Clipper package (Version 1.12.0)

Paolo Martini, Gabriele Sales and Chiara Romualdi

May 3, 2016

1 Along signal paths: an empirical gene set approach exploiting pathway topology

1.1 clipper approach

Different experimental conditions are usually compared in terms of their gene expression mean differences. In the univariate case, if a gene set changes significantly its multivariate mean expression in one condition with respect to the other, it is said to be differentially expressed. However, the difference in mean expression levels does not necessarily result in a change of the interaction strength among genes. In this case, we will have pathways with significant altered mean expression levels but unaltered biological interactions. On the contrary, if transcripts abundances ratios are altered, we expect a significant alteration not only of their mean, but also of the strength of their connections, resulting in pathways with completely corrupted functionality. Therefore, to look for pathways strongly involved in a biological process, we should look at pathways with both mean and variance significantly altered. *clipper* is based on a two-step approach: 1) it selects pathways with covariance matrices and/or means significantly different between experimental conditions and 2) on such pathways, it identifies the sub-paths mostly associated to the phenotype. This is a very peculiar feature in pathway analysis. To our knowledge this is the first approach able to systematically inspect a pathway deep in its different portions.

1.2 Other approaches on Bioconductor

Currently there are some pathway analysis methods implemented in Bioconductor (probably the most famous is GSEA), but very few of them try to exploit pathway topology. Example of the latter category are SPIA and DEGraph.

In *Martini et al. 2012* is provided a detailed comparison of the performance of non-topological analysis (*GSEA*) and topological analysis (*SPIA* and *clipper*) using both real and simulated data. In the next few words, we are going to highlight the defferences of these apperoaches.

GSEA uses pathway as a list of genes without taking into account the structure of the pathway while *SPIA* takes into account pathway topological information, gene fold-changes and pathway enrichment scores. Then SPIA takes as input only the list of differentially expressed genes. So, from a practical point of view clipper and SPIA test different null hypotheses.

More importantly, *clipper* is able to highlight the portions (sub-path) of the pathway that are mostly involved in the phenotype under study using graph decomposition theory.

For more ditails please refer to *Martini et al. 2012*.

2 Performing pathway analysis

In this section, we describe how to perform the topological pathway analysis on a whole pathway. As an example we used the gene expression data published by Chiaretti et al. on acute lymphocytic leukemia (ALL) cells associated with known genotypic abnormalities in adult patients. Several distinct genetic mechanisms lead to ALL malignant transformations deriving from distinct lymphoid precursor cells that have been committed to either T-lineage or B-lineage differentiation. Chromosome translocations and molecular rearrangements are common events in B-lineage ALL and reflect distinct mechanisms of transformation. The BCR breakpoint cluster region and the c-abl oncogene 1 (BCR/ABL) gene rearrangement occurs in about 25% of cases in adult ALL.

The expression values (available through Bioconductor) deriving from Affymetrix single channel technology, consist of 37 observations from one experimental condition (class 2, n1 37, BCR; presence of BCR/ ABL gene rearrangement) and 42 observations from another experimental condition (class 1, n2 41, NEG; absence of rearrangement). The *clipper* method is based on Gaussian graphical models, therefore it is strongly recommended to use log-transformed data.

In this example, we are going to evidence the differences between BCR/ABL (class 2) and NEG (class 1) though a topological pathway analysis.

We use the *graphite* bioconductor R package as a source of pathway topological information. In our test dataset, given the presence of the BCR/ABL chimera, we expect that all the pathways including BCR and/or ABL1 will be impacted. Here we retrieve, for example, the KEGG "Chronic myeloid leukemia" pathway.

```
> library(graphite)
> kegg  <- pathways("hsapiens", "kegg")
> graph <- convertIdentifiers(kegg[["Chronic myeloid leukemia"]], "entrez")
> graph <- pathwayGraph(graph)
> genes <- nodes(graph)
> head(genes)
```

```
[1] "10000" "1019"  "1021"  "1026"  "1029"  "1147"
```

Once the pathway (converted to a graphNEL object) is loaded in the workspace, we need to retrieve the expression matrix and the corresponding sample annotations (2 denoting samples with translocation and 1 denoting samples with no BCR/ABL translocation). We use as an example the ALL Bioconductor package.

```
> library(ALL)
> data(ALL)
```

Fist of all, we should take a look at the phenoData.

```
> head(pData(ALL))
```

```
       cod diagnosis sex age BT remission CR   date.cr
01005 1005 5/21/1997   M  53 B2        CR CR  8/6/1997
01010 1010 3/29/2000   M  19 B2        CR CR 6/27/2000
03002 3002 6/24/1998   F  52 B4        CR CR 8/17/1998
04006 4006 7/17/1997   M  38 B1        CR CR  9/8/1997
04007 4007 7/22/1997   M  57 B2        CR CR 9/17/1997
04008 4008 7/30/1997   M  17 B1        CR CR 9/27/1997
      t(4;11) t(9;22) cyto.normal        citog mol.biol
01005   FALSE    TRUE       FALSE      t(9;22)  BCR/ABL
01010   FALSE   FALSE       FALSE  simple alt.      NEG
03002      NA      NA          NA         <NA>  BCR/ABL
04006    TRUE   FALSE       FALSE      t(4;11) ALL1/AF4
04007   FALSE   FALSE       FALSE      del(6q)      NEG
04008   FALSE   FALSE       FALSE complex alt.      NEG
      fusion protein mdr   kinet   ccr relapse transplant
01005           p210 NEG dyploid FALSE   FALSE       TRUE
01010           <NA> POS dyploid FALSE    TRUE      FALSE
```

```
03002           p190 NEG dyploid FALSE    TRUE      FALSE
04006           <NA> NEG dyploid FALSE    TRUE      FALSE
04007           <NA> NEG dyploid FALSE    TRUE      FALSE
04008           <NA> NEG hyperd. FALSE    TRUE      FALSE
                   f.u date last seen
01005 BMT / DEATH IN CR           <NA>
01010               REL      8/28/2000
03002               REL     10/15/1999
04006               REL      1/23/1998
04007               REL      11/4/1997
04008               REL     12/15/1997
```

```
> dim(pData(ALL))
```

```
[1] 128  21
```

This data.frame summarized all the phenotypic features of the samples. In our analysis, we are interested in B-cell. This information is hosted in the column called 'BT'.

```
> pData(ALL)$BT
```

```
  [1] B2 B2 B4 B1 B2 B1 B1 B1 B2 B2 B3 B3 B3 B2 B3 B  B2 B3
 [19] B2 B3 B2 B2 B2 B1 B1 B2 B1 B2 B1 B2 B  B  B2 B2 B2 B1
 [37] B2 B2 B2 B2 B2 B4 B4 B2 B2 B2 B4 B2 B1 B2 B2 B3 B4 B3
 [55] B3 B3 B4 B3 B3 B1 B1 B1 B1 B3 B3 B3 B3 B3 B3 B3 B3 B1
 [73] B3 B1 B4 B2 B2 B1 B3 B4 B4 B2 B2 B3 B4 B4 B4 B1 B2 B2
 [91] B2 B1 B2 B  B  T  T3 T2 T2 T3 T2 T  T4 T2 T3 T3 T  T2
[109] T3 T2 T2 T2 T1 T4 T  T2 T3 T2 T2 T2 T2 T3 T3 T3 T2 T3
[127] T2 T
Levels: B B1 B2 B3 B4 T T1 T2 T3 T4
```

```
> pAllB <- pData(ALL)[grep("B", pData(ALL)$BT),]
> dim(pAllB)
```

```
[1] 95 21
```

After this selection, we are interest in the isolation of sample with translocation from those without translocation. This information is hosted in the column 'mol.biol'.

```
> pAllB$'mol.biol'
```

```
 [1] BCR/ABL  NEG      BCR/ABL  ALL1/AF4 NEG      NEG
 [7] NEG      NEG      NEG      BCR/ABL  BCR/ABL  NEG
[13] E2A/PBX1 NEG      BCR/ABL  NEG      BCR/ABL  BCR/ABL
[19] BCR/ABL  BCR/ABL  NEG      BCR/ABL  BCR/ABL  NEG
[25] ALL1/AF4 BCR/ABL  ALL1/AF4 NEG      ALL1/AF4 BCR/ABL
[31] NEG      BCR/ABL  NEG      BCR/ABL  BCR/ABL  ALL1/AF4
[37] NEG      BCR/ABL  BCR/ABL  BCR/ABL  NEG      E2A/PBX1
[43] BCR/ABL  NEG      NEG      NEG      BCR/ABL  p15/p16
[49] ALL1/AF4 BCR/ABL  BCR/ABL  NEG      E2A/PBX1 NEG
[55] NEG      NEG      BCR/ABL  BCR/ABL  NEG      NEG
[61] ALL1/AF4 NEG      ALL1/AF4 NEG      BCR/ABL  NEG
[67] NEG      NEG      NEG      NEG      BCR/ABL  ALL1/AF4
[73] BCR/ABL  NEG      E2A/PBX1 NEG      BCR/ABL  BCR/ABL
[79] NEG      NEG      NEG      NEG      BCR/ABL  NEG
[85] BCR/ABL  BCR/ABL  BCR/ABL  ALL1/AF4 NEG      NEG
[91] BCR/ABL  NEG      BCR/ABL  BCR/ABL  E2A/PBX1
Levels: ALL1/AF4 BCR/ABL E2A/PBX1 NEG NUP-98 p15/p16
```

```
> NEG <- pAllB$'mol.biol' == "NEG"
> BCR <- pAllB$'mol.biol' == "BCR/ABL"
> pAll <- pAllB[(NEG | BCR),]
```

Now we have to build the vector of classes.

```
> classesUn <- as.character(pAll$'mol.biol')
> classesUn[classesUn=="BCR/ABL"] <- 2
> classesUn[classesUn=="NEG"] <- 1
> classesUn <- as.numeric(classesUn)
> names(classesUn) <- row.names(pAll)
> classes <- sort(classesUn)
```

Now that we have the vector of classes, we can isolate the subset of sample from the original expression set and subsequently we convert affymetrix probe names into entrez gene ids.

```
> library("hgu95av2.db")
> all <- ALL[,names(classes)]
> probesIDS <- row.names(exprs(all))
> featureNames(all@assayData)<-unlist(mget(probesIDS, hgu95av2ENTREZID))
> all <- all[(!is.na(row.names(exprs(all))))]
```

At this point, we compute the intersection between pathway nodes and the genes for which an expression value is available. Thus we obtain a subgraph of the original graph. Moreover, we can extract from the expression set a smaller expression set corresponding to the expression of the genes in the pathway under investigation.

```
> library(graph)
> genes <- intersect(genes, row.names(exprs(all)))
> graph <- subGraph(genes, graph)
> exp <- all[genes,,drop=FALSE]
> exp
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 68 features, 79 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: 01010 04007 ... 84004 (79 total)
  varLabels: cod diagnosis ... date last seen (21
    total)
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
  pubMedIds: 14684422 16243790
Annotation: hgu95av2
```

```
> dim(exprs(exp))
```

```
[1] 68 79
```

Note that the usage of 'exp' as ExpressionSet or exprs(exp) as expression matrix leads exactly at the same result. Here we will use the ExpressionSet but all our function can be used with a simple expression matrix.

The analysis is performed using *pathQ* function as follows:

```
> library(clipper)
> pathwayAnalysis <- pathQ(exp, classes, graph, nperm=100, alphaV=0.05, b=100)
> pathwayAnalysis
```

```
$alphaVar
[1] 0.29
```

```
$alphaMean
[1] 0
```

The returned list contains the pvalue for the test on the concentration matrices (alphaVar) and the pvalue for the test on the means (alphaMean).

3 Performing clipper analysis

After a global inspection and identification of the most interesting/impacted pathways (the global analysis we have seen in the previous section), it is important to focus on the genes that drive the differences between the two phenotypes. The following example shows how to identify the sub-paths mostly associated to the phenotype.

```
> clipped <- clipper(exp, classes, graph, "var", trZero=0.01, permute=FALSE)
> clipped[,1:5]
```

```
     startIdx endIdx maxIdx lenght         maxScore
1;4         1      4      2      4 4.28739923789794
1;9         1      9      2      5 3.68413614879047
1;10        1     10      3      6 6.90775527898214
1;17        1     17      7      9 23.8204074708892
1;23        1     23      2     10 1.84206807439524
1;24        1     24      2      5 3.68413614879047
```

The analyzed cliques are indexed by the maximum cardinality search (mcs) algorithm and identified hereafter with the index number. The result of the clipper analysis is the matrix described in the following. For each analyzed path (named as <starting clique index>;<ending clique index>) the followiong information are reported:

1. Index of the starting clique
2. Index of the ending clique
3. Index of the clique where the maximum value is reached
4. Length of the path
5. Maximum score of the path
6. Average score along the path
7. Percentage of path activation
8. Impact of the path on the entire pathway
9. Involved and significant cliques
10. Cliques forming the path
11. Genes forming the significant cliques
12. Genes forming the path

A deeper look at the clipper matrix reveals that many paths overlap. To help users in focusing on the best candidates, we devised a function to prune the paths that are already part of other ones. The pruning process is performed according to a dissimilarity threshold *thr* (if paths dissimilarity value is greater than *thr* they are retained).

```
> clipped <- prunePaths(clipped, thr=0.2)
> clipped[,1:5]
```

```
     startIdx endIdx maxIdx lenght         maxScore
1;4         1      4      2      4 4.28739923789794
1;17        1     17      7      9 23.8204074708892
```

After the pruning, the results are smaller and much clear to read and interpret.

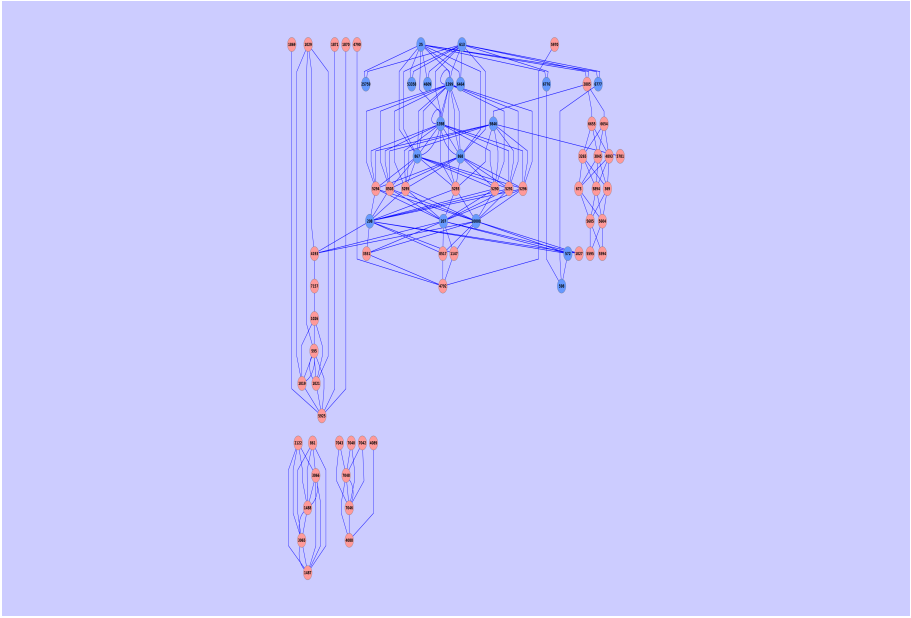

Figure 1: *clipper* visualization of the "Chronic myeloid leukemia" pathway from KEGG: nodes of the most significant path are reported in blue.

4 Visualizing clipper results.

clipper uses the *RCytoscape* package to connect to Cytoscape and display its results. Cytoscape is a Java based software specifically built to manage biological network complexity and for this reason it is widely used by the biological community.

```
> plotInCytoscape(graph, clipped[1,])
```

After the export of the pathway to Cytoscape, you can choose between different layouts. In figure 1 we show the KEGG "Chronic myeloid leukemia" pathway with an hierarchical layout and with the genes that belong to the most impacted path highlighted in blue.

5 Easy clipper analysis

The package provides also a function to easily run the analysis described in *Martini et al. 2012*. This analysis is able to start from a expression matrix and a pathway and returns the paths in the pathway that are altered between the two conditions.

```
> clipped <- easyClip(exp, classes, graph, method="mean")
> clipped[,1:5]
```

```
     startIdx endIdx maxIdx lenght         maxScore
1;17        1     17      9      9 51.0841685716396
1;13        1     13      3      3 12.7058667162116
```

A short summary of the results can be obtained with easyLook function.

```
> easyLook(clipped)
```

```
             maxScore
1;17 51.0841685716396
1;13 12.7058667162116
```

```
                                                                                              involvedGer
1;17 10000;1029;207;208;4193;1147;3551;8517;1398;1399;25;613;867;868;9846;572;6776;6777;25759;4609;53358;64
1;13                                                            10000;1029;207;208;4193;1019;1021;1026;595;71
```

6 Bibliography

References

[1] Chiaretti, S. and Li, X. and Gentleman, R. and Vitale, A. and Wang, K.S. and Mandelli, F. and Foà, R. and Ritz, J. **Gene expression profiles of B-lineage adult acute lymphocytic leukemia reveal genetic patterns that identify lineage derivation and distinct mechanisms of transformation.** Clinical cancer research. 2005.

[2] Martini P, Sales G, Massa MS, Chiogna M, Romualdi C. **Along signal paths: an empirical gene set approach exploiting pathway topology.** Nucleic Acids Research. 2012.

[3] Massa MS, Chiogna M, Romualdi C. **Gene set analysis exploiting the topology of a pathway.** BMC System Biol. 2010.

[4] Sales, G. and Calura, E. and Cavalieri, D. and Romualdi, C. **graphite-a Bioconductor package to convert pathway topology to gene network.** BMC bioinformatics. 2012.

[5] Laurent J, Pierre N and DudoitS. **More power via graph-structured tests for differential expression of gene networks** The Annals of Applied Statistics. 2012.

[6] Tarca AL, Draghici S, Khatri P, Hassan SS, Mittal P, and Kim J, and Kim CJ, Kusanovic JP and Romero R. **A novel signaling pathway impact analysis** Bioinformatics. 2009.

[7] Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, and others. **Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles** Proceedings of the National Academy of Sciences of the United States of America. 2005.