MGFM: Marker Gene Finder in Microarray gene expression data

Khadija El Amrani *†

October 17, 2016

Contents

1 Introduction

Identification of marker genes associated with a specific tissue/cell type is a fundamental challenge in genetic and genomic research. In addition to other genes, marker genes are of great importance for understanding the gene function, the molecular mechanisms underlying complex diseases, and may lead to the development of new drugs. We developed a new bioinformatic tool to predict marker genes from microarray gene expression data sets.
`MGFM` is a package enabling the detection of marker genes from microarray gene expression data sets.

*Charité-Universitätsmedizin Berlin, Berlin Brandenburg Center for Regenerative Therapies (BCRT), 13353 Berlin, Germany
†Package maintainer, Email: `khadija.el-amrani@charite.de`

2 Requirements

The tool expects replicates for each sample type. Using replicates has the advantage of increased precision of gene expression measurements and allows smaller changes to be detected. It is not necessary to use the same number of replicates for all sample types. Normalization is necessary before any analysis to ensure that differences in intensities are indeed due to differential expression, and not to some experimental factors that add systematic biases to the measurements. Hence, for reliable results normalization of data is mandatory. When combining data from different studies, other procedures should be applied to adjust for batch effects.

3 Contents of the package

The `MGFM` package contains the following objects:

```
> library("MGFM")
> ls("package:MGFM")

[1] "ds2.mat"        "getHtmlpage"    "getMarkerGenes"
```

The function `getMarkerGenes()` is the intended user-level interface, and `ds2.mat` is a normalized example data set that is used for demonstration.

3.1 getMarkerGenes

`getMarkerGenes()` is the main function and it returns a list of marker genes associated with each given sample type.

3.1.1 Parameter Settings

1. *data.mat*: The microarray expression data in matrix format with probe sets corresponding to rows and samples corresponding to columns. Please note that replicate samples should have the same label.

2. *samples2compare* (optional): A character vector with the sample names to be compared (e.g. c("liver", "lung", "brain")). By default all samples are used.

3. *annotate* (optional): A boolean value. If TRUE the gene symbol and the entrez gene id are shown. Default is TRUE. For mapping between microarray probe sets and genes, Bioconductor annotation packages (ChipDb) are used.

4. *chip*: Chip name.

5. *score.cutoff* (optional): It can take values in the interval [0,1]. This value is used to filter the markers according to the specificity score. The default value is 1 (no filtering)

3.1.2 Output

The function `getMarkerGenes()` returns a list as output. The entries of the result list contain the markers that are associated with each given sample type. For each marker the probe set, the gene symbol, the entrez gene id and the corresponding specificity score are shown in this order.

3.2 getHtmlpage

`getHtmlpage()` is a function to build HTML pages to show marker genes as tables with one row per marker gene, with links to Affymetrix, GenBank and Entrez Gene.

4 Example data

`ds1.mat`: is a microarray gene expression data set derived from 5 tissue types (lung, liver, heart atrium, kidney cortex, and midbrain) from the Serie GSE3526 [1] from the Gene Expression Omnibus (GEO) database [2]. Each tissue type is represented by 3 replicates. The following samples are used: GSM80699, GSM80700, GSM80701, GSM80654, GSM80655, GSM80656, GSM80686, GSM80687, GSM80688, GSM80728, GSM80729, GSM80730, GSM80707, GSM80710 and GSM80712.

`ds2.mat`: is a microarray expression data set derived from 5 tissue types (lung, liver, heart, kidney, and brain) from two GEO Series GSE1133 [3] and GSE2361 [4]. Each tissue type is represented by 3 replicates. The following samples are used: GSM44702, GSM18953, GSM18954, GSM44704, GSM18949, GSM18950, GSM44690, GSM18921, GSM18922, GSM44675, GSM18955, GSM18956, GSM44671, GSM18951 and GSM18952.

```
> data("ds2.mat")
> dim(ds2.mat)

[1] 22283    15

> colnames(ds2.mat)

 [1] "liver"  "liver"  "liver"  "lung"   "lung"   "lung"
 [7] "brain"  "brain"  "brain"  "kidney" "kidney" "kidney"
[13] "heart"  "heart"  "heart"
```

`ds3.mat`: is a microarray gene expression data set derived from 4 tissue types (lung, liver, heart, and kidney) from the GEO DataSet GDS596. Each tissue type is represented by 2 replicates. The following samples are used: GSM18953, GSM18954, GSM18949, GSM18950, GSM18951, GSM18952, GSM18955, GSM18956.
Since the size of the package submitted to Bioconductor is limited to 4MB, only the data matrix termed ds2.mat is included in the package. The other two data sets are available from GEO website.

5 Hierachical clustering

To evaluate the similarity of the samples, hierarchical clustering based on the expression values was performed using the R function hclust (using the Euclidian distance and the average linkage as clustering method). Figure 1 shows the clustering dendrogram of the samples in data set 1. The horizontal axis gives the distance between the clusters. As expected, all replicate samples of a tissue type are clustered together. Figure 2 shows the clustering dendrogram of samples of data set 2. The samples from the two studies are labeled with 1133 or 2361 according to GSE1133 or GSE2361, respectively. The samples of each study are clustered together. Hence, further normalization of the data is necessary. Figure 3 shows the clustering dendrogram after comBat normalization. As illustrated the differences are removed and the samples of each tissue type cluster together.

6 Normalization

The function `justRMA()` (robust multi-array average) from the R package *affy* [5] was used for background correction, normalization, and summarization of the AffyBatch probe-level data for data set 1 and 2. In addition, data set 2 was normalized using ComBat (Combating Batch Effects When Combining Batches of Gene Expression Microarray Data) method [6] from the R-package *sva* in order to remove batch effects. Please note that all samples of a study were considered by the normalization. The samples were selected after normalization.

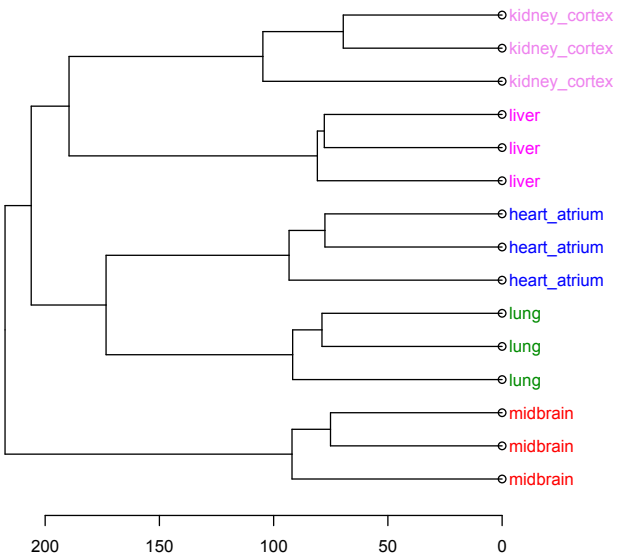

Figure 1: Hierarchical clustering of samples of data set 1 based on their gene expression values. Groups of samples of a tissue type are colored identically.

7 Marker search

To use the package, we should load it first.

```
>   library(MGFM)

> data("ds2.mat")
> require(hgu133a.db)
> marker.list2 <- getMarkerGenes(ds2.mat, samples2compare="all", annotate=TRUE,  chip="hgu133a",s
> names(marker.list2)

[1] "liver_markers"  "lung_markers"   "brain_markers"
[4] "kidney_markers" "heart_markers"

> # show the first 20 markers of liver
> marker.list2[["liver_markers"]][1:20]

 [1] "219466_s_at : APOA2 : 336 : 0.36"
 [2] "205041_s_at : ORM1,ORM2 : 5004,5005 : 0.38"
 [3] "219465_at : APOA2 : 336 : 0.38"
 [4] "205820_s_at : APOC3 : 345 : 0.39"
 [5] "1431_at : CYP2E1 : 1571 : 0.42"
 [6] "205477_s_at : AMBP : 259 : 0.44"
 [7] "204965_at : GC : 2638 : 0.45"
 [8] "205040_at : ORM1 : 5004 : 0.45"
 [9] "210929_s_at : AHSG : 197 : 0.45"
[10] "208147_s_at : CYP2C8 : 1558 : 0.46"
[11] "210049_at : SERPINC1 : 462 : 0.47"
```

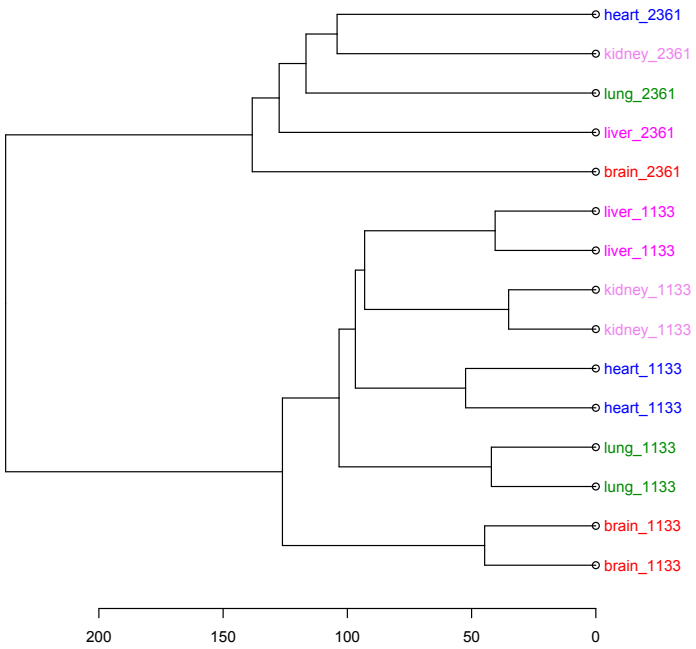

Figure 2: Hierarchical clustering of samples of data set 2 based on their gene expression values without ComBat normalization. Groups of samples of a tissue type are colored identically.

```
[12] "209975_at : CYP2E1 : 1571 : 0.48"
[13] "209976_s_at : CYP2E1 : 1571 : 0.48"
[14] "211298_s_at : ALB : 213 : 0.48"
[15] "214465_at : ORM2,ORM1 : 5005,5004 : 0.48"
[16] "219612_s_at : FGG : 2266 : 0.48"
[17] "204534_at : VTN,SEBOX : 7448,645832 : 0.5"
[18] "205216_s_at : APOH : 350 : 0.51"
[19] "206350_at : APCS : 325 : 0.51"
[20] "206697_s_at : HP : 3240 : 0.51"

> data("ds2.mat")
> # If no annotation (mapping of probe sets to genes) is desired, no chip name is needed.
> marker.list2 <- getMarkerGenes(ds2.mat, samples2compare="all", annotate=FALSE, score.cutoff=1)
> names(marker.list2)

[1] "liver_markers"  "lung_markers"   "brain_markers"
[4] "kidney_markers" "heart_markers"

> # show the first 20 markers of lung
> marker.list2[["lung_markers"]][1:20]

 [1] "37004_at : 0.46"    "218835_at : 0.48"
 [3] "214387_x_at : 0.49" "205982_x_at : 0.51"
 [5] "204446_s_at : 0.52" "209810_at : 0.52"
 [7] "211735_x_at : 0.53" "36711_at : 0.53"
```

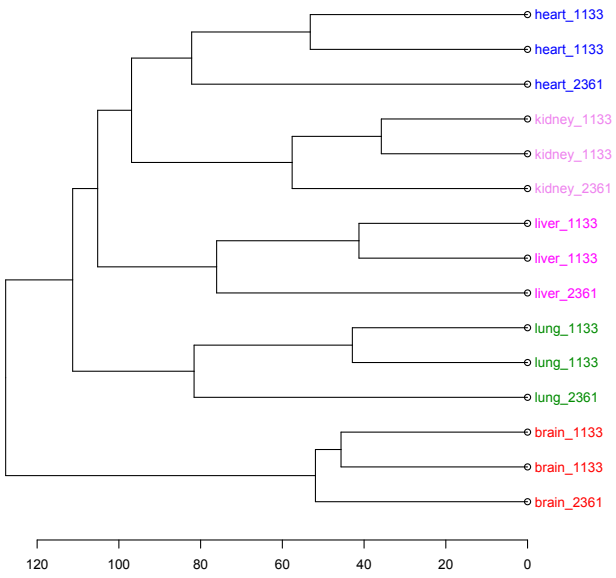

Figure 3: Hierarchical clustering of samples of data set 2 based on their gene expression values after ComBat normalization. Groups of samples of a tissue type are colored identically.

```
 [9] "210096_at : 0.54"    "213936_x_at : 0.54"
[11] "205725_at : 0.55"    "210081_at : 0.55"
[13] "215454_x_at : 0.56" "214199_at : 0.57"
[15] "219230_at : 0.57"    "217028_at : 0.58"
[17] "34210_at : 0.58"     "38691_s_at : 0.58"
[19] "205569_at : 0.59"    "211024_s_at : 0.59"
```

8 MGFM algorithm details

Marker genes are identified as follows:

1. **Sort of expression values for each probe set:** In this step the expression values are sorted in decreasing order.

2. **Marker selection**: A probe set is a potential candidate marker of a sample type if the highest expression values represent all replicates of this sample type. We consider a cut-point as the position in the sorted expression vector that segregates different sample types. For marker selection we consider two cut-points. The first cut-point segregates the replicates of the sample type with the highest expression values from the rest of the samples. The second cut-point is set at the first position in which a segregation of the different sample types is possible. If no such cut-point is found, the cut-point is set at the end of the sorted expression vector.

3. **Calculation of mean expression values**: Each cut-point segregates elements of sample types into two distinct sample-blocks. For each probe set, the expression levels of the two sample-blocks are summarized as the mean of expression values of the samples in these blocks.

4. **Score the probeset**: For scoring the probe set the first cut-point is relevant. The score is defined as the ratio of the second and first value in the vector of mean expression values of a probe set. This score is used to rank the markers according to their specificity. The score values range from 0 to 1. Values near 0 would indicate high specificity and large values closer to 1 would indicate low specificity.

   This approach of marker selection is strict, since a probeset is not considered as marker if the first highest expression values correspond to more than one type of samples. Using this method, markers that are associated with more than one type of samples will be missed.

9 Conclusion

The development of this tool was motivated by the desire to provide a software package with a fast runtime that enables the user to get marker genes associated with a set of samples of interest. A further objective of this tool was to enable the user to modify the set of samples of interest by adding or removing samples in a simple way.
In summary, the main contribution of the application presented herein are: i) The application has a running time of some seconds per analysis. This is achieved by sorting the gene expression values instead of using gene differential expression. ii) The tool offers the user the possibility to modify the set of samples by easily removing or adding new samples.

10 R sessionInfo

The results in this file were generated using the following packages:

```
> sessionInfo()

R version 3.3.1 (2016-06-21)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows Server 2012 R2 x64 (build 9600)

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] parallel  stats4    stats     graphics  grDevices
[6] utils     datasets  methods   base

other attached packages:
 [1] hgu133a.db_3.2.3     org.Hs.eg.db_3.4.0
 [3] MGFM_1.8.0           annotate_1.52.0
 [5] XML_3.98-1.4         AnnotationDbi_1.36.0
 [7] IRanges_2.8.0        S4Vectors_0.12.0
 [9] Biobase_2.34.0       BiocGenerics_0.20.0

loaded via a namespace (and not attached):
[1] bitops_1.0-6   xtable_1.8-2   DBI_0.5-1
[4] RSQLite_1.0.0  tools_3.3.1    RCurl_1.95-4.8
```

References

[1] Richard B Roth, Peter Hevezi, Jerry Lee, Dorian Willhite, Sandra M Lechner, Alan C Foster, and Albert Zlotnik. Gene expression analyses reveal molecular relationships among 20 regions of the human CNS. *Neurogenetics*, 7(2):67–80, May 2006.

[2] Ron Edgar, Michael Domrachev, and Alex E Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30:207–210, 2002.

[3] Andrew I Su, Tim Wiltshire, Serge Batalov, Hilmar Lapp, Keith A Ching, David Block, Jie Zhang, Richard Soden, Mimi Hayakawa, Gabriel Kreiman, Michael P Cooke, John R Walker, and John B Hogenesch. A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16):6062–7, April 2004.

[4] Xijin Ge, Shogo Yamamoto, Shuichi Tsutsumi, Yutaka Midorikawa, Sigeo Ihara, San Ming Wang, and Hiroyuki Aburatani. Interpreting expression profiles of cancers by genome-wide survey of breadth of expression in normal tissues. *Genomics*, 86(2):127–41, August 2005.

[5] R. Irizarry, L. Gautier, and L. Cope. An r package for analyses of affymetrix oligonucleotide arrays. In G. Parmigiani, E.S. Garrett, R.A. Irizarry, and S.L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*. Springer, New York, 2002.

[6] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics (Oxford, England)*, 8(1):118–27, January 2007.

[7] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. ISBN 3-900051-07-0.