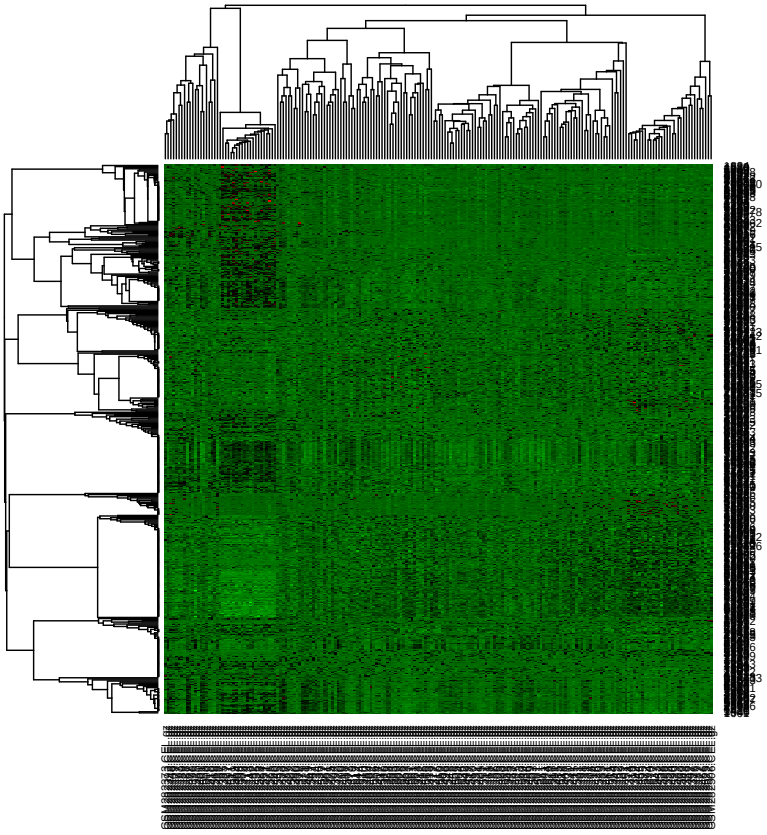
Like in the context-specific model, the experimental conditions are further organized into contexts but DCIM is designed to discern these contexts from the data rather than require their a priori specification (http://eh3.uc.edu/gimm/dcim/). The following code will run the DCIM analysis for a subset of the Schmidt *et al.* (2008) dataset. The subset is comprised of the top 500 DCS gene signature as described in Freudenberg *et al.* (2010). To invoke the DCIS algorithm set the *estimate_contexts* parameter to "y". By default, each sample is assigned to its own 'basic' context, and basic contexts are then grouped into 'meta'-contexts. It is possible to specify basic contexts by using the *nContexts* and *contextLengths* parameters as in the CSIMM setting. It should be noted that this code may take an extended period of time to complete. To run just a few test-iterations of the algorithm reduce the nIter and burnIn numbers.

```
data(DCE500)
dceGimm <- runGimmNPosthoc(DCE500, M=200, T=500, nIter=100, burnIn=50,
            estimate_contexts="y", verbose=TRUE, intFiles=TRUE)

Running GIMM Executable ..........
Running posthoc Executable .....

drawHeatmap(dceGimm, color="red-green")
```

The following figure shows the heatmap for the gimm object.



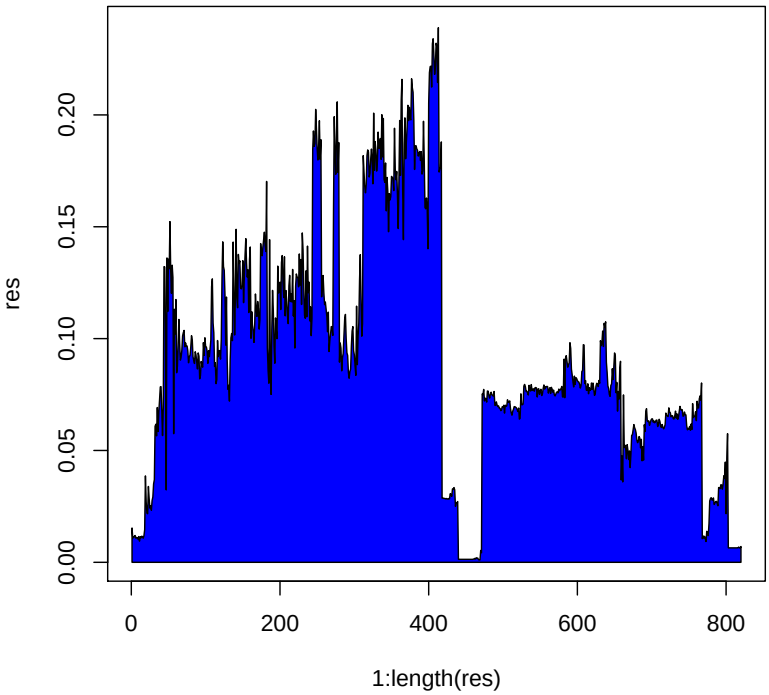
Given two contexts (usually estimated from the data), the following code will determine the gene-specific differential co-expression (DCE) score for these two contexts and plot their distribution.

```
data(GalData)
gimmOut <- runGimmNPosthoc(GalData, dataFile="galData", M=20,
         nIter=1000, burnIn=500, T=820, estimate_contexts="y",
 intFiles=TRUE, verbose=TRUE)
res <- computeDCEscore(which(cutree(gimmOut$hSClustData,
         k=2)==1), which(cutree(gimmOut$hSClustData, k=2)==2),
         paramsList=list(dataFile="galData", M=20, T=820,
         burnIn=500))
res <- res[gimmOut$hGClustData$order]
plot(1:length(res), res, type="n")
polygon(c(1:length(res), length(res), 1), c(res, 0, 0), col=4)

Running GIMM Executable ..........
Running posthoc Executable .....

Reading .out file ... done.
Reading .cc file ... done.
```

```
Reading .wc file ... done.
Converting contexts to samples ...
   iteration  500  ...
   iteration  600  ...
   iteration  700  ...
   iteration  800  ...
   iteration  900  ...
   iteration  1000  ... done.
Converting global clusters to local clusters ...
   iteration  500  ...
   iteration  600  ...
   iteration  700  ...
   iteration  800  ...
   iteration  900  ...
   iteration  1000  ... done.
Running posthoc Executable .....
Reading .out file ... done.
Reading .wcs file ... done.
Converting global clusters to local clusters ...
   iteration  500  ...
   iteration  600  ...
   iteration  700  ...
   iteration  800  ...
   iteration  900  ...
   iteration  1000  ... done.
Running posthoc Executable .....
```



Reference List

1. Medvedovic M and Sivaganesan S, Bayesian infinite mixture model based clustering of gene expression profiles. Bioinformatics 18: 1194-1206, 2002.

2. Medvedovic M, Yeung KY, and Bumgarner RE, Bayesian mixture model based clustering of replicated microarray data. Bioinformatics. 20: 1222-1232, 2004.

3. Medvedovic M and Guo J, Bayesian Model-Averaging in Unsupervised Learing From Microarray Data. BIOKDD 2004, 2004.

4. Liu X, Sivaganesan S, Yeung K.Y., Bumgarner RE, and Medvedovic M, Bayesian context-specific infinite mixture model for clustering of gene expression profiles accross diverse microarray datasets. Bioinformatics 22:1737-44. 2006.

5. Freudenberg JM, Sivaganesan S, Wagner M, Medvedovic M. A semi-parametric Bayesian model for unsupervised differential co-expression analysis. Accepted for Publication in BMC Bioinformatics.

6. Yeung KY, Medvedovic M, and Bumgarner RE, Clustering Gene Expression Data with Repeated Measurements. Genome Biology 4: R34, 2003.

7. Eisen MB, Spellman PT, Brown PO, and Botstein D, Cluster analysis and display of genome-wide expression patterns. Proc.Natl.Acad.Sci.U.S.A 95: 14863-14868, 1998.

8. Schmidt M, Böhm D, von Törne C, Steiner E, Puhl A, Pilch H, Lehr HA, Hengstler JG, Kölbl H, and Gehrmann M. The humoral immune system has a key prognostic impact in node-negative breast cancer. Cancer Res. 2008 Jul 1;68(13):5405-13.