An R Package suite for Microarray Meta-analysis in Quality Control, Differentially Expressed Gene Analysis and Pathway Enrichment Detection Supplemental Document

Xingbin Wang * Jia Li Dongwan D. Kang Kui Shen George C Tseng

November 2, 2012

Contents

*Department of Human Genetics, Pittsburgh University . Email: xbw1@pitt.eud

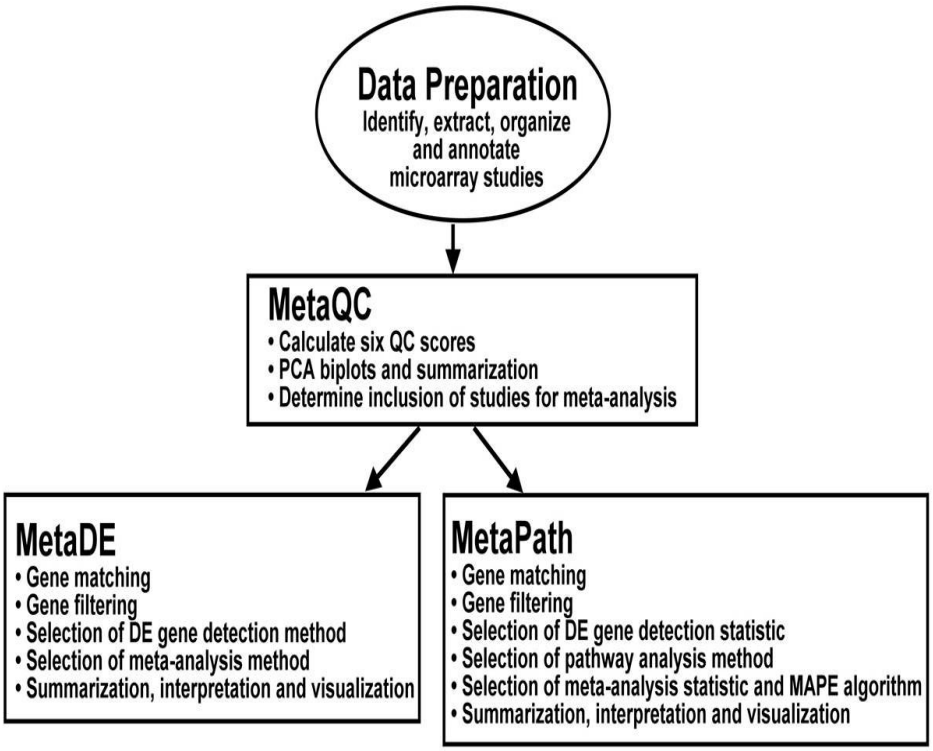


Figure 1: A diagram of meta-analysis pipeline using MetaQC, MetaDE and MetaPath.

1 INTRODUCTION

With the rapid advances and prevalence of high-throughput genomic technologies, integrating information of multiple relevant genomic studies has brought new challenges. Microarray meta-analysis has become a frequently used tool in biomedical research. Little effort, however, has been made to develop a systematic pipeline and user-friendly software. To fill this gap, we present MetaOmics, a suite of three R packages MetaQC, MetaDE and MetaPath, for quality control, differentially expressed gene identification and enriched pathway detection for microarray meta-analysis. MetaQC provides a quantitative and objective tool to assist study inclusion/exclusion criteria for meta-analysis. MetaDE and MetaPath are developed for candidate marker and pathway detection, that provide choices of marker detection, meta-analysis and pathway analysis methods. Figure 1 shows a generic diagram of meta-analysis pipeline using the three packages. After microarray studies are identified, extracted and annotated, MetaQC is applied to determine inclusion/exclusion criteria of the studies. MetaDE and MetaPath can then be used separately to detect candidate markers or pathways associated with disease outcome.

The MetaQC provides a main function,`"MetaQC"`,for calculating six quantitative quality control measures for quality control[4]: (1) internal homogeneity of co-expression structure among studies (internal quality control; IQC); (2) external consistency of co-expression structure correlating with pathway database (external quality control; EQC); (3) accuracy of differentially expressed gene detection (accuracy quality control; AQCg) or pathway identification (AQCp); (4) consistency of differential expression

ranking in genes (consistency quality control; CQCg) or pathways (CQCp). The package also provides a plot function to draw the PCA biplot for assisting visualization and decision. Results generate systematic suggestions to exclude problematic studies in microarray meta-analysis and potentially can be extended to GWAS or other types of genomic meta-analysis. The identified problematic studies can be scrutinized to identify technical and biological causes (e.g. sample size, platform, tissue collection, preprocessing etc) of their bad quality or irreproducibility for final inclusion/exclusion decision.

MetaDE package provides functions for conducting 12 major meta-analysis methods for differential expression analysis (see Table 1): Fisher [21, 10], Stouffer [24], adaptively weighted Fisher (AW)[16], minimum p-value (minP), maximum p-value (maxP)[30], rth ordered p-value (rOP) (Song and Tseng, 2012), fixed effects model (FEM), random effects model (REM)[3], rank product (rankProd)[14], naive sum of ranks and naive product of ranks [6]. Detailed algorithms, pros and cons of different methods have been discussed in a recent review paper [26]. In addition to selecting a meta-analysis method, two additional considerations are involved in the implementation: (1) Choice of test statistics: Different test statistics are available in the package for each type of outcome variable (e.g. t-statistic or moderated t-statistic for binary outcome, F-statistic for multi-class outcome, regression or correlation coefficient for continuous outcome and log-rank statistic for survival outcome). Additionally, a minimum multi-class correlation (min-MCC) has been included for multi-class outcome to only capture concordant expression patterns that F-statistic often fails (Lu, et al., 2010); (2) One-sided test correction: When combining two-sided p-values for binary outcomes, DE genes with discordant DE direction may be identified and the results are difficult to interpret (e.g. up-regulation in one study but down-regulation in another study). One-sided test correction is helpful to guarantee identification of DE genes with concordant DE direction. For ex-ample, Pearson's correction has been proposed for Fisher's method (Owen, 2009). In addition to the choices above, MetaDE also provides options for gene matching across studies and gene filtering before meta-analysis. Outputs of the meta-analysis results include DE gene lists with corresponding raw p-value, q-values and various visualization tools. Heatmaps can be plotted across studies.

The MetaPath package provides a main function,`"MAPE",`for implementing three meta-analysis framework for pathway enrichment analysis: MAPE_G, MAPE_P and MAPE_I [17]. In the original paper, meta-analyses for pathway enrichment integrated at the gene level (MAPE_G) and integrated at the pathway level (MAPE_P) were investigated. For MAPE_G, information across studies was combined at the gene level and then pathway enrichment analysis was applied. Conversely, for MAPE_P, pathway analysis was first performed in each study independently. The information across studies was then combined at the pathway level. In the simulation analyses and applications, MAPE_G and MAPE_P had complementary advantages and disadvantages under different scenarios and data structure. A hybrid framework, namely MAPE_I, was proposed to integrate advantages of both MAPE_G and MAPE_P. Similar to MetaDE, MetaPath also provides multiple options of gene matching, gene filtering, meta-analysis methods and test statistics to associate with different outcomes. The MetaPath package also provides functions to draw the heatmap of q-values of pathways and a Venn diagram to show the overlapped pathways identified by three MAPE methods.

The purpose of the present document is to provide a general overview of these three packages and their current capabilities. Not all of the possibilities and options are described, as this would require a much longer treatment. The primary package documentation in the form of standard help files can be viewed in R. The article is therefore a starting point for those interested in exploring the possibility of conducting meta-analyses in R with these three packages. All three packages have been uploaded to CRAN repository with standard documents and help files. They constantly maintained and updated to

Table 1: the list of test statistics in individual analysis and methods of meta-analysis can be implemented in MetaDE package

Outcome Variable		binary	multi-class	continuous	suvival
Test statistics		paired t-statistics unpaired t-statistics moderate t-statistics	F-statistics	Pearson correlation Spearman correlation	log-rank statistics
Combine p-values	Fisher (_OC)	✔	✔	✔	✔
	Stouffer (_OC)	✔	✔	✔	✔
	AW (_OC)	✔	✔	✔	✔
	minP (_OC)	✔	✔	✔	✔
	maxP (_OC)	✔	✔	✔	✔
	roP (_OC)	✔	✔	✔	✔
	SR	✔	✔	✔	✔
	PR	✔	✔	✔	✔
	minMCC		✔		
Combine effect sizes	FEM	✔	×	×	×
	REM	✔	×	×	×
combine ranks	randProd	✔	×	×	×

✔: the method can be applied on the corresponding type of outcome.

×: The method cannot be applied on the corresponding type of outcome.

_OC: The corresponding one-sided correction method can be implemented in MetaDE.

incorporate new methods and functionalities.

2 Citing MetaQC, MetaDE and MetaPath

If you use MetaQC, MetaDE or MetaPath and publish your analysis, please report the version of the program used and cite this paper [1]. If appropriate, you may also cite individual methodological paper associated with each package:

- MetaQC:

  Kangwan D. Don and George C. Tseng. MetaQC: objective quality control and inclusion/exclusion criteria for genomic meta-analysis. Nucleic Acids Research , 40, e15, 2012.

- MetaDE:

  Jia Li and George C. Tseng. An adaptively weighted statistic for detecting differential gene expression when combining multiple transcriptomic studies.
  Annals of Applied Statistics. 5:994-1019, 2012.

  Shuya Lu, Jia Li, Chi Song, Kui Shen and George C Tseng. Biomarker Detection in the Integration of Multiple Multi-class Genomic Studies.
  Bioinformatics. 26:333-340, 2010

  Xingbin Wang, Yan Lin, Chi Song, Etienne Sibille and George C Tseng.
  Detecting disease-associated genes with confounding variable adjustment and the impact on genomic meta-analysis: with application to major depressive disorder.
  BMC Bioinformatics. 13:52,2012.

  George C. Tseng, Debashis Ghosh and Eleanor Feingold. (2012) Comprehensive literature

review and statistical considerations for microarray meta-analysis.
Nucleic Acids Research accepted.

- MetaPath:

Kui Shen and George C Tseng. Meta-analysis for pathway enrichment analysis when combining multiple microarray studies.
Bioinformatics. 26:1316-1323,2010

3 Importing data into R

The most difficult aspect of learning to use a new package is importing your data. Once you have mastered this step, you can experiment with other commands. In the following sections, we describe how to prepare the data in Excel and import them into R.

3.1 Preparing data in Excel

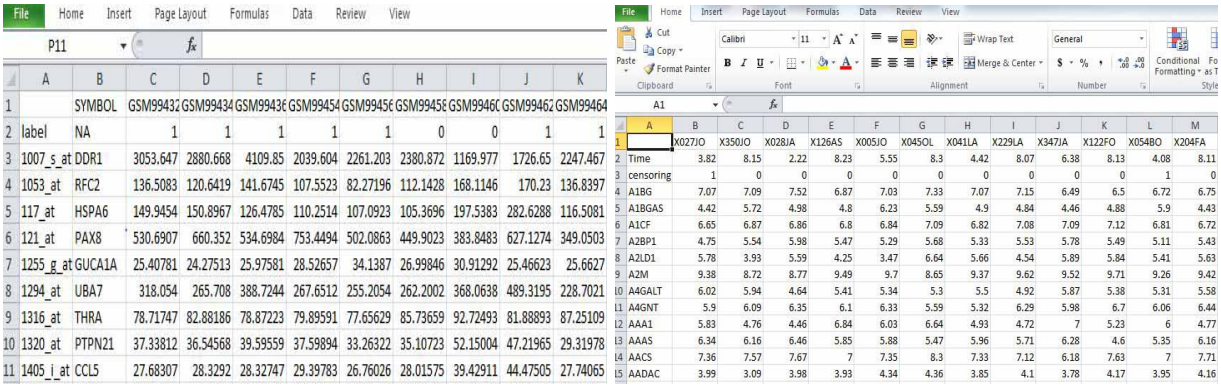
Microarray data sets are generally comprised of three components: (1) the gene expression data; (2) the outcome variable, such as disease status; and (3) patient-specific covariates, including treatment history and additional clinical and demographic information. The primary aim of many gene expression studies is to identify the DE genes by characterizing the relationship between the first two of these components, the gene expression and the disease outcome. Thus, we only consider these two components. The data should be arranged in a gene-by-sample format. That is, the columns represent genes and the rows represent samples. We accept two types of format: unmatched data and matched data as shown in Figure 2. If the probeIDs have not been summarized into unique gene symbols like in Figure 2(a)(i.e. multiple probe IDs may match to the same gene symbol.) , the first column has the probeIDs and the second column has the corresponding gene symbols, and the remaining columns have the expression data matrix. If the gene symbols already serve as a unique ID, the first column can show gene symbols and the expression data starts from the second column (Figure 2(b)). The second row has the outcome variable which should parallel to the corresponding samples like in Figure 2(a). Similarly, for survival data, the second row has the survival time and the third row has the censoring status like in 2(b). For a binary outcome, 0 refers to "normal" and 1 to "diseased". For a multiple class outcome, the first level being coded as 0, the second as 1, and so on. For a survival outcome, 0 refers to individual who was censored while 1 is used for patients who develop the event of interest. Then, you can save the data to tab-delimited ASCII file or comma-delimeted file named by "XX.txt" or "XX.csv".

3.2 Reading data into R

Once the data sets have been prepared and saved in a file directory. We provided a function, `MetaDE.Read`, in MetaDE package, which can read the data sets into R and transform them to the format required for MetaQC, MetaDE and MetaPath package. The arguments of this function are

```
MetaDE.Read(filenames, via = c("txt", "csv"), skip,matched=FALSE, log = TRUE)
```

where **filenmaes** is a vector of character strings specifying the names of data sets to read data from. **via** is a character to indicate the type of the data sets. "txt" means tab-delimited file and "csv" comma-delimited file. **skip** is a numeric vector consist of 1 or 2 , in which 1 means that gene expression data



	A	B	C	D	E	F	G	H	I	J	K
1		SYMBOL	GSM99432	GSM99434	GSM99436	GSM99454	GSM99456	GSM99458	GSM99460	GSM99462	GSM99464
2	label	NA	1	1	1	1	1	0	0	1	1
3	1007_s_at	DDR1	3053.647	2880.668	4109.85	2039.604	2261.203	2380.872	1169.977	1726.65	2247.467
4	1053_at	RFC2	136.5083	120.6419	141.6745	107.5523	82.27196	112.1428	168.1146	170.23	136.8397
5	117_at	HSPA6	149.9454	150.8967	126.4785	110.2514	107.0923	105.3696	197.5383	282.6288	116.5081
6	121_at	PAX8	530.6907	660.352	534.6984	753.4494	502.0863	449.9023	383.8483	627.1274	349.0503
7	1255_g_at	GUCA1A	25.40781	24.27513	25.97581	28.52657	34.1387	26.99846	30.91292	25.46623	25.6627
8	1294_at	UBA7	318.054	265.708	388.7244	267.6512	255.2054	262.2002	368.0638	489.3195	228.7021
9	1316_at	THRA	78.71747	82.88186	78.87223	79.89591	77.65629	85.73659	92.72493	81.88893	87.25109
10	1320_at	PTPN21	37.33812	36.54568	39.59559	37.59894	33.26322	35.10723	52.15004	47.21965	29.31978
11	1405_i_at	CCL5	27.68307	28.3292	28.32747	29.39783	26.76026	28.01575	39.42911	44.47505	27.74065

(a) Data un-matched

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		X027JO	X350JO	X028JA	X126AS	X005JO	X045OL	X041LA	X229LA	X347JA	X122FO	X054BO	X204FA
2	Time	3.82	8.15	2.22	8.23	5.55	8.3	4.42	8.07	6.38	8.13	4.08	8.11
3	censoring	1	0	0	0	0	0	0	0	0	0	1	0
4	A1BG	7.07	7.09	7.52	6.87	7.03	7.33	7.07	7.15	6.49	6.5	6.72	6.75
5	A1BGAS	4.42	5.72	4.98	4.8	6.23	5.59	4.9	4.84	4.46	4.88	5.9	4.43
6	A1CF	6.65	6.87	6.86	6.8	6.84	7.09	6.82	7.08	7.09	7.12	6.81	6.72
7	A2BP1	4.75	5.54	5.98	5.47	5.29	5.68	5.33	5.53	5.78	5.49	5.11	5.43
8	A2LD1	5.78	3.93	5.59	4.25	3.47	6.64	5.66	4.54	5.89	5.84	5.41	5.63
9	A2M	9.38	8.72	8.77	9.49	9.7	8.65	9.37	9.62	9.52	9.71	9.26	9.42
10	A4GALT	6.02	5.94	4.64	5.41	5.34	5.3	5.5	4.92	5.87	5.38	5.31	5.58
11	A4GNT	5.9	6.09	6.35	6.1	6.33	5.59	5.32	6.29	5.98	6.7	6.06	6.44
12	AAA1	5.83	4.76	4.46	6.84	6.03	6.64	4.93	4.72	7	5.23	6	4.77
13	AAAS	6.34	6.16	6.46	5.85	5.88	5.47	5.96	5.71	6.28	4.6	5.35	6.16
14	AACS	7.36	7.57	7.67	7	7.35	8.3	7.33	7.12	6.18	7.63	7	7.71
15	AADAC	3.99	3.09	3.98	3.93	4.34	4.36	3.85	4.1	3.78	4.17	3.95	4.16

(b) Data matched

Figure 2: Example of the organization of a dataset in Excel prior to importing into R.

starts from the 2nd row and 2 means we should skip 2 rows to read in the gene expression data. If the ith data set is survival data, the corresponding element of **skip** should be 2 otherwise 1. **matched** is a logical value to specify whether probeIDs have been matched into gene symbols or not. **log** is a logical value to specify whether data sets need to be log2-transformed. The following is an example of the usage of this function. We have nine datasets that studied the gene expressions between prostate cancer and normal samples. The probeIDs were already annotated and matched to a unique gene symbol. So we saved them in the Figure 2(b) format as tab delimited files with the first row representing outcome labels. When we read the data, we only need to specify that the data is matched, and first row is sample labels for all 9 studies.

```
> library(MetaDE)
> study.names<-c("Welsh","Yu","Lapointe","Varambally","Singh","Wallace","Nanni",
"Dhanasekaran","Tomlins")
> prostate.raw<-MetaDE.Read(study.names,skip=rep(1,9),via="txt",matched=T,log=F)
```

Tip: When the size of the data sets is too large, it may take a while to read them into R. Another effective way is that you can read them separately into R and then save them as "xxx.Rdata" using `save` function and import them into R using `load` function:

```
> load("Dhanasekaran.rdata")
> load("Lapointe.rdata")
> load("Nanni.rdata")
> load("Singh.rdata")
> load("Tomlins.rdata")
> load("Varambally.rdata")
> load("Wallace.rdata")
> load("Welsh.rdata")
> load("Yu.rdata")
> prostate.raw<-list(
+ Welsh=Welsh,
+ Yu=Yu,
+ Lapointe=Lapointe,
+ Varambally=Varambally,
```

```
+ Singh=Singh,
+ Wallace=Wallace,
+ Nanni=Nanni,
+ Dhanasekaran=Dhanasekaran,
+ Tomlins=Tomlins
+ )
```

4 Data preprocessing

4.1 Gene matching

Usually different microarray platforms use their own probe IDs. To perform metan-analysis, we need to match probe IDs from different platforms to the unique official gene ID, such as ENTREZ ID or gene symbol. In this package, we focus on the gene symbol. In MetaDE package, we provide two options for the summarization when multiple probes (or probe sets) matched to an identical gene symbol: one option is to take the average value of expression values across multiple probe IDs to represent the corresponded gene symbol; another one is the "IQR" method in which we selected the probe ID with the largest interquartile range (IQR) of expression values among all multiple probe IDs to represent the gene. The procedure of gene matching can be implemented by function `MetaDE.match`. Although "average" method has been widely used due to its simplicity, "IQR" is biologically more reasonable and robust and is highly recommended (e.g. see page 225 in Bioconductor Case study[9]). The arguments of this function are

```
MetaDE.match(x,pool.replicate=c("average","IQR"))
```

where **x** is a list of datasets. Each data set is a list with components, x-the gene expression matrix,y- the outcome, and censoring.status (only for survival data). The arguments for **pool.replicate** are then:

- "**average**": the average method mentioned as above was chosen to perform gene matching;
- "**IQR**": the "IQR" method mentioned as above was chosen to perform gene matching;

4.2 Gene merging

The multiple gene expression data sets may not be very well aligned by genes, and the number of genes in each study maybe different. The `MetaDE.merge` function is used to extract the common genes across multiple studies so that the merged data sets have the same genes in the same order. When we combine a large number of studies, the number of common genes may be very small, so we allow to include some gene appearing in most studies and missing in few studies. For example, if you set the argument **MVperc** is 0.2, the genes appearing in 80% studies and missing in 20% studies will be included for the analysis. The default is zero which means that we only include genes appearing in all the studies.

```
> prostate.merged<-MetaDE.merge(prostate.raw)
> dim(prostate.merged[[1]][[1]])
> [1] 1903   34
```

From the output, we see that there are total 1903 common genes among 9 studies.

4.3 Gene filtering

Biologically, it is likely that most genes are either un-expressed or un-informative. In gene expression analysis to find DE genes, these genes contribute to the false discoveries, so it is desirable to filter out these genes prior to analysis. After genes are matched across studies, the unique gene symbols are available across all studies. Two sequential steps of gene filtering can be performed. In the first step, we filter out genes with very low gene expression that are identified with small average expression values across majority of studies. Specifically, mean intensities of each gene across all samples in each study are calculated and the corresponding ranks are obtained. The sum of such ranks across all studies of each gene is calculated and genes with the smallest $\alpha\%$ rank sum (small mean intensity) are considered un-expressed genes (i.e. small expression intensities) and were filtered out. Similarly, in the second step, we filter out non-informative (small variation) genes by replacing mean intensity in the first step with standard deviation. Genes with the lowest $\beta\%$ rank sum of standard deviations were filtered out. Finally, the total number of matched genes is $G \times (1-\alpha) \times (1-\beta)$, which are used for further analysis. The procedure of gene filtering can be implemented by function `MetaDE.filter`. The arguments of this function are

```
MetaDE.filter(x,DelPerc=c(alpha,beta)),
```

where **x** is a list of data sets described as before; argument **DelPerc** is a numeric vector of length 2, which specify how many percent of genes need to be filtered out during the two sequential steps of gene filtering.

```
> prostate.filtered<-MetaDE.filter(prostate.merged,c(0.3,0.3))
> dim(prostate.filtered[[1]][[1]])
> [1] 932   34
```

Here, we first filtered out 30% un-expressed genes and then 30% non-informative genes. Finally, $932 = 1903 \times (1-0.3) \times (1-0.3)$ genes were remained for further analysis.

5 The MetaQC package

The MetaQC package provides two main functions, `MetaQC` and `runQC` to implement the objective quality control and inclusion/exclusion criteria for genomic meta-Analysis.

5.1 The MetaQC

The `MetaQC` function is used to implement the six quantitative quality control measures. For the default interface, the arguments of the function are

```
MetaQC(DList, GList, isParallel = FALSE, nCores = NULL,
    useCache = TRUE, filterGenes = TRUE,
    maxNApctAllowed=.3, cutRatioByMean=.4, cutRatioByVar=.4, minNumGenes=5,
    verbose = FALSE, resp.type = c("Twoclass", "Multiclass", "Survival"))
```

where

- **DList**: Either a list of all data matrices (Case 1) or a list of lists (Case 2); The first case is simplified input data structure only for two classes comparison. Each data name should be set as the name of

each list element. Each data should be a numeric matrix that has genes in the rows and samples in the columns. Row names should be official gene symbols and column names be class labels. For the full description of input data, you can use the second data format. Each data is represented as a list which should have x, y, and geneid (geneid can be replaced to row names of matrix x) elements, representing expression data, outcome or class labels, and gene ids, respectively. Additionally, in the survival analysis, censoring.status should be set.

- **GList**: The location of a file which has sets of gene symbol lists such as gmt files. By default, the gmt file will be converted to list object and saved with the same name with ".rda". Alternatively, a list of gene sets is allowed; the name of each element of the list should be set as a unique pathway name, and each pathway should have a character vector of gene symbols.
- **isParallel**: Whether to use multiple cores in parallel for fast computing. By default, it is false.
- **nCores**: When isParallel is true, the number of cores can be set. By default, all cores in the machine are used in the unix-like machine, and 2 cores are used in windows.
- **useCache**: Whether imported gmt file should be saved for the next use. By default, it is true.
- **filterGenes**: Whether to use gene filtering (recommended).
- **maxNApctAllowed**: Filtering out genes which have missing values more than specified ratio (Default .3). Applied if filterGenes is TRUE.
- **cutRatioByMean**: Filtering out specified ratio of genes which have least expression value (Default .4). Applied if filterGenes is TRUE.
- **cutRatioByVar**: Filtering out specified ratio of genes which have least sample wise expression variance (Default .4). Applied if filterGenes is TRUE.
- **minNumGenes**: Mininum number of genes in a pathway. A pathway which has members smaller than the specified value will be removed. verbose Whether to print out logs.
- **resp.type**: The type of response variable. Three options are: "Twoclass" (unpaired), "Multiclass", "Survival." By default, Twoclass is used

First you can create an QC object with the following code:

```
> Data.QC<-list()
> for(i in 1:9){
+ colnames(prostate.filtered[[i]][[1]])<-prostate.filtered[[i]][[2]]
+ Data.QC[[i]]<-impute.knn(prostate.filtered[[i]][[1]])$data
+ }
> names(Data.QC)<-names(prostate.filtered)
> ProstateQC <- MetaQC(Data.QC, "c2.all.v3.0.symbols.gmt", filterGenes=F,verbose=TRUE,
+ isParallel=F,resp.type="Twoclass")
```

5.2 The runQC function

The `runQC` function is a utility function to RunQC method in MetaQC object. The usage and arguments are listed below

```
runQC(QC, nPath=NULL, B=1e4, pvalCut=.05,
    pvalAdjust=FALSE, fileForCQCp="c2.all.v3.0.symbols.gmt")
```

- **QC**: A proto R object which obtained by MetaQC function.
- **nPath**: The number of top pathways which would be used for EQC calculation. The top pathways are automatically determined by their mean rank of over significance among given studies. It is important that gene sets used for EQC are expected to have higher correlation than background. For better performance, this should be set as a reasonably small number.
- **B**: The number of permutation tests used for EQC calculation. More than 1e4 is recommended.
- **pvalCut**: P-value threshold used for AQC calculation.
- **pvalAdjust**: Whether to apply p-value adjustment due to multiple testing (B-H procedure is used).
- **fileForCQCp**: Gene set used for CQCp calculation. Usually larger gene set is used than EQC calculation.

Then, the users can run QC procedure with

```
> runQC(ProstateQC, B=1e4, fileForCQCp="c2.all.v3.0.symbols.gmt")
```

5.3 Summary output and visualization in MetaQC

The users can use the `print` function to view the information of data sets and the table of the quantitative quality control measures.

```
>  print(ProstateQC)
Number of Studies:  9

Dimension of Each Study:
        Welsh  Yu Lapointe Varambally Singh Wallace Nanni Dhanasekaran Tomlins
Genes     932 932      932        932   932     932   932          932     932
Samples    34 146      103         13   102      89    30           28      66

         Study   IQC   EQC   CQCg  CQCp  AQCg  AQCp Rank
1           Yu  8.83  3.82 307.65 32.81   3.3 16.37 1.75
2        Welsh  5.19  2.28 307.65  43.6  6.17 13.42 2.42
3     Lapointe  4.47  3.22  16.41  37.8 2.09* 21.36 2.83
4   Varambally  4.89  2.49   5.28 10.95 1.18*  2.76 4.67
5        Singh  3.39 1.84*   8.38  10.2 2.04* 10.71 5.17
6      Wallace  6.32 1.85*  0.01* 18.74 0.45* 0.23* 6.00
7        Nanni 1.79*   3.6  0.22* 1.25* 0.24* 2.25* 6.50
8      Tomlins 1.55* 1.34*  0.02* 1.95* 0.68* 0.19* 7.67
9 Dhanasekaran 0.01* 1.12*  0.04*  8.87 0.45* 0.03* 8.00
```

The users can also draw the PCA biplot (see Figure 3) for assisting visualization and decision with

```
plot(ProstateQC)
```

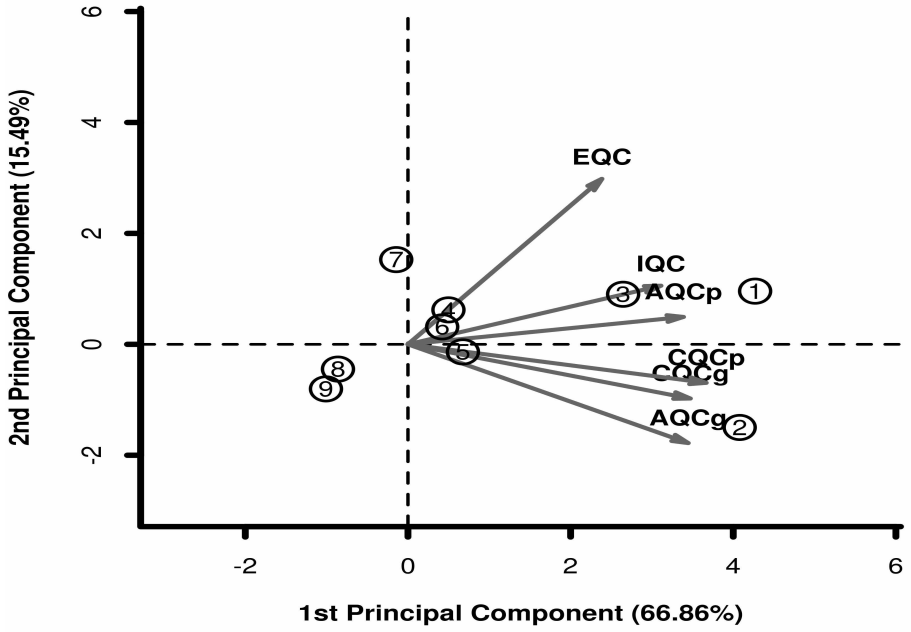


Figure 3: PCA biplot of QC measures in nine prostate studies.

From above table and the PCA biplot(Figure3), although the first two PCs also captured high percentage of variance (83%), the studies were more scattered in the biplot and even good performing studies had quite different performance when judged by different QC criteria. For example, Varambally and Wallace had better scores in IQC and EQC but not in CQC and AQC while Welsh, Lapointe and Singh, had better performance in CQC and AQC but not IQC and EQC. Yu had performed the best in all criteria. In considering sample size, array platform and QC measures, we regarded the bottom three studies (Nanni, Tomlins and Dhanasekaran) as definite exclusion cases and remove them from further analysis in MetaDE and MetaPath.

Based on the results of MetaQC, three studies,"Nanni","Dhanasekaran" and "Tomlins" need to be excluded for further analysis. To remove impact on the results of meta-analysis of these three studies, we need re-merge the studies and re-filter genes:

```
> study.names<-c("Welsh","Yu","Lapointe","Varambally","Singh","Wallace")
> data.QC.raw<-list()
> for(i in 1:length(study.names)){
+ data.QC.raw[[i]]<-prostate.raw[[study.names[[i]]]] + }
> data.QC.merged<-MetaDE.merge(data.QC.raw)
> dim(data.QC.merged[[1]][[1]])
[1] 6940   34
> data.QC.filtered<-MetaDE.filter(data.QC.merged,c(0.2,0.2))
> dim(data.QC.filtered[[1]][[1]])
>[1] 4441   34
```

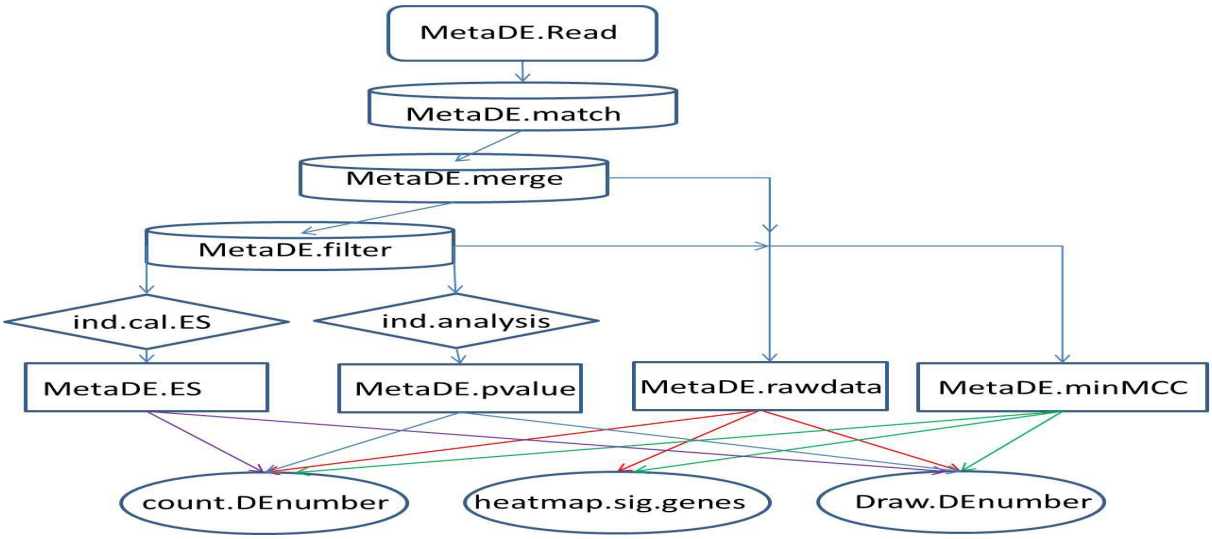


Figure 4: the flowchart presents a brief overview of the main functions for conducting meta-analysis.

We saw that there are total 6940 common genes across these six studies. Then, we first filtered out 20% un-expressed genes and then 20% non-informative genes. Finally,$4441 = 6940 \times (1-0.2) \times (1-0.2)$ genes were remained for further analysis.

6 The MetaDE package

In current version, MetaDE package provides functions for conducting 12 major meta-analysis methods for differential expression analysis (see Table 1). In Figure 4, the flowchart presents a brief overview of the main functions implementing these methods.

6.1 Perform analysis for individual study

Before beginning with a meta-analysis, one must first obtain a set of p-values or effect size estimates with their corresponding sampling variances. The MeteDE package provides the `ind.analysis()` function, which can be used to perform various test statistics for DE analysis based on the type of the outcome and choice of p-value calculation by either fast parametric or robust permutation inferences. For the default interface, the arguments of the function are

```
ind.analysis(Dlist, ind.method = c("regt", "modt", "pairedt",
"pearsonr", "spearmanr",
 "F", "logrank"), miss.tol =0.3, nperm = NULL, tail, ...)
```

where **Dlist** is the input variable,which is a list of datasets and each data set is a list with components: **x**- the gene expression matrix; **y**- the outcome variable (see 3.1),for survival data, this is the survival time of patients; **censoring.status**- the censoring status. argument **ind.mehtod**is a character vector specifying which test statistic should be used to calculated the p-values in each study.The options for argument **ind.method** are then:

- **regt**: The regular t-statistics.
- **modt**: The moderated-t statstics.

- **pairedt**: The paired t-statistics.
- **pearsonr**: The Pearson's product correlation statistics.
- **F**: The F-statistics from one way anova.
- **spearmanr**: The Spearman's rank correlation statistics.
- **logrank**: The log-rank statistics.

nperm is an argument to specify the choice of p-value calculation by fast parametric or robust permutation inferences. If it is `NULL`(default), the parametric method is used; If it is an integer, the permutation method is used, and the integer is the number of permutations used to infer the p-values. **tail** is a character string specifying the direction of alternative hypothesis , must be one of "low"(left-side p-value), "high"(right-sided p-value) or "abs"(two-sided p-value). The users can choose the appropriate test statistics based on the type of outcome in their data sets as described in Table 1. For example, if your studies are pair-designed, you may chose "pairedt" as the ind.methos.

```
> ind.Res1<-ind.analysis(data.QC.filtered,ind.method=rep("modt",6),nperm=300,tail="abs")
Cluster size 4436 broken into 2416 2020
Cluster size 2416 broken into 510 1906
Done cluster 510
Cluster size 1906 broken into 1070 836
Done cluster 1070
Done cluster 836
Done cluster 1906
Done cluster 2416
Cluster size 2020 broken into 1533 487
Cluster size 1533 broken into 611 922
Done cluster 611
Done cluster 922
Done cluster 1533
Done cluster 487
Done cluster 2020
gene: GRB10 SCARB1 FAM179B OLFML2A CUL9 will not be analyzed due to > 0.3 missing
dataset 1 is done
dataset 2 is done
dataset 3 is done
dataset 4 is done
dataset 5 is done
dataset 6 is done
```

The output of the `"ind.analysis"`function is a list with components: stat–the value of test statistic for each gene; p– the p-value for the test for each gene; bp the p-value from nperm permutations for each gene. The bp values from the output will be used for the meta analysis. But it can be NULL if you chose asymptotic results. We can look at the results with:

```
> head(ind.Res1$stat)
          Welsh          Yu   Lapointe Varambally        Singh    Wallace
```

```
KLK3  -2.7788918 -2.42749548 -0.7689485 -0.9014151 -3.085452640  1.4452094
ACPP  -0.8785095  0.05370996  1.0796471  0.5908097 -1.583299947  2.4004244
KLK2  -2.9347306 -2.43864422 -0.3074178 -0.6470538 -3.050150236 -1.8343092
ACTA2  2.2738735  3.13297708  5.0134537  0.3519709 -0.379635185  0.9729776
MSMB   0.3767477  2.18256555  1.2681645  0.6959927 -1.467310235  2.8007777
TAGLN  2.4506658  2.90666452  4.1497288  0.6672679 -0.005584158  0.9778782
> head(ind.Res1$p)
              Welsh           Yu     Lapointe Varambally        Singh      Wallace
KLK3  0.0001726338 7.655933e-05 1.298114e-01  0.1019530 7.505817e-07 1.040832e-02
ACPP  0.1183344592 9.137469e-01 3.873084e-02  0.2652458 6.951888e-03 1.666291e-04
KLK2  0.0000900698 7.430759e-05 5.338330e-01  0.2250124 7.505817e-07 2.090370e-03
ACTA2 0.0010553179 1.000000e-20 1.000000e-20  0.4974007 4.370510e-01 6.469114e-02
MSMB  0.4604653607 2.852210e-04 1.723925e-02  0.1944442 1.107183e-02 3.752909e-05
TAGLN 0.0005786985 5.254072e-06 1.000000e-20  0.2119770 9.906125e-01 6.352323e-02
```

The MetaDE package also provides a function `ind.cal.ES` to calculate various effect sizes (and the corresponding sampling variances) that are commonly used in meta-analyses.The arguments for this interface are

```
ind.cal.ES(x, paired, nperm = NULL)
```

where arguments **y** and **l** are the gene expression matrix and the vector of labels of outcome, respectively; **paired** is a vector of logical values to specify whether the corresponding study is paired design or not. If the study is pair-designed, the effect sizes (corresponding variances) are calculated using the formula in morris's paper[18], otherwise calculated using the formulas in choi et al[3]. Argument **nperm** is an integer to specify the number of permutations.If it is not "NULL", the permutated effect sizes and corresponding variances will be calculated.

```
> ind.Res2<-ind.cal.ES(data.QC.filtered,paired=rep(F,6),nperm=300,miss.tol=0.3)
Cluster size 4436 broken into 2416 2020
Cluster size 2416 broken into 510 1906
Done cluster 510
Cluster size 1906 broken into 1070 836
Done cluster 1070
Done cluster 836
Done cluster 1906
Done cluster 2416
Cluster size 2020 broken into 1533 487
Cluster size 1533 broken into 611 922
Done cluster 611
Done cluster 922
Done cluster 1533
Done cluster 487
Done cluster 2020
gene: GRB10 SCARB1 FAM179B OLFML2A CUL9 will not be analyzed due to > 0.3 missing
> head(ind.Res2$ES)
           Welsh          Yu    Lapointe Varambally       Singh    Wallace
```

```
KLK3   1.5848901  0.60182438  0.21573880  0.9646776 0.704811088 -0.6095348
ACPP   0.5985059 -0.01214459 -0.34942015 -0.5815326 0.361049284 -1.2022999
KLK2   1.9979439  0.54617644  0.08703751  0.6880705 0.690029042  0.6630656
ACTA2 -1.3935275 -0.83448418 -1.99178219 -0.4761814 0.093401267 -0.5050199
MSMB  -0.1853324 -0.46117042 -0.40327690 -0.6051680 0.331568663 -1.0163406
TAGLN -1.4227497 -0.83143018 -1.51385329 -0.6631125 0.001375977 -0.4906553
> head(ind.Res2$Var)
          Welsh         Yu   Lapointe Varambally      Singh    Wallace
KLK3  0.1880505 0.02897068 0.04074521  0.3453162 0.04166586 0.06658002
ACPP  0.1563789 0.02773080 0.04111197  0.3225307 0.03986977 0.07261368
KLK2  0.2098138 0.02875190 0.04055605  0.3277331 0.04156479 0.06696273
ACTA2 0.1796687 0.03011510 0.05977751  0.3182449 0.03927353 0.06592559
MSMB  0.1516162 0.02845864 0.04130875  0.3236095 0.03976968 0.07029583
TAGLN 0.1808790 0.03009768 0.05164428  0.3264360 0.03923078 0.06584524
```

6.2 Perform meta-analysis

The various meta-analyses can be implemented by three main functions, `MetaDE.radata()`, `MetaDE.pvalue()` and `MetaDE.ES()`, in MetaDE package. The arguments of function
`MetaDE.rawdata()` are given by

```
MetaDE.rawdata(x, ind.method = c("modt", "regt", "pairedt", "F",
                 "pearsonr", "spearmanr", "logrank"), meta.method =
                 c("maxP", "maxP.OC", "minP", "minP.OC", "Fisher",
                 "Fisher.OC", "AW", "AW.OC", "roP", "roP.OC",
                 "Stouffer", "Stouffer.OC", "SR", "PR", "minMCC",
                 "FEM", "REM", "rankProd"), paired = NULL, miss.tol =
                 0.3, rth = NULL, nperm = NULL, ind.tail = "abs",
                 asymptotic = FALSE, ...)
```

As above,**x** is the raw data (the gene expression matrices and the labels of outcome),which is a list of a list datasets and a list of labels; the argument **ind.method** is the same as that in function `ind.analysis()`; The various meta-analysis methods described in Table 1 that can be specified via the **meta.method** argument are then:

- **maxP**: The maximum p-value method;
- **maxP.OC**: The maximum p-value with one-sided correction;
- **minP**: The minimum p-value method;
- **minP.OC**: The minimum p-value method with one-sided correction;
- **Fisher** The Fisher's method;
- **Fisher.OC**: The Fisher's method with one-sided correction;
- **AW**: The adaptive weight method;
- **AW.OC**: The adaptive weight method with one-sided correction;

- **roP**: The r-th ordered p-value method;
- **roP.OC**: The r-th ordered p-value method with one-sided correction;
- **Stuoffer**: The Stuoffer's method;
- **Stouffer.OC**: The Stuoffer's method with one-side correction;
- **minMCC**: The the minimum multi-class correlation method [15];
- **rankProd**: The rank product method [14];
- **SR**:The naive rank summation method[6];
- **PR**:The naive rank product method[6];
- **FEM**: The fixed-effect model method [3];
- **REM**: The random-effect model method [3];

If the **meta.method** is chosen as "roP" or "roP.OC", an integer need input via argument **rth** to specify which rth ordered p-value as the statistic; If the argument **asymptotic** is `TRUE`, then the parametric method is used in meta-analysis to calculate the p-values permutation should be used otherwise; the argument **nperm** is the same as in function `ind.analysis()`.

If the raw data sets are available, all the meta-analysis mentioned in Table 1 can be implemented with this function. This function offers much wider options of analysis methods for both individual dataset analysis and meta-analysis. It is suitable to researchers who want to obtain an analysis easily and tailor their choices to the biological questions of interest. For example, if one is interested in finding genes that are differentially expressed between cases and controls in all datasets. One could select "moderated t-test" from the individual analysis and select "maxP" from the meta-analysis to combine the p-values from moderated t-test. The researchers may also want to make a comparison among different meta-analysis methods. For example, the users want to make a comparison among four meta-analysis methods, "Fisher", "maxP", "roP", and "AW". This goal can be done with:

```
> MetaDE.Res1<-MetaDE.rawdata(data.QC.filtered,ind.method=rep("modt",6),
meta.method=c("Fisher","maxP","roP","AW"),rth=4,nperm=300,asymptotic=F)
```

If p-values or effect sizes (and corresponding variances) have been calculated already, for example by other methods not used in functions `ind.analysis()` or `ind.cal.ES()` with the help of other software, then the meta-analysis can be implemented by function `MetaDE.pvalue()` or `MetaDE.ES()` to combine p-values and effect sizes across studies respectively. The arguments of these two functions are given by

```
MetaDE.pvalue(x, meta.method = c("maxP", "maxP.OC", "minP",
                  "minP.OC", "Fisher", "Fisher.OC", "AW", "AW.OC",
                  "roP", "roP.OC", "Stouffer", "Stouffer.OC", "SR",
                  "PR"), rth = NULL, miss.tol = 0.3, asymptotic = FALSE)
```

, where argument **x** is a list with components:**p**–a list of p values for each dataset;**bp**– a list of p values calculated from permutation for each dataset. This part can be NULL if you just have the p-values from your own method. If the second object of **bp** is `NULL`, the parametric method is then used in meta-analysis.

```
MetaDE.ES(x, meta.method = c("FEM", "REM"))
```

, where **x** is a list with components; **ES**– the observed effect sizes;**Var**– the observed Variances corresponding to **ES**; **perm.ES**–the effect sizes calculated from permutations; **perm.Var**–the corresponding variances calculated from permutations. When **perm.ES** and **perm.Var** are `"NULL"`,the parametric method is used to calculated the p-values,otherwise permutation method is used. argument **meta.method** is a character to specify whether a fixed- or a random/mixed-effects model should be fitted. In the following, we randomly generated p-values for 10 genes in 10 studies, and illustrated how to combine them using the Fisher's and maxP methods in the `"MetaDE.pvalue"`function.

```
> set.seed(123)
> x<-list()
> x$p<-matrix(runif(10*10),10,10)
> x$bp<-NULL
> res1<-MetaDE.pvalue(x,meta.method=c("Fisher","maxP"))
> head(res1$meta.analysis$pval)
        Fisher      maxP
[1,] 0.3841967 0.6860759
[2,] 0.8459936 0.3576885
[3,] 0.8818869 0.1059398
[4,] 0.1440220 0.9441530
[5,] 0.1479757 0.5412986
[6,] 0.1901586 0.3480009
> set.seed(124)
> x<-list()
> x$ES<-matrix(rnorm(10*10),10,10)
> x$Var<-matrix(rchisq(10*10,5),10,10)
> res2<-MetaDE.ES(x,meta.method="REM")
> head(res2$pval)
[1] 0.5193695 0.8438519 0.6652518 0.9381124 0.4649985 0.9796356
```

6.3 Summary output and visualization in MetaDE

The MetaDE package provides several functions for creating plots that are frequently used in meta-analyses.For example,the `heatmap.sig.genes()` function is used to create the heatmaps plots of the DE genes under a specified p-value or FDR threshhold across studies. Figure 5 is an example showing the identified genes between cases (1) and controls (0) across two studies. The heatmap (Figure 5) can be generated with

```
> label1<-rep(0:1,each=5)
> label2<-rep(0:1,each=5)
> exp1<-cbind(matrix(rnorm(5*200),200,5),matrix(rnorm(5*200,2),200,5))
> exp2<-cbind(matrix(rnorm(5*200),200,5),matrix(rnorm(5*200,1.5),200,5))
> x<-list(list(exp1,label1),list(exp2,label2))
> meta.res2<-MetaDE.rawdata(x=x,ind.method=c("modt","modt"),meta.method=c("Fisher","maxP"),nperm=200)
Please make sure the following is correct:
*You input 2 studies
```

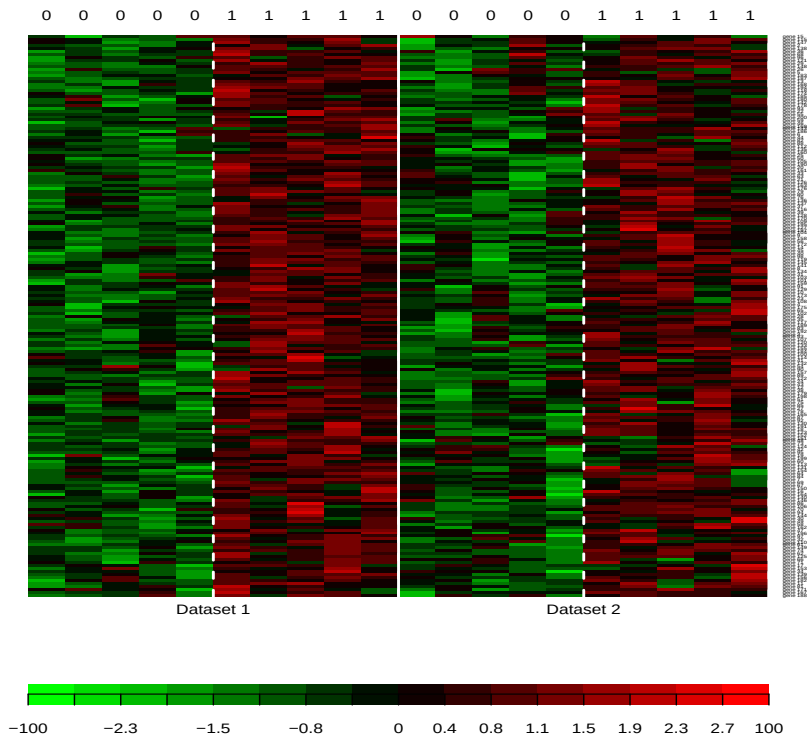


Figure 5: The heatmap plot.

```
*You selected modt modt for your 2 studies respectively
*They are not paired design
* Fisher maxP was chosen to combine the 2 studies,respectively
dataset 1 is done
dataset 2 is done
Permutation was used instead of the asymptotic estimation
> heatmap.sig.genes(meta.res2, meta.method="maxP",fdr.cut=1,color="GR")
```

To assess performance of these different methods, we applied two evaluation criteria. The users may want to compare the numbers of detected DE genes from different methods under different p-value thresholds using detection competency curves (x-axis: p-value or FDR threshold; y-axis: number of detected DE genes)(see Figure 6). This can be implemented with the `draw.DEnumber()` function.

```
> mylty<-rep(c(1,2),c(6,4))
> mycol<-c(rep("black",6),c("red","green","blue","orange"))
> mylwd<-rep(c(1,2),c(6,4))
> mypch<-1:10
> draw.DEnumber(MetaDE.Res1,0.05,mlty=mylty,mcol=mycol,mlwd=mylwd,mpch=mypch,FDR=T)
```

To make a prettier figure, the users can specify the line type, line color, line width and line symbol for each method. For example, here, we set the lines of 6 individual analysis with the same line type, line width and line color ("black") while lines for each meta-analysis method have different line types, widths and colors. It is clearly seen that meta-analysis usually detects more candidate markers, except for maxP (which we know is a very conservative meta-analysis procedure [26]). To view the exact number of DE genes detected by different methods, the function `count.DEnumber()` can be used to generate the

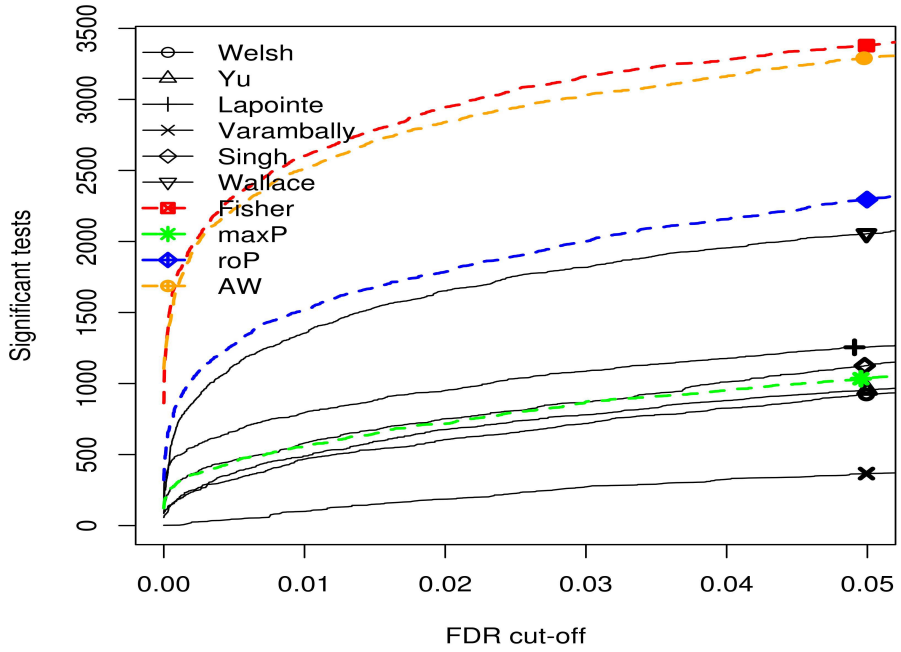


Figure 6: The detection competency curves to compare the DE numbers detected in individual analysis and meta-analysis.

tables in which the numbers of DE genes detected by different methods under various p-value and FDR thresholds are listed.

```
> count.DEnumber(MetaDE.Res1,p.cut=c(0.001,0.005),q.cut=c(0.01,0.05))
$pval.table
        Welsh  Yu Lapointe Varambally Singh Wallace Fisher maxP  roP   AW
p=0.001   461 478      713         199   533    1060   2001  527 1188 1959
p=0.005   724 772      986         396   835    1520   2539  818 1643 2475

$FDR.table
         Welsh  Yu Lapointe Varambally Singh Wallace Fisher maxP  roP   AW
FDR=0.01   466 489       794          99   581    1354   2602  560 1518 2511
FDR=0.05   923 954      1255         366  1125    2054   3379 1034 2295 3290
```

7 The MetaPath package

The MetaPath package provides a major function `MAPE` to implement the Meta-analysis for Pathway Enrichment (MAPE) methods introduced [17]. The function automatically performs MAPE_G (integrating multiple studies at gene level), MAPE_P (integrating multiple studies at pathway level) and MAPE_I (a hybrid method integrating MAEP_G and MAPE_P methods). MAPE_G and MAPE_P have complementary advantages and detection power depending on the data structure. In general, the integrative form of MAPE_I is recommended to use. In the case that MAPE_G (or MAPE_P) detects almost none

pathway, the integrative MAPE_I does not improve performance and MAPE_P (or MAPE_G) should be used.

7.1 The MAPE function

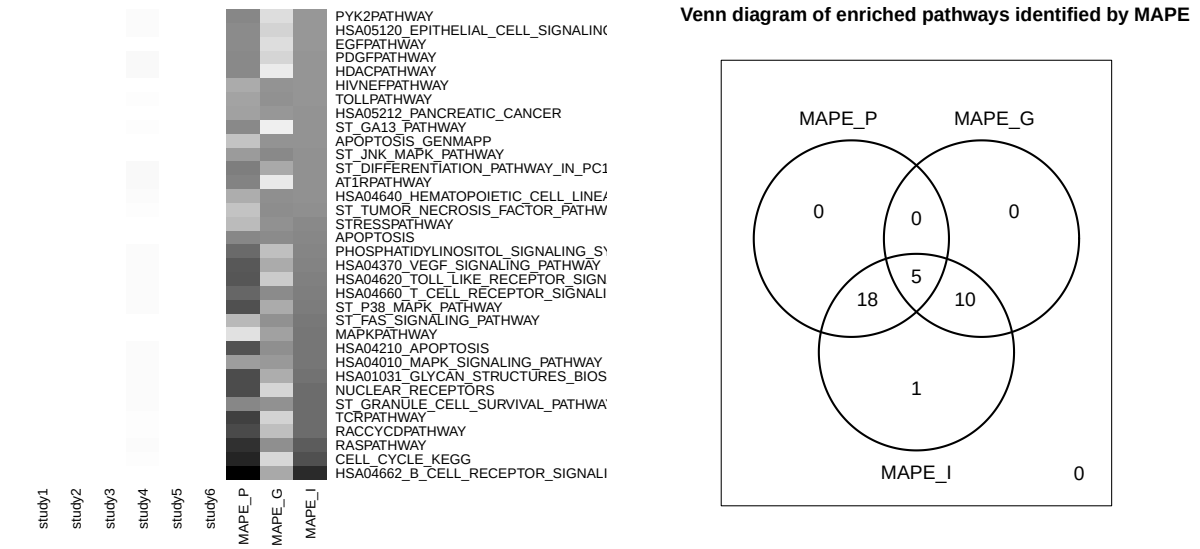
```
MAPE(arraydata, class.label, censoring.status = NULL, DB.matrix, size.min = 15, size.max = 500,
     nperm = 500, stat, rth.value = NULL, resp.type, permutation = "sample")
```

- **arraydata**: The arraydata is a list of microarray data sets. Each microarray data set can be either an eSet or a list. If the microarray data set is a list, then it includes five elements as follows: x–Cexprs data;y–C the phenotype of interests z–C censoring.status if applicable. 1 stands for the event occurred and 0 stands for censored. 4)geneid 5)samplename If the microarray data set is an eSet, the users need to indicate the slots for phenotype of interests and slots for the censoring.status if applicable. (See examples) class.label The slot for the phenotype of interests. It is only applicable when arraydata is an eSet.
- **censoring.status**: The slot for the censoring.status. It is only applicable when arraydata is an eSet.
- **DB.matrix**: The pathway database in a matrix form. Each row is a pathway and each column is a gene. Zeros stands for that the gene does not exist in the pathway and one stands for that the gene exists in the pathway.
- **size.min**: The minimum size of pathways to be considered. The default value is 15.
- **size.max**: The maximum size of pathways to be considered. The default value is 500.
- **nperm**: Number of permutations to be performed.
- **stat**: The meta-analysis statistics to be used to combine two studies. It is one of the four values: 'minP','maxP','rth','Fisher'.
- **rth.value**: The value of the rth statistics if the meta-anlaysis statistic is 'rth'. For example, rth.value=0.6.
- **resp.type**: The type of phenotype. It is one of the three values: "discrete", "continuous", 'survival".
- **permutation**: The options for using sample permutation or gene permutation when performing enrichment analysis. it is one of the two values: 'gene' and 'sample'. The default option is sample permutation.

7.2 Summary output and visualization in MetaDE

The MetaPath package also provides functions to draw the heatmap(see Figure 7(a) of q-values of pathways and a Venn diagram (see Figure 7(b)) to show the overlapped pathways identified by three MAPE methods.

```
> Prostate.data=vector(mode = "list", length =length(data.QC.filtered) )
> for(t1 in 1:length(data.QC.filtered)){
+     temp<-impute.knn(data.QC.filtered[[t1]][[1]])$data
```



(a) Heatmap

(b) Ven diagram

Figure 7: (a) The heatmap of the q-values of pathways detected by MAPE_I under q-value=0.2 threshold. (b) The Ven diagram of the pathways detected by three MAPE methods.

```
+     Prostate.data[[t1]]=list(x=temp, y=data.QC.filtered[[t1]][[2]],
+     geneid=rownames(data.QC.filtered[[1]][[1]]),
+     samplename=paste("s",1:ncol(data.QC.filtered[[t1]][[1]]),sep=""))
+ }
> start<-Sys.time()
> prostate.MAPE<-MAPE(arraydata=Prostate.data,
+    pathway.DB=pathway.DB,resp.type="discrete",
+    stat="Fisher",nperm=300,permutation="sample",size.min=15,size.max=500)
Performing MAPE_P analysis...
Performing MAPE_G analysis...
Performing MAPE_I analysis...
> MetaPath.Time<-Sys.time()-start
> print(MetaPath.Time)
Time difference of 18.46028 mins
> plot.MAPE(prostate.MAPE, cutoff=.2, MAPE.method="MAPE_I")
```

Majority of the detected pathways appeared to be cancer related. Single study analyses showed weak pathway enrichment and detected almost no pathways(see Figure 7(a)). MAPE_P and MAPE_G appeared to have complementary detection power (identified 23 and 15 pathways with only 5 in common). MAPE_I detected the largest number of pathways (34 pathways).

8 Example

In previous sections, we described the usages of the major functions in each of three packages. To demonstrate overall application of MetaQC, MetaDE and MetaPath, we collected nine prostate cancer

Table 2: Summary information of nine prostate studies.

Author	Year	Platform	Sample size(Normal/Primary)	Source
Dhanasekaran et al.	2001	cDNA	28(14/14)	www.pathology.med.umich.edu
Welsh et al.	2001	HG-U95A	34(9/25)	public.gnf.org/cancer/prostate/
Singh et al.	2002	HG-U95Av2	102(50/52)	www.broad.mit.edu/
Lapointe et al.	2004	cDNA	103(41/62)	GSE3933
Yu et al.	2004	HG-U95Av2	146 (81/65)	GSE6919
Varambally et al.	2005	HG-U133 Plus2	13 (6/7)	GSE3325
Nanni et al.	2006	HG-U133A	30 (7/23)	GSE3868
Tomlins et al.	2006	cDNA	57(17/30)	GSE6099
Wallace et al.	2008	HG-U133A2	89 (20/69)	GSE6956

studies (Welsh, Yu, Lapointe, Varambally, Singh, Wallace, Nanni, Tomlins and Dhanasekaran) which contain normal and primary cancer samples. Details of the nine studies are listed in Table 8. This example data can be downloaded at `http://www.biostat.pitt.edu/bioinfo/software.htm`.

The users can use the following code to replicate the results in previous sections:

```
rm(list=ls())
#--------------------importing data into R-------------------------------------------------------------#
library(MetaDE)
study.names<-c("Welsh","Yu","Lapointe","Varambally","Singh","Wallace","Nanni","Dhanasekaran",
"Tomlins")
prostate.raw<-MetaDE.Read(study.names,skip=rep(1,9),via="txt",matched=T,log=F)
#--------------------merge and filter data-------------------------------------------------------------#
prostate.merged<-MetaDE.merge(prostate.raw)
dim(prostate.merged[[1]][[1]])
prostate.filtered<-MetaDE.filter(prostate.merged,c(0.3,0.3))
dim(prostate.filtered[[1]][[1]])
#--------------------MetaQC----------------------------------------------------------------------------#
library(MetaQC)
Data.QC<-list()
for(i in 1:9){
    colnames(prostate.filtered[[i]][[1]])<-prostate.filtered[[i]][[2]]
    Data.QC[[i]]<-impute.knn(prostate.filtered[[i]][[1]])$data
    print(dim(Data.QC[[1]]))
}
names(Data.QC)<-names(prostate.filtered)
start<-Sys.time()
ProstateQC<-MetaQC(Data.QC, "c2.all.v3.0.symbols.gmt", filterGenes=F,verbose=TRUE,isParallel=TRUE,
nCores=12, resp.type="Twoclass")
runQC(ProstateQC, B=1e4, fileForCQCp="c2.all.v3.0.symbols.gmt")
QC_time<-Sys.time()-start
png(filename = "Prostate_QC0421.png", width = 3500, height = 3500,res=600)
plot(ProstateQC)
```

```
dev.off()
jpeg(filename = "Prostate_QC0421.jpeg", width = 3500, height = 3500,res=600)
plot(ProstateQC)
dev.off()
#-------------------------------------------------------------------------------------------------#
# (1) To remove the three  studies ("Nanni","Dhanasekaran","Tomlins") with bad quality
# (2) To remerge the remaining six studies
# (3) To re-filter the data
#-------------------------------------------------------------------------------------------------#
study.names<-c("Welsh","Yu","Lapointe","Varambally","Singh","Wallace")
data.QC.raw<-list()
for(i in 1:length(study.names)){
    data.QC.raw[[i]]<-prostate.raw[[study.names[[i]]]]
}
names(data.QC.raw)<-study.names
data.QC.merged<-MetaDE.merge(data.QC.raw)
dim(data.QC.merged[[1]][[1]])
data.QC.filtered<-MetaDE.filter(data.QC.merged,c(0.2,0.2))
dim(data.QC.filtered[[1]][[1]])

#---------------------- MetaDE-----------------------------------------------------------------#
start<-Sys.time()
MetaDE.Res<-MetaDE.rawdata(data.QC.filtered,ind.method=rep("modt",6),meta.method=c("Fisher","maxP",
"roP","AW"),rth=4,nperm=300,asymptotic=F)
b<-Sys.time()-start
print(b)
mylty<-rep(c(1,2),c(6,4))
mycol<-c(rep("black",6),c("red","green","blue","orange"))
mylwd<-rep(c(1,2),c(6,4))
mypch<-1:10
png(filename = "MetaDE_Prostate0421.png", width = 3500, height = 3500,res=600)
    draw.DEnumber(MetaDE.Res,0.05,mlty=mylty,mcol=mycol,mlwd=mylwd,mpch=mypch,FDR=T)
dev.off()
#-------------------------------MetaPath-------------------------------------------------------#
library(MetaPath)
library(GSA)
library(Biobase)
library(genefilter)
library(GSEABase)
library(limma)
Prostate.data=vector(mode = "list", length =length(data.QC.filtered) )
for(t1 in 1:length(data.QC.filtered)){
    temp<-impute.knn(data.QC.filtered[[t1]][[1]])$data
    Prostate.data[[t1]]=list(x=temp, y=data.QC.filtered[[t1]][[2]], geneid=rownames(data.QC.filtered
```

```
}
data(pathway.DB)
start<-Sys.time()
prostate.MAPE<-MAPE(arraydata=Prostate.data,pathway.DB=pathway.DB,resp.type="twoclass",stat='Fisher',
MetaPath.Time<-Sys.time()-start

subset(prostate.MAPE$qvalue, MAPE_I<0.2)
plot.MAPE(prostate.MAPE, cutoff=.2, MAPE.method='MAPE_I')
```

9 Reporting Bugs and Errors

Please contact us with any bug or difficulty you may discover while running this program. Please feel free to contact:Dongwan D. Kang (donkang75@gmail.com) for the MetaQC package; Xingbin Wang(xingbinw@gmail.com) or Jia Li (jiajiaysc@gmail.com) for the MetaDE package; Kui Shen (kuishen@gmail.com) for the MetaPath package.

References

[1] Xingbin Wang, Dongwan Kang, Kui Shen, Chi Song, Lunching Chang, Serena G. Liao, Zhiguang Huo, Naftali Kaminski, Etienne Sibille, Yan Lin, Jia Li and George C. Tseng (2012) An R Package suite for Microarray Meta-analysis in Quality Control, Differentially Expressed Gene Analysis and Pathway Enrichment Detection.

[2] Benjamini, Y. and Hochberg, Y. Controlling the False Discovery Rate - a Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society Series B-Methodological*, **57**, 289-300,1995.

[3] Choi JK, Yu U, Kim S, Yoo OJ. Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, 19 Suppl 1:i84-90,2003.

[4] Kangwan D. Don and George C. Tseng. (2012) MetaQC: objective quality control and inclusion/exclusion criteria for genomic meta-analysis. *Nucleic Acids Research*, 40, e15.

[5] DeConde, R.P., Hawley, S., Falcon, S., Clegg, N., Knudsen, B. and Etzioni, R. Combining results of microarray experiments: a rank aggregation approach. *Stat Appl Genet Mol Biol*, **5**, Article15,2006.

[6] Dreyfuss, J.M., Johnson, M.D. and Park, P.J. (2009) Meta-analysis of glioblastoma multiforme versus anaplastic astrocytoma identifies robust gene markers. *Molecular cancer*, 8, 71.

[7] Dunlap WP, Cortina JM, Vaslow JB, Burke MJ. Meta-analysis of experiments with matched groups or repeated measures designs. *Psychological Methods* 1996, 1(2):170-177, 1996.

[8] Efron B., Tibshirani, R., Storey J. D., and Tusher V. Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association* 96, 1151-1160, 2001.

[9] Florian Hahne, Wolfgang Huber, Robert Gentleman, Seth Falcon. Bioconductor Case Studies (Use R!) *Springer* ISBN: 0387772391

[10] Fisher R. Combining independent tests of significance. *American Statistician*, 2(5):30 1948.

[11] Hedges,L.V. Distribution theory for glasss estimator of effect size and related estimators. *J. Educ. Stat.*, 6, 107C128, 1981.

[12] Hedges L, Olkin I. Statistcal Methods for meta-analysis. London: Academeic Press, 1985.

[13] Hong, F. and Breitling, R. A comparison of meta-analysis methods for detecting differentially expressed genes in microarray experiments. *Bioinformatics*, **24**, 374-382.

[14] Hong, F., Breitling, R., McEntee, C.W., Wittner, B.S., Nemhauser, J.L. and Chory, J. RankProd: a bioconductor package for detecting differentially expressed genes in meta-analysis. *Bioinformatics*, **22**, 2825-2827, 2006.

[15] Lu, S., Li, J., Song, C., Shen, K. and Tseng, G.C. Biomarker detection in the integration of multiple multi-class genomic studies. *Bioinformatics*, 26, 333-340, 2010.

[16] Li J and Tseng,G.C. An adaptively weighted statistic for detecting differential gene expression when combining multiple transcriptomic studies. *Annals of Applied Statistics*. 5:994-1019, 2012.

[17] Kui Shen and George C Tseng. (2010) Meta-analysis for pathway enrichment analysis when combining multiple microarray studies. *Bioinformatics*. 26:1316-1323.

[18] Morris, S. B.. Distribution of the standardized mean change effect size for meta-analysis on repeated measures. *British Journal of Mathematical and Statistical Psychology*, 53, 17C29,2000.

[19] Art B. Owen(2009) KARL PEARSON'S META-ANALYSIS REVISITED, *The Annals of Statistics*,37(6B): 3867-3892, 2009.

[20] Ramasamy, A., Mondry, A., Holmes, C.C. and Altman, D.G. Key issues in conducting a meta-analysis of gene expression microarray datasets. *PLoS Med*, 5, e184,2008.

[21] Rhodes, D.R., Barrette, T.R., Rubin, M.A., Ghosh, D. and Chinnaiyan, A.M. Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer. *Cancer Res*, **62**, 4427-4433,2002.

[22] Rhodes, D.R., Yu, J., Shanker, K., Deshpande, N., Varambally, R., Ghosh, D., Barrette, T., Pandey, A. and Chinnaiyan, A.M. Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *Proc Natl Acad Sci U S A*, 101, 9309-9314,2004.

[23] Shuya Lu, Jia Li, Chi Song, Kui Shen and George C Tseng. (2010) Biomarker Detection in the Integration of Multiple Multi-class Genomic Studies. *Bioinformatics*. 26:333-340.

[24] Stouffer, S., Suchman,E., DeVinnery,L., Star,S.,, and Wiliams,J.. The American Soldier,volumn I: Adjustment during Army Life. *Princeton University Press*, 1949.

[25] Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, 102, 15545-15550.

[26] George C. Tseng, Debashis Ghosh and Eleanor Feingold. (2012) Comprehensive literature review and statistical considerations for microarray meta-analysis. *Nucleic Acids Research* accepted

[27] L.H.C. Tippett. The Methods in Statistics. *Williams and Norgate, Ltd.*, **1** edition, 1931.

[28] Xingbin Wang, Yan Lin, Chi Song, Etienne Sibille and George C Tseng (2012). A statistical framework to integrate weak-signal microarray studies adjusted for confounding variables with application to major depressive disorder. *BMC bioinformatics*.13:15.

[29] Wilcoxon,Frank Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80-83, 1945.

[30] Wilkinson B. A statistical consideration in psychological research. *Psychol Bull*, 48(3):156-158,1951.
The source code for example