AB1700 Microarray Data Analysis

Yongming Andrew Sun, Applied Biosystems
sunya@appliedbiosystems.com

October 30, 2017

Contents

1 ABarray Package Introduction

The package *(*ABarray) is designed to work with Applied Biosystems whole genome microarray platform, as well as any other platform whose data can be tranformed into expression data matrix. In the following functions described, items 1 and 2 are specific to AB1700 datasets. Items 3 to 9 can be applied to any datasets. However, for AB1700 data, filtering is performed using S/N ratio threshold (default = 3), while filtering is not applied to other data.

The package *(*ABarray) will perform the following functions:

1. Read output from AB1700 software output (AB1700 specific)
2. Perform analysis on hybridization control spike-ins (AB1700 specific)
3. Perform raw data QA and associated plots (see *doPlotEset*) including:
   - boxplot for signal distrubution range
   - MA plot for signal distribution and signal variability
   - CV plot for variation among hybridization replicates
   - Scatter plot for correlation between hybridization arrays
   - Correlation heatmap for visualization
   - S/N detection concordance
4. Perform quantile normalization
5. Repeat data QA after normalization. See *doPlotEset*
6. Perform t test, fold change and ANOVA and produce graphics to visualize t test results. The default t test assumes unequal variances. See *doPlotFCT* for more details.
7. Peform LPE for low number of replicates. See *doLPE* for more details
8. For more details on ANOVA analysis, see *doANOVA*
9. For drawing Venn diagram, see *doVennDiagram*

1.1 Required Files and Format

To take full advantage of automated process provided by package *ABarray*, two files should be available before running *ABarray*: a data file and an experiment design file.

- Experiment Design File. The rows of the file are samples or arrays. The first column should be sampleName. Perhaps, sampleName should be concise and no spaces between characters. Additional columns maybe assayName and arrayName (one of these). Additional columns should specify what type of samples. Note: It is best to have assayName the same as in dataFile. See an example of experiment design file (Table 1).

  In the example (Table 1), assayName is included and arrayName is not. There is no need for both assayName and arrayName to be listed in the design file. Note: every column name in the header is ONE word, there

Table 1: Experiment design and sample information.

	sampleName	assayName	tissue
1	A1	HB003U2_9/2/05_1:55_PM	TissueA
2	A2	HB003U3_9/2/05_2:11_PM	TissueA
3	A3	HB003U8_9/2/05_2:26_PM	TissueA
4	B1	HB003TV_9/2/05_2:41_PM	TissueB
5	B2	HB003TW_9/2/05_2:55_PM	TissueB
6	B3	HB003TZ_9/2/05_3:10_PM	TissueB

needs NO spaces in each word. Make sure sampleName, assayName, arrayName spelled correctly. There is no need to worry about lower case or upper case.

- Data File. The rows of the file represent probes. The first column is probeID (or Probe Name), the second column is geneID, next set of columns should contain Signal, S/N and FLAGS for all samples. Optionally columns with Assay_Normalized_Signal, SDEV, CV can be included in the data file, but these columns will be ignored in the analysis. The columns are: probeID, geneID, Signal, S/N, Flags, and optionally SDEV, CV, and Assay_Normalized_Signal.
- Control Probes. It is optional to have control probes. If they are present, plots will be generated for the control probes, but they will be removed for further analysis.

1.2 Data File and Experiment Design File

The data file and experiment design file are essential. They should be either tab-delimt txt file or comma-delimit csv file. It is very important to make these files compliant. If array name in the header of data file is present and arrayName is defined in experiment design file, arrayName will be used to distinguish which column is for which hybridization sample. In the absence of arrayName in experiment design file, assayName will be used to identify which column in data file is for which hybridization sample. It is important to spell assayName in the experiment design file the same as in data file. For example, if the assayName in data file is called *HA004J7_1A_2*, which will show as *Signal HA004J7_1A_2* in data file, the assayName in experiment design file should be called exactly as *HA004J7_1A_2*.

Note: assayName should be unique and not part of another assayName. For example, assayName S1 is part of another assayName S11. In this case, S1 is better renamed to S01.

Sometimes, there could be more assays in data file than we actually want to analyze. For example, we decided to ignore one or two of the arrays, because the images are bad and there are reasons to exlude these arrays. But instead to delete the related columns in data file, we simply change the experimental design file to just include those arrays we need to analyze. However, on the other hand, if there are more hybridization samples in experiment design file than in data file, it will produce an error, as this does not make sense.

1.3 ABarray Function Details

1.3.1 Processing of control probes

There are a number of internal control probes for AB1700 Expression Array Systems. The QC analysis is performed on the following controls before any data normalization.

1. Hybridization Controls.
2. IVT Labeling Controls.
3. RT Labeling Controls.
4. Negative Controls.

All other controls are ignored in the analysis. The behavior of signals of the controls are plotted in the figures. The control probes and associated measurements are removed before data normalization and transformation.

1.3.2 Data normalization and transformation

Once the control probes and associated measurements are removed, the quantile normalization procedure is applied. The data is then transformed into log2 scale. If the imputation is selected (default is to use average imputation), the missing value imputation is applied after quantile normalization and log2 transformation. What is considered missing value? If the *Flag* value of a probe is above 5000 (actually above $2^{12} = 4096$), the signal value of the probe is considered missing value and is replaced with NA. The imputation is then performed on these NA values. If 'No imputation' is used, the probe signal values are used as is (they are not replaced with NA, nor are they imputed) in the downstream analysis.

There are 2 choices in the program for imputation.

1. No imputation. Imputation is not performed, and the signal values are used as is.
2. Average imputation. The probe signal values are averaged from other arrays in the same subgroup. The missing values are replaced with the averaged probe. If there is not enough values to average for certain probes, the signal values of these probes remain missing (NA).

1.3.3 Array QC Analysis

A number of array QC analysis is performed and associated figures are generated. The QC analysis is performed on both the pre-normalized data and normalized data.

- boxplot
- MA plot
- scatter plot
- correlation plot
- CV plot
- percentage of probes detected
- probe detection concordance

1.3.4 Statistical Analysis for Differential Expression

A default two sample t test is performed between any 2 subgroups. A probe filtering procedure precedes the t test. If a probe is not detected (S/N < 3) in both subgroups, it is not included in the t test. If a probe is not detected in more than 50% of samples in a subgroup, it is not detected in that subgroup. The 50% is default, and can be set to a different threshold value.

The fold change (FC) and log2(FC) for the detected probes are also calculated. The FC is divided into 5 FC bins. In addition, the p values from the t test are used to calculate adjusted p value as FDR using Biocondutor package *multtest*. The FDR is calculated using Benjamini-Hochberg procedure. The results are written into files and plotted into MA and volcano plots.

If there are more than 2 subgroups, ANOVA will also be performed. The probe detection threshold (same as t test) is also applied.

1.3.5 Clustering heatmap

The clustering heatmap is also produced if the number of differentially expressed probes is less than 800.

1.3.6 Description of Results

There are 3 folders generated after the analysis for each group.

- DataResult. This folder contains several text files.
  - 'ControlRawData.csv' contains signals of control probes.
  - 'ProbeImputed_' contains probeID whose expression measurement may be imputed.

  - 'QuantileNormalizedExpression' contains quantile normalized, log2 transformed, and imputed values.
  - 'RawExpressionData.csv' contains non-normalized, non-imputed raw measurement.
  - 'ST_DetectPct' files contain statistical analysis results.

- jpgFigure. This folder contains figures in jpg or bitmap format.
- pdfFigure. This folder contains figures in pdf format.

2 ABarray Function Demonstration

After loading the data file and experiment design file, the ABarray functions check to see if control signals are present in data file. If they are present, several plots for control assesement will be produced. The program then removes these control probes and perform QA analysis on raw data. It also produces a number of plots to visualize those QA assesement. After this, it performs quantile normalization and repeat QA assessment for quantile normalized data. Futhermore, fold changes between subgroups and t test to identify differential expressed genes will be performed (and ANOVA if there are more than 2 subgroups in the group).

2.1 Required Libraries

The bioconductor library *Biobase* and *multtest* are required to run ABarray. In addition, several other optional libraries may be needed to perform various functions. *impute* is needed if KNN imputation will be performed, *limma* is needed if drawing Venn diagram function is performed.

2.2 Preparing R for Data Analysis

After launching R, we need to set the R working directory to inform R where we find data file and design file. It also inform R where to store the analysis results and associated figures from the analysis.

To set R working directory, choose 'Change dir...' from 'File' menu, and a dialog box appears to let you browse which directory (folder) to set as working directory (See Figure 1 and Figure 2).

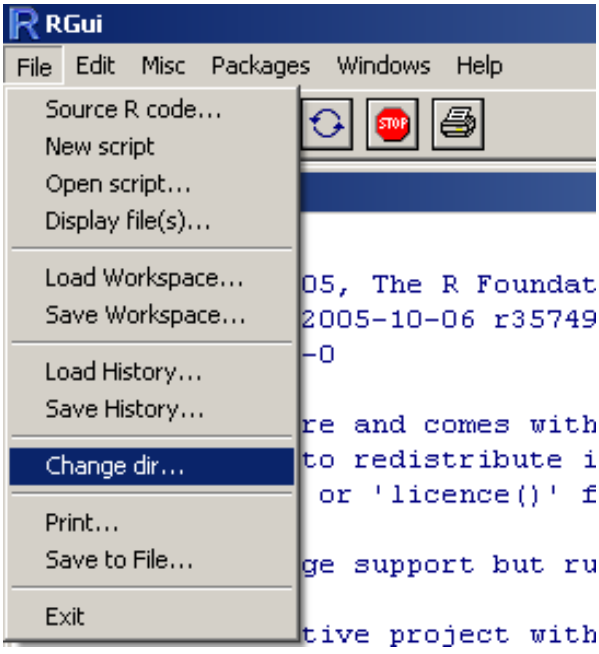


Figure 1: Change R working directory. Select Change dir... from File menu.

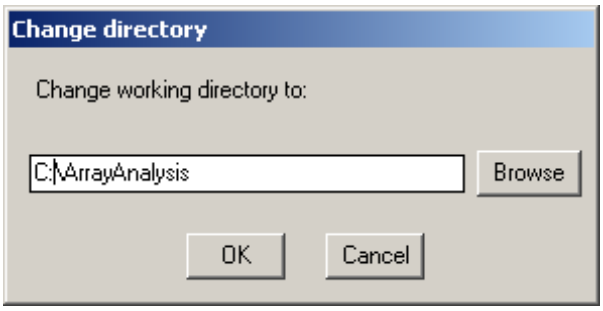


Figure 2: Browse the directory to set as working directory.

2.3 File List in Working Directory

To see the content of the working directory, type the following command

```
> dir()
```

```
[1] "ABarray_Example.tex"      "ExperimentDesign.csv"    "RawExpressionABData.txt"
```

2.4 Loading ABarray library

The *ABarray* has to be loaded to perform the automated analysis. To load *ABarray*, type the following command after launching R.

```
> library(ABarray)
```

2.5 Performing Automated Expression Analysis

The automated gene expression analysis requires only one command. The function takes 3 parameter arguments. The first argument is the name of the data file. The second argument is the name of the expriment design file. The third argument is the name of the experiment group defined in the experiment design file on which t test and other analysis should be performed. It will generate an `ExpressionSet` expression value object.

In the following example, the name of the data file is called 'RawExpressionABData.txt'; the name of the experiment design file is called 'ExperimentDesign.csv'; and the 'tissue' is the group name selected to perform analysis on. The expression value object is assigned to `eset` object. Since the 'tissue' has 2 subgroup, ANOVA will NOT be performed in this example.

```
> eset = ABarray("RawExpressionABData.txt", "ExperimentDesign.csv", "tissue")
```

```
Using assayName to match experiment with signal in file: RawExpressionABData.txt
The sample names are:
[1] "A1" "A2" "A3" "B1" "B2" "B3"
AssayNames in dataFile: RawExpressionABData.txt
[1] "HB003U8_9/2/05_2:26_PM" "HB003U2_9/2/05_1:55_PM" "HB003TZ_9/2/05_3:10_PM" "HB003U3_9/2/05_2:11_PM"
[5] "HB003TW_9/2/05_2:55_PM" "HB003TV_9/2/05_2:41_PM"
Reading data from RawExpressionABData.txt ......
    This may take several minutes....
Checking data format:::::::::::::::::::||

The results will be in the folder: Result_tissue/
[1] "Creating plot for Signal Hybridization_Control ..."
[1] "Creating plot for Signal Negative_Control ..."
[1] "Creating plot for Signal IVT_Kit_Control_BIOB ..."
[1] "Creating plot for Signal IVT_Kit_Control_BIOC ..."
[1] "Creating plot for Signal IVT_Kit_Control_BIOD ..."
[1] "Creating plot for Signal RT_Kit_Control_DAP ..."
[1] "Creating plot for Signal RT_Kit_Control_LYS ..."
[1] "Creating plot for Signal RT_Kit_Control_PHE ..."

Perform basic analysis for non-normalized data ...
[1] "Creating barplot for probes detectable ... QC_DetectableProbeSN3"
[1] "Creating boxplot for Raw_tissue  ..."
[1] "Creating correlation matrix and plot Raw_tissue  ..."
[1] "Creating detection concordance plot Raw_tissue ..."
Perform Average imputation for probes with FLAG > 5000
[1] "Array A1 has 72 FLAGS > 5000" "Array A2 has 89 FLAGS > 5000" "Array A3 has 47 FLAGS > 5000"
[4] "Array B1 has 91 FLAGS > 5000" "Array B2 has 96 FLAGS > 5000" "Array B3 has 75 FLAGS > 5000"
The signals of each flagged probe were replaced with average signals
   from replicate arrays within the same subgroup (TissueA, TissueB).
```

```
The imputed probes were written to file: Result_tissue/DataResult/ProbeImputed_tissue.csv

 Quantile normalized data was saved to:
     Result_tissue/DataResult/QuantileNormalizedExpression.csv
The following probes still contain missing values even after imputation:
[1] "128976" "215914" "226588" "249634"

The expression data object was saved to R workspace file:
     ExperimentDesign_eset_28Mar2006.Rdata

Perform data analysis on Quantile normalized data ...
[1] "Creating boxplot for Quantile_tissue  ..."
[1] "Creating MA plot for Quantile_tissue"
[1] "Creating MA Plot Quantile_tissue: TissueA ..."
[1] "Creating QC_CVplot_Quantile_tissue_TissueA ..."
[1] "Creating MA Plot Quantile_tissue: TissueB ..."
[1] "Creating QC_CVplot_Quantile_tissue_TissueB ..."
[1] "Creating scatter plot filtering S/N < 3  for Quantile_tissue  ..."
[1] "Creating correlation matrix and plot Quantile_tissue  ..."
[1] "Creating detection concordance plot Quantile_tissue ..."

Performing t test between TissueA and TissueB ...
1 probes had sdev = 0, they were removed in the t test
[1] "520680"
[1] "Creating volcano plot ..."
[1] "Writing t test result to file:  Result_tissue/DataResult/ST_DetectPct50FCpvalFDR_TissueA-TissueB.csv"

The t test was performed if the probe shows S/N >= 3 in 50% or more samples
    in either TissueA or TissueB
The number of probes after filtering is: 22323

p=0.01, significant probes: 12611
[1] "Creating Fold Change plot for: TissueA-TissueB pVal 0.01  ..."
[1] "Clustering was not performed, because there are 12611 probes."
[1] "    The limit of probe counts for clustering is: 800"
p=0.05, significant probes: 16461
[1] "Creating Fold Change plot for: TissueA-TissueB pVal 0.05  ..."
[1] "Clustering was not performed, because there are 16461 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.01, significant probes: 10666
[1] "Creating Fold Change plot for: TissueA-TissueB FDR 0.01  ..."
[1] "Clustering was not performed, because there are 10666 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.05, significant probes: 15681
[1] "Creating Fold Change plot for: TissueA-TissueB FDR 0.05  ..."
[1] "Clustering was not performed, because there are 15681 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.1, significant probes: 17416
[1] "Creating Fold Change plot for: TissueA-TissueB FDR 0.1  ..."
[1] "Clustering was not performed, because there are 17416 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.25, significant probes: 19373
[1] "Creating Fold Change plot for: TissueA-TissueB FDR 0.25  ..."
[1] "Clustering was not performed, because there are 19373 probes."
[1] "    The limit of probe counts for clustering is: 800"
```

3 Information about QC and Analysis Results

The function *ABarray* produced a lot of results. In addition, it returns an `ExpressionSet` object. This make it possible to use vast amount of packages and functions in Bioconductor (www.bioconductor.com) for further statistical analysis, e.g., classification, prediction.

The results will be saved in a folder within the R working directory. The name of the folder begins with Result_ and the name of the group. In the example run, the fold name is Result_tissue. There will be 3 directories created inside this folder to store the analysis results: One DataResult folder to store normalized signals, p values and FDR from t test, and ANOVA p values if ANOVA is performed. One jpgFigure folder to store QC and analysis plots, and one pdfFigure folder. In pdfFigure, the plots are the same as in jpgFigure, but in higer resolution of pdf format.

Now, let's take a look at the `eset`.

```
> eset
```

```
Expression Set (exprSet) with
        32878 genes
        6 samples
                 phenoData object with 3 variables and 6 cases
         varLabels
                sampleName: read from file
                assayName: read from file
                tissue: read from file
```

We can also check the content of the experiment design by method in *Bioconductor* package *Biobase*, *pData*.

```
> pData(eset)
```

```
  sampleName              assayName  tissue
1         A1 HB003U2_9/2/05_1:55_PM TissueA
2         A2 HB003U3_9/2/05_2:11_PM TissueA
3         A3 HB003U8_9/2/05_2:26_PM TissueA
4         B1 HB003TV_9/2/05_2:41_PM TissueB
5         B2 HB003TW_9/2/05_2:55_PM TissueB
6         B3 HB003TZ_9/2/05_3:10_PM TissueB
```

4 Data Analysis on eSet object: doPlotEset

In this example, we will show the utility of *doPlotEset*. This function perform a number of data quality assessment and some statistical analysis (see *doPlotFCT* for fold change and t test, *doANOVA* for one- or two-way ANOVA analysis).

4.1 Data Quality Plots

- Boxplot showing distribution of signal range.
- MA plot showing data distribution and signal variability between hybridizations. There might be several MA plots produced. All pairs of hybridizations within the group will be plotted if number of pairs is less than 35 pairs. In addition, MA plots will be produced for each pairs within subgroup of the group to see signal variability between each individual members of the subgroup.
- CV plot showing amount of variation within replicate hybridizations of each subgroup.
- Scatterplot showing visually the correlation between any two hybridizations within the group. Correlation coefficient is also calculated for each pair.
- Correlation heatmap showing correlation between any pairs of hybridization. The heatmap is color coded to quickly visualize the extent of the correlation.
- S/N detection concordance. The signal is detected if the probe shows S/N ratio equal to or above a user defined threshold (default is 3). The S/N detection heatmap shows detection concordance between any pairs of hybridization.

4.2 Statistical Analysis

The function *doPlotEset* accepts a parameter *test* either *TRUE* (the default) or *FALSE*. If *test* = *TRUE* (the default), regular t test will be performed. For details, please refer to *doPlotFCT* in *ABarray* package. If there are more than two subgroups within the group, pairwise t test will be performed on each pairs of subgroup. In addition, one way ANOVA test will also be performed. The t test results are written to files and several graphics plots will be produced. ANOVA test results are written to file as well if it is performed.

4.3 Requred Data and Function Arguments

- An `ExpressionSet` object. The S/N ratio information is stored in `assayDataElement(eset, "snDetect")`. This will automatically be created from *ABarray* in *ABarray* package. The analysis will filter probes based on S/N >= 3 in 75% of samples. If S/N is not available, all probes will be used for analysis (no filtering will be applied).
- An *group* parameter. The group is a factor that t test should be performed on. The name of the group should correspond to one of the names in experiment design file that define sample type. If there are 3 or more levels for the group, pairwise t test and fold change will be peformed for all possible pairs.
- An optional *name* parameter. The name will be used for output filenames to distinguish different types of runs. For example, *quantile* to indicate the results were produced after *quantile* process. *raw* to indicate the results were produced with raw data. If *name* is missing, the analysis and output files will still be produced.
- An optional *snThresh* parameter. If this is ommitted, default snThresh = 3 will be used. To use a different threshold, use *snThresh* = *newValue*.
- Parameter *test*. By default, *test* = *TRUE*. This indicates to perform statistical test (t test and ANOVA if applicable). To avoid these test, use *test* = *FALSE*. For more details about the t test and fold change, see *doPlotFCT* function. For more details for ANOVA analysis, see *doANOVA*.

4.4 doPlotEset Function Demonstration

First let's see what the `eset` contains

```
> eset

Expression Set (exprSet) with
        32878 genes
        6 samples
                 phenoData object with 3 variables and 6 cases
         varLabels
                sampleName: read from file
                assayName: read from file
                tissue: read from file
```

Let's check the experiment design and pick a group for testing

```
> pData(eset)

  sampleName              assayName  tissue
1         A1 HB003U2_9/2/05_1:55_PM TissueA
2         A2 HB003U3_9/2/05_2:11_PM TissueA
3         A3 HB003U8_9/2/05_2:26_PM TissueA
4         C1 HB003UA_9/2/05_3:24_PM TissueC
5         C2 HB003UB_9/2/05_3:44_PM TissueC
6         C3 HB003UD_9/2/05_3:58_PM TissueC
```

Let's begin to perform the analysis

```
> doPlotEset(eset, "tissue", name = "quantile")
```

```
[1] "Creating boxplot for quantile_tissue  ..."
[1] "Creating MA plot for quantile_tissue"
[1] "Creating MA Plot quantile_tissue: TissueA ..."
[1] "Creating CVplot_quantile_tissue_TissueA ..."
[1] "Creating MA Plot quantile_tissue: TissueC ..."
[1] "Creating CVplot_quantile_tissue_TissueC ..."
[1] "Creating scatter plot for quantile_tissue  ..."
[1] "Creating scatter plot filtering S/N < 3  for quantile_tissue  ..."
[1] "Creating correlation matrix and plot quantile_tissue  ..."
[1] "Creating detection concordance plot quantile_tissue ..."

Performing t test between TissueA and TissueC ...
2 probes had sdev = 0, they were removed in the t test
[1] "207943" "520680"
[1] "Creating volcano plot ..."
[1] "Writing t test result to file:  Result_tissue/DataResult/DetectPct50FCpvalFDR_TissueA-TissueC.csv"

The t test was performed if the probe shows S/N >= 3 in 50% or more samples
    in either TissueA or TissueC
The number of probes after filtering is: 21628

p=0.01, significant probes: 3470
[1] "Creating Fold Change plot for: TissueA-TissueC pVal 0.01  ..."
[1] "Clustering was not performed, because there are 3470 probes."
[1] "    The limit of probe counts for clustering is: 800"
p=0.05, significant probes: 7595
[1] "Creating Fold Change plot for: TissueA-TissueC pVal 0.05  ..."
[1] "Clustering was not performed, because there are 7595 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.01, significant probes: 429
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.01  ..."
[1] "Creating cluster heatmap for FDR 0.01 using Correlation ..."
[1] "Creating cluster heatmap for FDR 0.01 using Euclidean ..."
FDR=0.05, significant probes: 2700
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.05  ..."
[1] "Clustering was not performed, because there are 2700 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.1, significant probes: 5577
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.1  ..."
[1] "Clustering was not performed, because there are 5577 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.25, significant probes: 10894
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.25  ..."
[1] "Clustering was not performed, because there are 10894 probes."
[1] "    The limit of probe counts for clustering is: 800"
```

5 Fold Change and t test: doPlotFCT

In this example, we will show the utility of *doPlotFCT*. This function perform calculation for fold change and t test. The t test p values are adjusted using Benjamini-Hochberg FDR at preset levels: 0.01, 0.05, 0.1, 0.25, and raw p values 0.01 (no FDR) and 0.05 (no FDR). The results are written to a cvs file, whose name begin with DetectPctFCpvalFDR and the group name that the test was performed on. In addition, fold change are plotted into graphics for various FDR levels, and volcano plot of fold change and t test p values are plotted.

The analysis uses S/N ratio for filtering. By default, if a probe is detected in 50% or more samples within any subgroup, it is included in the analysis. These thresholds can be changed at run time with optional parameters.

If there are less than 800 probes for a particular threshold level, clustering heatmap may also be produced. There will be 2 clustering figures, one uses correlation coefficient for clustering, and the other one uses distance for clustering.

The t test p values will be calculated using R implementation of t test assuming unequal variances and using log2 transformed normalized data. The fold changes will also be calculated. If a paired t test is performed, the fold changes will be average of paired fold changes. If not paired t test, the fold changes will be fold changes of averaged expression values.

5.1 Required Data and Function Arguments

- An `ExpressionSet` object. The S/N ratio information is stored in `assayDataElement(eset, "snDetect")`. This will automatically be created from *ABarray* in *ABarray* package. The analysis will filter probes based on S/N and percentage of samples detectable within subgroups. If S/N is not available, all probes will be used for analysis (no filtering will be applied).
- An *group* parameter. The group is a factor that t test should be performed on. The name of the group should correspond to one of the names in experiment design file that define sample type. If there are 3 or more levels for the group, pairwise t test and fold change will be peformed for all possible pairs.
- An optional *grpMember* parameter. If this is missing, all members in the group will be used for analysis. If it is defined and passed to the *doPlotFCT* analysis will be performed only on those members within the group.
- order1 optional, For a pairwise comparison the ordering of the first group of replicates
- order2 optional, For a pairwise comparison the ordering of the second group of replicates
- snThresh optional S/N ratio threshold. Default = 3
- detectSample Percentage of samples the probe is detected in order to consider in t test. Default = 0.5

5.2 doPlotFCT Function Demonstration

First let's see what the `eset` contains

```
> eset

Expression Set (exprSet) with
        32878 genes
        6 samples
                 phenoData object with 3 variables and 6 cases
         varLabels
                sampleName: read from file
                assayName: read from file
                tissue: read from file
```

Let's check the experiment design and pick a group for testing

```
> pData(eset)

  sampleName              assayName  tissue
1         A1 HB003U2_9/2/05_1:55_PM TissueA
2         A2 HB003U3_9/2/05_2:11_PM TissueA
3         A3 HB003U8_9/2/05_2:26_PM TissueA
4         C1 HB003UA_9/2/05_3:24_PM TissueC
5         C2 HB003UB_9/2/05_3:44_PM TissueC
6         C3 HB003UD_9/2/05_3:58_PM TissueC
```

Let's begin to perform the analysis

```
> doPlotFCT(eset, "tissue")
```

```
Performing t test between TissueA and TissueC ...
2 probes had sdev = 0, they were removed in the t test
[1] "207943" "520680"
[1] "Creating volcano plot ..."
[1] "Writing t test result to file:  Result_tissue/DataResult/DetectPct50FCpvalFDR_TissueA-TissueC.csv"

The t test was performed if the probe shows S/N >= 3 in 50% or more samples
    in either TissueA or TissueC
The number of probes after filtering is: 21628

p=0.01, significant probes: 3470
[1] "Creating Fold Change plot for: TissueA-TissueC pVal 0.01  ..."
[1] "Clustering was not performed, because there are 3470 probes."
[1] "    The limit of probe counts for clustering is: 800"
p=0.05, significant probes: 7595
[1] "Creating Fold Change plot for: TissueA-TissueC pVal 0.05  ..."
[1] "Clustering was not performed, because there are 7595 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.01, significant probes: 429
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.01  ..."
[1] "Creating cluster heatmap for FDR 0.01 using Correlation ..."
[1] "Creating cluster heatmap for FDR 0.01 using Euclidean ..."
FDR=0.05, significant probes: 2700
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.05  ..."
[1] "Clustering was not performed, because there are 2700 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.1, significant probes: 5577
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.1  ..."
[1] "Clustering was not performed, because there are 5577 probes."
[1] "    The limit of probe counts for clustering is: 800"
FDR=0.25, significant probes: 10894
[1] "Creating Fold Change plot for: TissueA-TissueC FDR 0.25  ..."
[1] "Clustering was not performed, because there are 10894 probes."
[1] "    The limit of probe counts for clustering is: 800"
```

6 ANOVA analysis

The function *doANOVA* in package *ABarray* is designed to perform one way or two way ANOVA analysis. If only one factor is provided in parameter, one way ANOVA is performed. If two factors are provided, two way ANOVA is performed.

The S/N ratio will be used to perform filtering. If a probe is detected (S/N >= 3, default value) in 50% (default value) or more samples within any subgroups, it is included in the ANOVA analysis. If a probe is not detected in 50% or more samples in all the subgroups, it is removed from ANOVA analysis.

6.1 Required Data Object

The function *doANOVA* works on either an *ExpressionSet* object or expression `matrix`. If *ExpressionSet* object is used, experiment design information should be already in the *ExpressionSet* object. If expression `matrix` is used, rows of the matrix should be probes or genes, and columns of the matrix should be hybridization (or arrays).

- `ExpressionSet` object. The object can be generated from function *ABarray* in package *ABarray*. For additional information about functionality of *ABarray*, refer to document from *ABarray*. The object can also be generated from other packages in *Bioconductor* project. See www.bioconductor.org for more information about other packages.

- factor1. Factor name that the ANOVA test will be performed on. If an `ExpressionSet` object is used, either name or labels can be used. The name should be present in experiment design informaton in the `ExpressionSet` object. Function *ABarray* in package *ABarray* automatically creates such information if appropriate experiment design file is used when function *ABarray* was run. If expression matrix is to be used, `lables` should be used. A `label` is a vector which labels each hybridization (or array) into groups. For example, a `label` <- c('drug1', 'drug1', 'drug2', 'drug2', 'none', 'none') indicates samples for arrays 1 and 2 were treated with drug1, and samples for arrays 5 and 6 were not treated.
- optional factor2. If the second factor is provided, two way ANOVA will be performed.

6.2 doANOVA Analysis Output

If one way ANOVA is performed, a vector with ANOVA p values will be returned. The names of the vector are probeIDs. If two way ANOVA is performed, a matrix will be returned. The rows of the matrix are probeIDs and the columns are the levels of factors and interactions of the levels.

The ANOVA results will be saved into file automatically.

6.3 doANOVA Function Demonstration

6.3.1 Required Data

First let's see what the `eset` contains

```
> eset

Expression Set (exprSet) with
        33096 genes
        12 samples
                 phenoData object with 6 variables and 12 cases
         varLabels
                sampleName: read from file
                assayName: read from file
                arrayName: read from file
                pool: read from file
                IVT: read from file
                multIVT: read from file
```

Let's check the experiment design and pick factors for testing

```
> pData(eset)

   sampleName    assayName arrayName  pool  IVT multIVT
1     EL_1A_1 HA004J2_1A_1   HA004J2 pool1 IVT1    IVT1
2     EL_1A_2 HA004J7_1A_2   HA004J7 pool1 IVT1    IVT1
3     EL_1A_3 HA004JB_1A_3   HA004JB pool1 IVT1    IVT1
4     EL_1B_1 HA004JD_1B_1   HA004JD pool1 IVT2    IVT2
5     EL_1B_2 HA004JI_1B_2   HA004JI pool1 IVT2    IVT2
6     EL_1B_3 HA004JK_1B_3   HA004JK pool1 IVT2    IVT2
7     EL_2A_1 HA004JN_2A_1   HA004JN pool2 IVT1    IVT3
8     EL_2A_2 HA004JO_2A_2   HA004JO pool2 IVT1    IVT3
9     EL_2A_3 HA004JP_2A_3   HA004JP pool2 IVT1    IVT3
10    EL_2B_1 HA004JS_2B_1   HA004JS pool2 IVT2    IVT4
11    EL_2B_2 HA004JT_2B_2   HA004JT pool2 IVT2    IVT4
12    EL_2B_3 HA004JU_2B_3   HA004JU pool2 IVT2    IVT4
```

6.3.2 One Way ANOVA

Let's begin to perform the analysis. Here we take a subset of probes (the first 20 probes) to perform the ANOVA. We chose `multIVT`, since it has 4 levels: IVT1, IVT2, IVT3, IVT4.

```
> aResult <- doANOVA(eset[1:20, ], "multIVT")

Performing one-way ANOVA for multIVT ...
    Probes with S/N < 3 in at least 50% of samples in all subgroups of multIVT were not considered.
[1] "One-way ANOVA results (9 probes) were written to file:
     Result_multIVT/DataResult/ANOVA_oneway_multIVT.csv"
```

The results of this ANOVA analysis were saved into a file called `ANOVA_oneway_multIVT.txt`.
Since we chose to perform one way ANOVA, `aResult` should be a vector of length 20 (we only used 20 probes).

```
> aResult

      100002       100037       100039       100058
1.620190e-03 5.705200e-03 1.009991e-03 9.374965e-05
      100060       100079       100089       100027
1.016003e-01 5.021948e-06 1.462993e-04 1.652952e-01
      100052
5.952003e-01
```

To save the results into file, we use R *write.table* function. It takes an object whos data need to be written to file. Here our object is `aResult`. A file argument (i.e., the name of the file). In the following command, *sep = ","* tells it to save the file as comma delimit format. Argument *row.names = TRUE* tells it to write probeID into file as well.

```
> write.table(aResult, file = "multIVT_oneWay_ANOVA.csv",
+     sep = ",", row.names = TRUE)
```

You can check the content of the file. The first column is probeID, the second column is p values from one way ANOVA analysis.

6.3.3 Two Way ANOVA

Let's perform two way ANOVA. Here we pass 2 factors ('pool' and 'IVT'), and as before we will use only 20 probes for demonstration purpose.

```
> bResult <- doANOVA(eset[1:20, ], "pool", "IVT")

Performing two-way ANOVA for pool and IVT ...
    Probes with S/N < 3 in at least 50% of samples in all subgroups of pool and IVT were not considered.
[1] "Two-way ANOVA results (8 probes) were written to file: Result_pool_IVT/ANOVA_pool_IVT.csv"
```

The results of the ANOVA analysis were saved into file `ANOVA_pool_IVT.txt`.
Let's check if the `bResult` is a matrix

```
> dim(bResult)

[1] 8 3
```

Let's get the p values for the two way ANOVA

```
> bResult

            p(pool)      p(IVT)  p(pool*IVT)
100002 4.215936e-03 0.002562953 3.098940e-02
100037 1.116917e-02 0.156183104 5.328180e-03
100039 5.423037e-04 0.173732795 5.336480e-03
```

```
100058 6.514385e-04 0.009500397 8.888504e-05
100060 7.294050e-01 0.051155428 1.058891e-01
100079 3.244984e-05 0.001540973 5.688470e-06
100089 6.787365e-04 0.961322381 8.243031e-05
100052 7.863245e-01 0.206235388 8.623153e-01
```

To save the result to file, we can use the following command

```
> write.table(bResult, file = "pool_IVT_twoway_ANOVA.csv",
+     sep = ",", row.names = TRUE, col.names = NA)
```

7 LPE analysis

The function *doLPE* in package *ABarray* is based on *LPE* package from www.bioconductor.org.

The *LPE* package describes local-pooled-error (LPE) test for identifying differentially expressed genes especially for low number of replicates (2-3) (See Jain:2003).

The test statistics is calculated as follows, which calculates difference in medians between the two experimental conditions:

$$Z = \frac{Med_1 - Med_2}{\sigma_{pooled}}$$

where the variance is estimated as follows:

$$\sigma^2_{pooled} = \frac{\pi}{2}\left[\frac{\sigma_1^2(Med_1)}{n1} + \frac{\sigma_2^2(Med_2)}{n2}\right]$$

The function *doLPE* will automatically adjust p values using Benjamini-Hochberg FDR approach.

Jain et. al. (2003) Local pooled error test for identifying differentially expressed genes with a small number of replicated microarrays. Bioinformatics, 19, 1945-1951

7.1 Required Data Object

The function *doANOVA* works on an *ExpressionSet* object.

- `ExpressionSet` object. The object can be generated from function *ABarray* in package *ABarray*. For additional information about functionality of *ABarray*, refer to document from *ABarray*. The object can also be generated from other packages in *Bioconductor* project. See www.bioconductor.org for more information about other packages.

  The S/N ratio information is stored in `assayDataElement(eset, "snDetect")`. This will automatically be created from *ABarray* in *ABarray* package. The analysis will filter probes based on S/N >= 3 in 75% of samples. If S/N is not available, all probes will be used for analysis (no filtering will be applied).

- *group* parameter. The group is a factor that the test should be performed on. The name of the group should correspond to one of the names in experiment design file that define sample type. If there are 3 or more levels for the group, the members of the group should be specified for LPE test.

- *member* parameter. Where there is more than 2 levels in the *group* factor, any two members that need to be compared should be specified in this paramter. For example, *member <- c('drug1', 'drug4')* will indicate to perform LPE between *drug1* and *drug4* only.

7.2 doLPE Analysis Output

The function *doLPE* returns a dataframe with p values and FDR adjusted p values for each probes (if S/N available, filtered probes). It also produces two files for preset FDR = 0.01 and FDR = 0.05.

7.3 doLPE Function Demonstration

7.3.1 Required Data

First let's see what the `eset` contains

```
> eset
```

```
Expression Set (exprSet) with
        32878 genes
        6 samples
                phenoData object with 3 variables and 6 cases
         varLabels
                sampleName: read from file
                assayName: read from file
                tissue: read from file
```

Let's check the experiment design and pick factors for testing

```
> pData(eset)
```

```
  sampleName              assayName  tissue
1         A1 HB003U2_9/2/05_1:55_PM TissueA
2         A2 HB003U3_9/2/05_2:11_PM TissueA
3         A3 HB003U8_9/2/05_2:26_PM TissueA
4         B1 HB003TV_9/2/05_2:41_PM TissueB
5         B2 HB003TW_9/2/05_2:55_PM TissueB
6         B3 HB003TZ_9/2/05_3:10_PM TissueB
```

7.3.2 LPE analysis

Let's begin to perform the analysis.

```
> aLPE.result <- doLPE(eset, "tissue")
```

```
Performing LPE analysis for tissue ...
[1] "LPE results were written to file Result_tissue/DataResult/LPE_tissue_TissueB_TissueA.csv"
```

Let's take a few entries in aLPE.result to see what the dataframe looks like.

```
> aLPE.result[1:4, ]
```

```
         x.x.B1   x.x.B2   x.x.B3 median.1  std.dev.1   y.y.A1   y.y.A2   y.y.A3 median.2  std.dev.2 media
520680 21.78399 21.78399 21.78399 21.78399 0.07155299 21.78399 21.78399 21.78399 21.78399 0.11310696   0.0
643139 21.62803 21.62801 21.62809 21.62803 0.07155299 21.62810 21.50560 21.43475 21.50560 0.11310696   0.1
701229 21.50559 21.50557 21.50566 21.50559 0.07155299 21.43456 21.62804 21.62819 21.62804 0.11310696  -0.1
128174 21.43441 21.43437 21.43454 21.43441 0.07155299 18.45148 18.63286 18.47317 18.47317 0.05467501   2.9
       pooled.std.dev abs.z.stats p.adj.adjp.rawp p.adj.adjp.BH p.adj.index    z.real
520680     0.09682188    0.000000       1.0000000     1.0000000       17602  0.000000
643139     0.09682188    1.264497       0.2060517     0.2422529       18865  1.264497
701229     0.09682188    1.264658       0.2059938     0.2421976        1972  1.264658
128174     0.06514452   45.456439       0.0000000     0.0000000        8984 45.456439
```

To selectively view the result

```
> aLPE.result[1:5, c("median.1", "median.2", "median.diff", "p.adj.adjp.rawp", "p.adj.adjp.BH")]
```

```
       median.1 median.2 median.diff p.adj.adjp.rawp p.adj.adjp.BH
520680 21.78399 21.78399   0.0000000       1.0000000     1.0000000
643139 21.62803 21.50560   0.1224310       0.2060517     0.2422529
701229 21.50559 21.62804  -0.1224466       0.2059938     0.2421976
128174 21.43441 18.47317   2.9612377       0.0000000     0.0000000
703444 21.34799 19.30107   2.0469234       0.0000000     0.0000000
```

8 Creating Venn Diagram: doVennDiagram

Venn Diagrams are frequently used for examining similarities and differences in gene lists generated from different analysis. It is made up of two or more overlapping circles. The functionality of the *doVennDiagram* in *ABarray* relies on some functions provided in *limma* package.

8.1 Required Data

doVennDiagram accepts two or three vectors of gene lists. If two gene lists are provided, two way Venn diagram is produced. If three gene lists are provided, three way Venn diagram is produced.

- list1. The first vector of gene list.
- list2. The second vector of gene list.
- list3. Optional third vector of gene list.
- names. The names for use for each list in diagram.

8.2 doVennDiagram Function Demonstration

We will demonstrate two-way and three-way Venn diagram by creating three gene lists. Note: the result of joined lists are automatically saved into file.

- list1 contains probe names from index 1 to 100
- list2 contains probe names from index 61 to 160
- list3 contains probe names from index 81 to 200

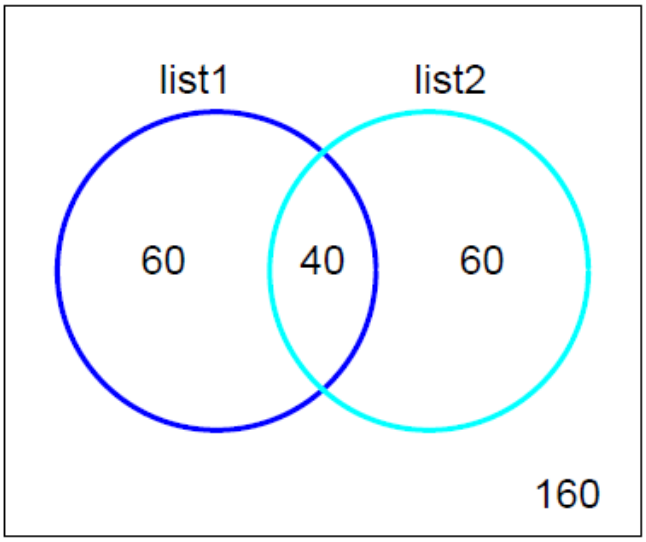
After these three gene lists are created, we perform *doVennDiagram*.

```
> list1 <- geneNames(eset)[1:100]
> list2 <- geneNames(eset)[61:160]
> list3 <- geneNames(eset)[81:200]
> length(list1)

[1] 100

> length(list2)

[1] 100

> length(list3)

[1] 120
```

8.2.1 Two Way Venn Diagram

There are 40 probes common between `list1` and `list2`. Let's create Venn diagram between `list1` and `list2`.

```
> doVennDiagram(list1, list2)

[1] "List information is written to file Venn_list_1+2.csv"
```



8.2.2 Three way Venn Diagram

There are 40 probes common between `list1` and `list2`, and 20 probes between `list1` and `list3`, and 80 probes between `list2` and `list3`. Let's create Venn diagram between `list1` and `list2`.

```
> doVennDiagram(list1, list2, list3)
```

```
[1] "List information is written to file Venn_list_1+2+3.csv"
```