

Bioconductor mAPKL Package

Argiris Sakellariou^{1,2}

[1cm] ¹ Biomedical Informatics Unit, Biomedical Research Foundation of the Academy of Athens, Athens, Greece

[0cm] ² Department of Informatics and Telecommunications, National and Kapodistrian Univ. of Athens, Athens, Greece

[0cm] argisake@gmail.com

November 1, 2022

Contents

1	Introduction	2
2	Identification of differentially expressed genes	2
2.1	Breast cancer data	2
2.2	Data normalization and transformation	4
2.3	mAPKL gene selection	4
2.4	Building and evaluating classification models	8
3	Advanced usage of the package	9
3.1	Annotation analysis	9
3.2	Network characteristics	13
4	Reporting	16
5	Session info	16
6	Reference	17

1 Introduction

The mAPKL bioconductor R package implements a hybrid gene selection method, which is based on the hypothesis that among the statistically significant genes in a ranked list, there should be clusters of genes that share similar biological functions related to the investigated disease. Thus, instead of keeping a number of N top ranked genes, it would be more appropriate to define and keep a number of gene cluster exemplars.

The proposed methodology combines filtering and cluster analysis to select a small yet highly discriminatory set of non-redundant genes. Regarding the filtering step, a multiple hypothesis testing approach from *multtest* r-package (maxT) is employed to rank the genes of the training set according to their differential expression. Then the top N genes (e.g. $N = 200$) are reserved for cluster analysis. First the index of Krzanowski and Lai as included in the *ClusterSim* r-package is applied on the disease samples of the training set to determine the number of clusters. The Krzanowski and Lai index is defined by $DIFF(k) = (k - 1)^{\frac{2}{p}} W_{k-1} - k^{\frac{2}{p}} W_k$ when choosing the number of clusters (k) to maximize the quantity $KL(k) = \left| \frac{DIFF(k)}{DIFF_{(k+1)}} \right|$. The W_k denotes the within-cluster sum of squared errors.

Finally, cluster analysis is performed with the aid of Affinity Propagation (AP) clustering algorithm, which detects n ($n = k$ the Krzanowski and Lai index) clusters among the top N genes, the so called exemplars. Those n exemplars are expected to form a classifier that shall discriminate between normal and disease samples (Sakellariou et al. 2012, *BMC Bioinformatics* **13**:270). This package implements the pre-mentioned methodology through a core function, the *mAPKL*. In the upcoming sections follows a guidance of the included functions and its functionality through differential expression analysis scenarios on a breast cancer dataset (GSE5764) which is part of the *mAPKLData* package.

2 Identification of differentially expressed genes

2.1 Breast cancer data

Throughout this tutorial we utilized a publicly available breast cancer dataset comprised of 30 samples, where 20 of them represent normal cases and the remaining 10 samples stand for tumor cases. We first load the package and then the breast cancer data. Then with the aid of the *sampling* function we create a separate training and validation sets where 60% of the samples will be used for training and the rest 40% of the samples will be used for evaluation purposes.

Bioconductor mAPKL Package

```
library(mAPKL)
library(mAPKLData)
data(mAPKLData)
varLabels(mAPKLData)
breast <- sampling(Data=mAPKLData, valPercent=40, classLabels="type", seed=135)

## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

The *sampling* function returns an S3 class (breast) with two eSet class objects that nest the relevant training and validation samples. Those two objects are used throughout the rest of the analysis.

```
breast

## $trainData
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 54675 features, 18 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: GSM134588 GSM134687 ... GSM134695 (18 total)
##   varLabels: title type
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
##   pubMedIds: 17389037
## Annotation: hgu133plus2
##
## $testData
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 54675 features, 12 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: GSM134584 GSM134696 ... GSM134698 (12 total)
##   varLabels: title type
```

```
## varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## pubMedIds: 17389037
## Annotation: hgu133plus2
```

2.2 Data normalization and transformation

We perform normalization to the expression values through the *preprocess* function.

```
normTrainData <- preprocess(breast$trainData)
normTestData <- preprocess(breast$testData)
```

The *preprocess* function produces a list of several available normalization and transformation options. Besides density plots per method are produced and saved to current working directory to assist the user to decide upon which method to select before proceeding to mAPKL analysis.

```
attributes(normTrainData)

## $names
## [1] "rawdata"      "mc.normdata"  "z.normdata"  "q.normdata"
## [5] "cl.normdata"  "mcl2.normdata" "zl2.normdata" "ql2.normdata"
## [9] "cll2.normdata"
```

The following graph presents the density plots of 8 possible normalization process with or without log2 transformation. The *preprocess* function applies all of them and it is up to the user, which one will engage for the rest of the analysis. In brief, the available approaches are mean-centering, z-score, quantile, and cyclic loess. During this case study we will proceed with the expression values following log2 transformation and cyclic loess normalization.

2.3 mAPKL gene selection

In this example we employ the expression values of log2 transformation and cyclic loess normalization to proceed with the *mAPKL* analysis.

Bioconductor mAPKL Package



Figure 1: Density plots of normalized intensity values

```
exprs(breast$trainData) <- normTrainData$cLL2.normdata
exprs(breast$testData) <- normTestData$cLL2.normdata
out.cLL2 <- mAPKL(trObj = breast$trainData, classLabels = "type",
  valObj = breast$testData, dataType = 7)

## Warning in xtfrm.data.frame(x): cannot xtfrm data frames

## b=10 b=20 b=30 b=40 b=50 b=60 b=70 b=80 b=90 b=100
## b=110 b=120 b=130 b=140 b=150 b=160 b=170 b=180 b=190 b=200
```

Bioconductor mAPKL Package

```
## b=210 b=220 b=230 b=240 b=250 b=260 b=270 b=280 b=290 b=300
## b=310 b=320 b=330 b=340 b=350 b=360 b=370 b=380 b=390 b=400
## b=410 b=420 b=430 b=440 b=450 b=460 b=470 b=480 b=490 b=500
## b=510 b=520 b=530 b=540 b=550 b=560 b=570 b=580 b=590 b=600
## b=610 b=620 b=630 b=640 b=650 b=660 b=670 b=680 b=690 b=700
## b=710 b=720 b=730 b=740 b=750 b=760 b=770 b=780 b=790 b=800
## b=810 b=820 b=830 b=840 b=850 b=860 b=870 b=880 b=890 b=900
## b=910 b=920 b=930 b=940 b=950 b=960 b=970 b=980 b=990 b=1000

## Please wait! The (KL) cluster indexing may take several minutes...

## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'

## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'

## Warning in is.nan(d) || is.nan(d) || is.infinite(d): 'length(x)
= 19900 > 1' in coercion to 'logical(1)'

## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'

## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'

## Warning in is.nan(d) || is.nan(d) || is.infinite(d): 'length(x)
= 19900 > 1' in coercion to 'logical(1)'

## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'

## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'

## Warning in is.nan(d) || is.nan(d) || is.infinite(d): 'length(x)
= 19900 > 1' in coercion to 'logical(1)'
```

Bioconductor mAPKL Package

```
## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'
```

```
## Warning in is.nan(d) || is.nan(d): 'length(x) = 19900 > 1' in coercion
to 'logical(1)'
```

```
## Warning in is.nan(d) || is.nan(d) || is.infinite(d): 'length(x)
= 19900 > 1' in coercion to 'logical(1)'
```

```
## Asking for 22 number of clusters
```

```
## Warning in .local(s, x = x, ...): algorithm did not converge; turn
on details
```

```
## and call plot() to monitor net similarity. Consider
## increasing 'maxits' and 'convits', and, if oscillations occur
## also increasing damping factor 'lam'.
```

```
## Warning in .local(s, x = x, ...): algorithm did not converge; turn
on details
```

```
## and call plot() to monitor net similarity. Consider
## increasing 'maxits' and 'convits', and, if oscillations occur
## also increasing damping factor 'lam'.
```

```
## Warning in .local(s, x = x, ...): algorithm did not converge; turn
on details
```

```
## and call plot() to monitor net similarity. Consider
## increasing 'maxits' and 'convits', and, if oscillations occur
## also increasing damping factor 'lam'.
```

```
## Warning in .local(s, x = x, ...): algorithm did not converge; turn
on details
```

```
## and call plot() to monitor net similarity. Consider
## increasing 'maxits' and 'convits', and, if oscillations occur
## also increasing damping factor 'lam'.
```

```
## fc according to limma
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

2.4 Building and evaluating classification models

After having get the exemplars from *mAPKL* analysis we build an SVM classifier to test their discriminatory performance. Regarding the SVM setup, we utilize a linear kernel for which the cost attribute is inferred by the `tune.svm` function. however, the user may freely use another kernel and a different Cross Validation approach than 5-folds.

```
clasPred <- classification(out.clL2@exemplTrain, "type", out.clL2@exemplTest)

## The training set has 10 Negative and 8 Positive samples. Using
## k-fold=5 C-V

## ##### THE BEST PARAMETERS TUNING STAGE #####

## ##### THE TRAINING STAGE #####

##
## Call:
## svm.default(x = train.mtx, y = lbls, scale = FALSE, type = "C-classification",
## kernel = "linear", gamma = best_gamma, cost = best_cost, cross = k_fold)
##
##
## Parameters:
## SVM-Type: C-classification
## SVM-Kernel: linear
## cost: 2
##
## Number of Support Vectors: 6

## ##### THE PREDICTION STAGE #####

## Test Labels Prediction Labels
## GSM134584 0 1
## GSM134696 0 0
## GSM134705 0 0
## GSM134702 0 0
## GSM134709 0 0
## GSM134693 0 0
## GSM134708 0 0
## GSM134697 0 0
```



```
## GSM134703      0      1
## GSM134586      0      0
## GSM134710      1      1
## GSM134698      1      1
## Negative samples: 10
## Positive samples: 2
## TN=8
## FP=2
## TP=2
## FN=0
## AUC=0.90
## Accuracy=83.00
## MCC=0.63
## Specificity=0.80
## Sensitivity=1.00
```

The output of the *classification* inform us about the SVM set up, the number of Support Vectors and finally show the the predicted labels along with the initial. In this example there is a validation set different from the training set and therefore we may use the respective labels to obtain the performance characteristics. The relevant function *metrics* called inside the *classification* function, calculates five key measures: the Area Under the ROC curve AUC, the classification accuracy, the Matthews correlation coefficient MCC classification measure, the degree of true negative's identification Specificity, and finally the degree of true positive's identification Sensitivity.

3 Advanced usage of the package

3.1 Annotation analysis

For each contemporary chip technology, there is a relevant annotation file, in which the the user may drag several *genome oriented* information. Regarding the breast cancer microarray data, the gene expression values were stored on

Bioconductor mAPKL Package

Affumetrix gene chips. Using the *annotate* function, the user may obtain several info related to probe id, gene symbol, Entrez id, ensembl id, and chromosomal location.

```
gene.info <- annotate(out.clL2@exemplars, "hgu133plus2.db")
gene.info@results
```

##	PROBEID	SYMBOL	ENTREZID	ENSEMBL	MAP
## 1	239492_at	SEC14L4	284904	ENSG00000133488	22q12.2
## 2	229947_at	PI15	51050	ENSG00000137558	8q21.13
## 3	1556499_s_at	COL1A1	1277	ENSG00000108821	17q21.33
## 4	201069_at	MMP2	4313	ENSG00000087245	16q12.2
## 5	214598_at	CLDN8	9073	ENSG00000156284	21q22.11
## 6	217127_at	CTH	1491	ENSG00000116761	1p31.1
## 7	37892_at	COL11A1	1301	ENSG00000060718	1p21.1
## 8	33768_at	DMWD	1762	ENSG00000185800	19q13.32
## 9	217637_at	KCNB1	3745	ENSG00000158445	20q13.13
## 10	1569828_at	LOC101928107	101928107	ENSG00000226527	21q22.11
## 11	210170_at	PDLIM3	27295	ENSG00000154553	4q35.1
## 12	212236_x_at	KRT17	3872	ENSG00000128422	17q21.2
## 13	220932_at	<NA>	<NA>	<NA>	<NA>
## 14	225664_at	COL12A1	1303	ENSG00000111799	6q13-q14.1
## 15	206391_at	RARRES1	5918	ENSG00000118849	3q25.32
## 16	243177_at	<NA>	<NA>	<NA>	<NA>
## 17	1565733_at	<NA>	<NA>	<NA>	<NA>
## 18	1555926_a_at	<NA>	<NA>	<NA>	<NA>
## 19	201377_at	UBAP2L	9898	ENSG00000143569	1q21.3
## 20	220033_at	<NA>	<NA>	<NA>	<NA>
## 21	205044_at	GABRP	2568	ENSG00000094755	5q35.1
## 22	215131_at	IQCK	124152	ENSG00000174628	16p12.3

We may exploit the output of the *annotate* function to extent our analysis. For instance, we may perform *pathway analysis* on the exemplars. For this purpose we will utilize the *probes2pathways* function that utilizes the *reactome.db* package. This function employs the probe ids to identify the relevant pathways.

```
probes2pathways(gene.info)
```

```
##
```

R-HSA-10958

Bioconductor mAPKL Package

```
## "Homo sapiens: Hemostasis
## R-HSA-114604
## "Homo sapiens: GPVI-mediated activation cascade
## R-HSA-128021
## "Homo sapiens: Adaptive Immune System
## R-HSA-144249
## "Homo sapiens: Collagen degradation
## R-HSA-147422
## "Homo sapiens: Degradation of the extracellular matrix
## R-HSA-147424
## "Homo sapiens: Extracellular matrix organization
## R-HSA-147429
## "Homo sapiens: Collagen formation
## R-HSA-16258
## "Homo sapiens: Signal Transduction
## R-HSA-165081
## "Homo sapiens: Collagen biosynthesis and modifying enzymes
## R-HSA-16825
## "Homo sapiens: Immune System
## R-HSA-19893
## "Homo sapiens: Immunoregulatory interactions between a Lymphoid and a non-Lymphoid cell
## R-HSA-202209
## "Homo sapiens: Assembly of collagen fibrils and other multimeric structures
## R-HSA-20273
## "Homo sapiens: Cell surface interactions at the vascular wall
## R-HSA-21243
## "Homo sapiens: Generic Transcription Pathway
## R-HSA-21608
## "Homo sapiens: Integrin cell surface interactions
## R-HSA-217378
## "Homo sapiens: Binding and Uptake of Ligands by Scavenger Receptors
## R-HSA-221432
## "Homo sapiens: Anchoring fibril formation
## R-HSA-224391
## "Homo sapiens: Crosslinking of collagen fibrils
```

Bioconductor mAPKL Package

```
## R-HSA-3000170
## "Homo sapiens: Syndecan interactions
## R-HSA-3000170
## "Homo sapiens: Non-integrin membrane-ECM interactions
## R-HSA-3000170
## "Homo sapiens: ECM proteoglycans
## R-HSA-3000480
## "Homo sapiens: Scavenging by Class A Receptors
## R-HSA-430110
## "Homo sapiens: GP1b-IX-V activation signalling
## R-HSA-565365
## "Homo sapiens: Vesicle-mediated transport
## R-HSA-680683
## "Homo sapiens: Signaling by MET
## R-HSA-73850
## "Homo sapiens: RNA Polymerase II Transcription
## R-HSA-74160
## "Homo sapiens: Gene expression (Transcription)
## R-HSA-75890
## "Homo sapiens: Platelet Adhesion to exposed collagen
## R-HSA-76000
## "Homo sapiens: Platelet activation, signaling and aggregation
## R-HSA-76000
## "Homo sapiens: Platelet Aggregation (Plug Formation)
## R-HSA-887408
## "Homo sapiens: MET activates PTK2 signaling
## R-HSA-887587
## "Homo sapiens: MET promotes cell motility
## R-HSA-887816
## "Homo sapiens: Transcriptional regulation by RUNX2
## R-HSA-894097
## "Homo sapiens: RUNX2 regulates osteoblast differentiation
## R-HSA-894132
## "Homo sapiens: RUNX2 regulates bone development
## R-HSA-894821
```

```
## "Homo sapiens: Collagen chain trimerization
## R-HSA-900693
## "Homo sapiens: Signaling by Receptor Tyrosine Kinases
```

3.2 Network characteristics

Regarding the network characteristics, we compute through the *netwAttr* function three different types of centralities (degree, closeness, betweenness) and a measure for clustering coefficient called transitivity. The degree centrality of a node refers to the number of connections or edges of that node to other nodes. The closeness centrality describes the reciprocal accumulated shortest length distance from a node to all other connected nodes. The betweenness centrality depicts the number of times a node intervenes along the shortest path of two other nodes. Transitivity measures the degree of nodes to create clusters within a network. For all four network measures we provide both global and local values. Furthermore, we compose an edge list (Node1-Node2-weight) based on the *N* top ranked genes. We may exploit that measures to depict the exemplars' network characteristics

```
net.attr <- netwAttr(out.clL2)
wDegreeL <- net.attr@degree$wdegreeL[out.clL2@exemplars]
wClosenessL <- net.attr@closeness$wclosenessL[out.clL2@exemplars]
wBetweennessL <- net.attr@betweenness$wbetweennessL[out.clL2@exemplars]
wTransitivityL <- net.attr@transitivity$wtransitivityL[out.clL2@exemplars]
```

```
Global.val <- c(net.attr@degree$wdegreeG, net.attr@closeness$wclosenessG,
  net.attr@betweenness$wbetweennessG, net.attr@transitivity$wtransitivityG)
```

```
Global.val <- round(Global.val, 2)
exempl.netattr <- rbind(wDegreeL, wClosenessL, wBetweennessL, wTransitivityL)
```

```
netAttr <- cbind(Global.val, exempl.netattr)
netAttr <- t(netAttr)
netAttr
```

##	wDegreeL	wClosenessL	wBetweennessL	wTransitivityL
## Global.val	350.90	3.24	1542.12	0.53
## 239492_at	387.54	2.09	0.00	0.53

Bioconductor mAPKL Package

##	229947_at	320.55	3.65	1342.00	0.51
##	1556499_s_at	363.44	1.51	0.00	0.54
##	201069_at	294.55	5.07	17550.00	0.51
##	214598_at	400.74	4.30	1564.00	0.51
##	217127_at	301.88	1.70	2.00	0.51
##	37892_at	384.02	2.67	0.00	0.54
##	33768_at	295.85	4.26	1286.00	0.51
##	217637_at	310.87	1.68	2.00	0.52
##	1569828_at	344.24	1.08	0.00	0.54
##	210170_at	310.57	1.44	0.00	0.50
##	212236_x_at	412.59	4.10	1176.00	0.56
##	220932_at	321.35	2.93	0.00	0.51
##	225664_at	366.51	3.22	764.00	0.53
##	206391_at	343.50	3.23	524.00	0.54
##	243177_at	364.64	3.59	4.00	0.51
##	1565733_at	390.27	3.66	470.00	0.52
##	1555926_a_at	352.14	3.99	3360.00	0.53
##	201377_at	499.64	1.99	397.00	0.54
##	220033_at	460.89	4.25	396.00	0.55
##	205044_at	310.78	1.09	0.00	0.50
##	215131_at	308.69	1.73	0.00	0.56

and identify potential hubs. The calculations of this example are based on the "clr" network reconstruction method. However, there are for the time being two more options, including the "aracne.a" and "aracne.m".

```
# For local degree > global + standard deviation
sdev <- sd(net.attr@degree$WdegreeL)
msd <- net.attr@degree$WdegreeG + sdev
hubs <- wDegreeL[which(wDegreeL > msd)]
hubs

##      214598_at 212236_x_at  201377_at  220033_at
##      400.74   412.59     499.64     460.89
```

Bioconductor mAPKL Package

Finally, we may plot the network for those nodes that their local weighted degree is greater than Global weighted degree plus 2 times the standard deviation. We set this rule for both significance and illustration purposes (that edge list has dimension 604 x 3).

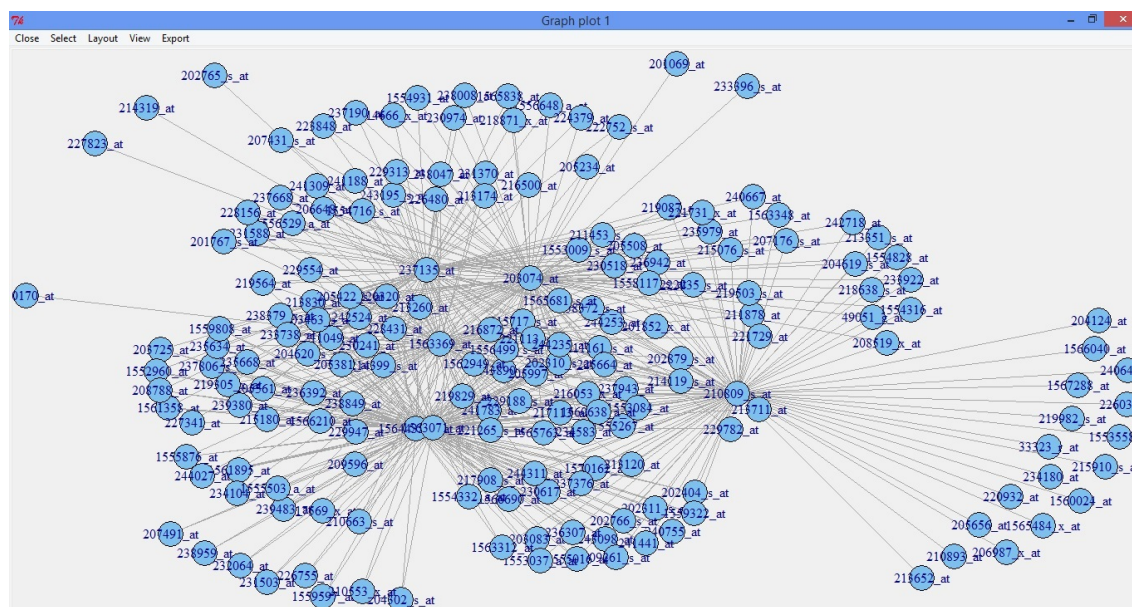


Figure 2: Degree centrality network

```
sdev <- sd(net.attr@degree$WdegreeL)
ms2d <- net.attr@degree$WdegreeG + 2 * sdev
net <- net.attr@degree$WdegreeL[which(net.attr@degree$WdegreeL >
  ms2d)]
idx <- which(net.attr@edgelist[, 1] %in% names(net))
new.edgeList <- net.attr@edgelist[idx, ]
dim(new.edgeList)

## [1] 428 3

require(igraph)
g = graph.data.frame(new.edgeList, directed = FALSE)
# tkplot(g, layout=layout.fruchterman.reingold)
```

4 Reporting

The overall analysis is summarized in an **html** report produced by the *report* function. It covers the dataset representation depicting the samples' names and their respective class labels, the exemplars section where statistical results and network characteristics are included. The classification performance section illustrates the performance metrics achieved in either cross-validation or hold-out validation. Finally, several annotation info are presented if an annotation analysis has occurred.

5 Session info

```
sessionInfo()

## R version 4.2.1 Patched (2022-07-09 r82577)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] C/en_US.UTF-8/en_US.UTF-8/C/en_GB/en_US.UTF-8
##
## attached base packages:
## [1] stats4      stats      graphics  grDevices  utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] igraph_1.3.5          hgu133plus2.db_3.13.0 org.Hs.eg.db_3.16.0
## [4] AnnotationDbi_1.60.0 IRanges_2.32.0        S4Vectors_0.36.0
## [7] mAPKLData_1.29.0      mAPKL_1.28.0          Biobase_2.58.0
## [10] BiocGenerics_0.44.0  knitr_1.40
##
## loaded via a namespace (and not attached):
```


Bioconductor mAPKL Package

```
## [1] Rcpp_1.0.9          parmigene_1.1.0      lattice_0.20-45
## [4] png_0.1-7           class_7.3-20         Biostrings_2.66.0
## [7] digest_0.6.30       R6_2.5.1             GenomeInfoDb_1.34.0
## [10] RSQLite_2.2.18      evaluate_0.17        e1071_1.7-12
## [13] httr_1.4.4          highr_0.9            zlibbioc_1.44.0
## [16] rlang_1.0.6         blob_1.2.3           Matrix_1.5-1
## [19] clusterSim_0.50-1   rmarkdown_2.17       apcluster_1.4.10
## [22] splines_4.2.1       stringr_1.4.1        RCurl_1.98-1.9
## [25] bit_4.0.4           proxy_0.4-27         compiler_4.2.1
## [28] xfun_0.34           pkgconfig_2.0.3      multtest_2.54.0
## [31] htmltools_0.5.3     KEGGREST_1.38.0      GenomeInfoDbData_1.2.9
## [34] crayon_1.5.2        MASS_7.3-58.1        bitops_1.0-7
## [37] grid_4.2.1          DBI_1.1.3            magrittr_2.0.3
## [40] formatR_1.12        cli_3.4.1            stringi_1.7.8
## [43] cachem_1.0.6        XVector_0.38.0       limma_3.54.0
## [46] vctrs_0.5.0         BiocStyle_2.26.0     tools_4.2.1
## [49] ade4_1.7-19         bit64_4.0.5          fastmap_1.1.0
## [52] survival_3.4-0      yaml_2.3.6           cluster_2.1.4
## [55] BiocManager_1.30.19 reactome.db_1.82.0    R2HTML_2.3.3
## [58] memoise_2.0.1
```

6 Reference

Sakellariou, A., D. Sanoudou, and G. Spyrou. "Combining Multiple Hypothesis Testing and Affinity Propagation Clustering Leads to Accurate, Robust and Sample Size Independent Classification on Gene Expression Data. " BMC Bioinformatics 13 (2012): 270.

<http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5764>

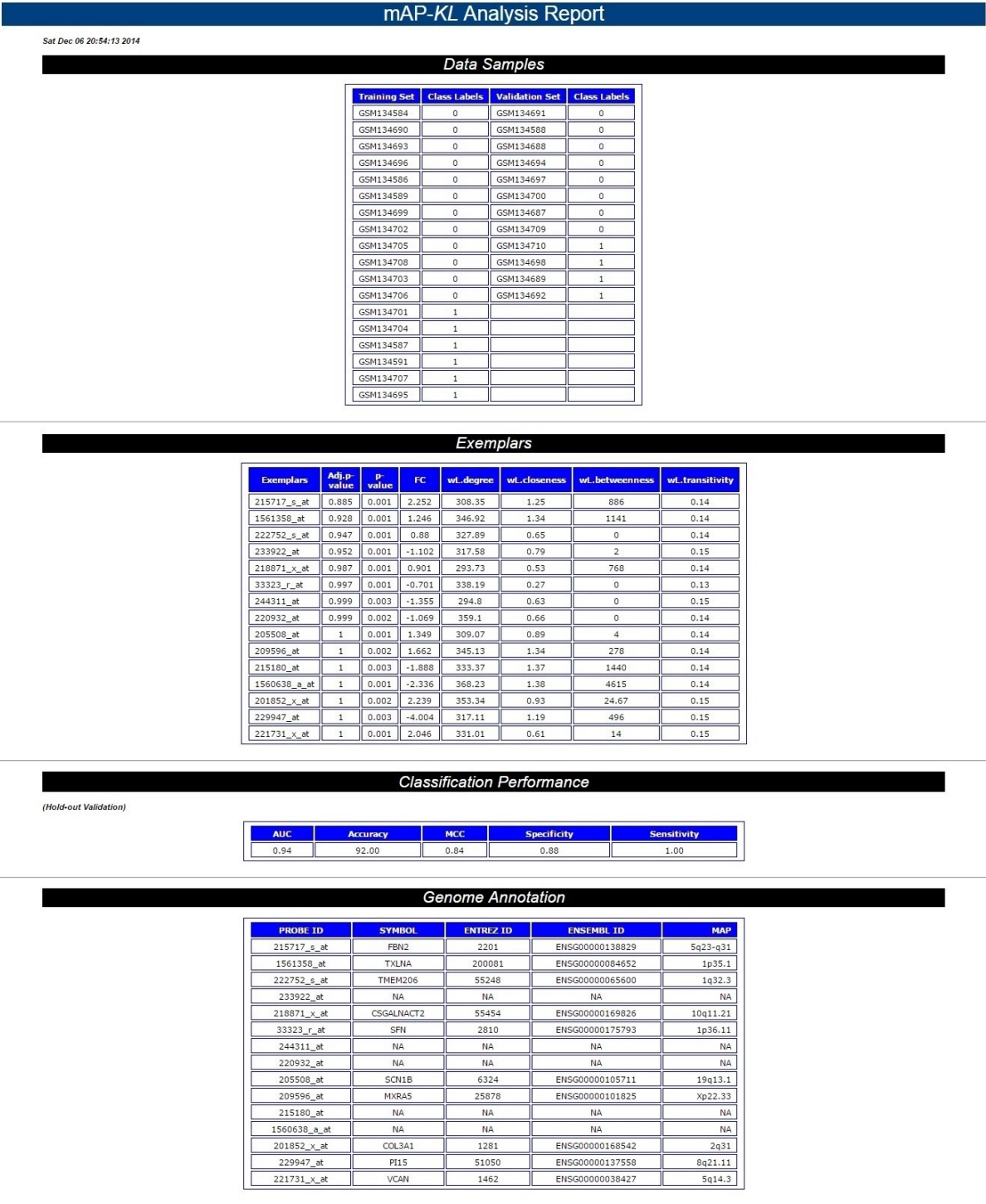


Figure 3: mAPKL analysis report