Standard operation procedure for Exiqon miRNA microarray data analysis

Written by Ji-Hoon Cho at Institute for Systems Biology on 11.21.2013
Last modified and added by Taek-Kyun Kim on 06.19.2014

Checklist before starting analysis

- Information of experiment design
- Sample annotation such as sample identifier, batch information, phenotype, etc.
- Software, e.g. R, MATLAB, etc.

Input data

- Raw Exiqon microarray data - .txt file format, output from Agilent Feature Extraction
- Array annotation file – either .gal file provided from the manufacturer or .txt file obtained from GEO (Gene Expression Omnibus) database
- Sample annotation – recommend to store in a spreadsheet (.txt or .xls)

Analysis steps

1. Quality check

- RNA quality check using Agilent Bioanalyzer
- Generate a QC (quality control) report when scanning a microarray image using Agilent Feature Extraction software

2. Preprocess the microarray data

- One-channel (one-color) microarray
    1) No need to do *within-array normalization*
    2) Take 'gMeanSignal' from each file (= each array = each sample) and attach 'gMeanSignal's of all files (i.e. samples) to generate a data matrix of size, [# probes × # samples]. If a user wants to use a background-corrected signal, one can use 'gProcessedSignal' instead.
    3) Perform *between-array normalization* using quantile normalization approach
    4) If necessary, import gene annotations from the annotation file.
    5) Summarize microRNA signals by taking median of the replicated probes and generate the processed data matrix of size [# miRNA × # samples]

- Two-channel (two-color) microarray – two dyes are exclusively used (e.g. cy3 for only disease samples and cy5 for only control samples).
    1) Perform *within-array normalization* to remove dye-bias
        i. If a user decides to use 'gProcessedSignal' and 'rProcessedSignal' (dye-bias is already corrected in Feature Extraction software using LOESS method) from each file for cy3- and cy5-channel signals, no need to any action.

ii. Otherwise, one can take 'gMeanSignal' and 'rMeanSignal' for cy3- and cy5-channel signals and perform (customized) LOESS normalization

2) Calculate log-ratio in each file
   i. One can either simply take 'LogRatio' from each file, where LogRatio = $\log_{10}$(rProcessedSignal/gProcessedSignal), or calculate log-ratio using *within-array normalized* signals.
   ii. If necessary, convert $\log_{10}$-ratio into $\log_2$-ratio.
3) Perform *between-array normalization* using quantile normalization approach
4) If necessary, import gene annotations from the annotation file.
5) Summarize microRNA signals by taking median of the replicated probes and generate the processed data matrix of size [# miRNA × # samples]

- Two-channel (two-color) microarray – two dyes are interchangeably used (e.g. cy3 and cy5 for disease and control samples).
  1) The preprocessing procedure is somewhat similar to the case of one-channel microarray.
  2) Perform *within-array normalization* to remove dye-bias by taking 'gProcessedSignal' and 'rProcessedSignal' (dye-bias corrected signals) from each file
  3) Generate a data matrix of size, [# probes × # samples] by attaching 'gProcessedSignal' and 'rProcessedSignal' of all files (i.e. samples)
  4) Perform *between-array normalization* using quantile normalization approach
  5) If necessary, import gene annotations from the annotation file.
  6) Summarize microRNA signals by taking median of the replicated probes and generate the processed data matrix of size [# miRNA × # samples]

3. Post-normalization quality check

- Calculate correlation coefficients among all samples to figure out whether there is any outliers showing poor correlation with other samples and whether the samples in a specific group (e.g. control samples) are well-correlated.
- Check all pairwise scatter plots among samples

4. (Optional) Batch effect adjustment

- If a user has to combine two (or more) datasets produced from different batches, the adjustment of a potential batch effect is highly recommended.
- Check the existence of a batch effect using principal component analysis (PCA) and/or hierarchical clustering of samples across batches
- Adjustment of a batch effect using ComBat method

5. Statistical testing to identify differentially expressed miRNAs

- Moderated t-test using LIMMA (Bioconductor package in R environment)

- FDR (false discovery rate, usually Benjamini-Hochberg corrected) computation
- (Optional) Present/absent call – probes with sample-wise average intensity greater than global average intensity are considered to be *present* (i.e. reliably expressed)
- Identification of differentially expressed miRNAs using threshold cutoffs to p-value and/or FDR and/or fold-change and/or present/absent call
- Volcano plot, p-values (or FDRs) as a function of fold-changes may be helpful to quickly understand the miRNA expression changes in the dataset

6. Feature (gene) selection approach for biomarker discovery

- N times of K-fold cross validation (N×K times in total) by four different classification methods (two linear and two non-linear classifiers)
    i. Nearest Shrunken Centroid (PAM; linear)
    ii. LASSO-panelized logistic regression (LASSO; linear)
    iii. Random Forest with feature selection (RF; non-linear)
    iv. Recursive Feature Elimination combined with support vector machine (SVM-RFE; non-linear)
- Selection of miRNAs used more than (N×K)/2 cross validations (50%) for each method
- Selection of informative miRNAs
    i. commonly used in various classification methods
    ii. differentially expressed between PTSD patients and controls
    iii. amplifiable by specific primer

7. Diagnostic modeling

- Use of the classification method showing the lowest cross validation error at the previous step. In our study, SVM-RFE (kernlab; Bioconductor package in R) was adopted.
- Identification of optimal panel of miRNAs using SVM-RFE with
    i. C-svc algorithm
    ii. radial basis kernel function
    iii. automatically determined hyperparameter σ and manually selected marginal parameter C (C = 0.1, 1 and 10)
- Computation of performances through K-fold cross validation with training set composed of the identified optimal set of miRNAs (Output scores of samples representing probability for being PTSD positive or negative will be produced from the model)
- Generation of confusion matrix (True Negative, True Positive, False Negative and False Positive) and Receiver Operating Characteristic curve using ROCR (Bioconductor packages in R)

Standard operation procedure for Exiqon high-throughput miRNA qPCR data analysis

Checklist before starting analysis

- Information of experiment design
- Sample annotation such as sample identifier, batch information, phenotype, etc.
- Software, e.g. R, MATLAB, etc.

Input data

- Raw real-time PCR data – .txt file format, output from SDS software
- Sample annotation – recommend to store in a spreadsheet (.txt or .xls)

Analysis steps

1. Quality check

- RNA quality check using Agilent Bioanalyzer

2. Preprocess the qPCR data

- Attach output files (txt files) of all samples and generate a raw data matrix of size, [# probes × # samples]
- Normalization of qPCR data using the approaches recommended by manufacturer (interplate calibration and global mean normalization, shown in http://www.exiqon.com/ls/Documents/Scientific/miRNA-qPCR-guidelines.pdf, http://www.exiqon.com/ls/Documents/Scientific/Exiqon-data-analysis-guide.pdf and Mestdagh et al. *Genome Biology* 2009, 10:R64)

3. (Optional) Batch effect adjustment

- If a user has to combine two (or more) datasets produced from different batches, the adjustment of a potential batch effect is highly recommended.
- Adjustment of a batch effect using ComBat method
- For the proper running of ComBat (and the following missing value imputation), probes with missing values in more than 50% of samples should be removed in advance.

4. Missing value imputation

- While qPCR data usually has several *undetermined* values, many downstream analyses require a complete dataset without any missing values.
- A Bioconductor package, IMPUTE (*k*-nearest neighbor imputation method) can be used to impute missing values.
- For a proper running of IMPUTE, the probes with missing values in more than 50% of samples are removed in advance.
- Missing value imputation using IMPUTE with default parameters
- Users can take other approaches; e.g. ignore the probes with any missing values, replace missing values with a constant value, etc.

5. Post-normalization quality check

- Calculate correlation coefficients among all samples to figure out whether there is any outliers showing poor correlation with other samples and whether the samples in a specific group (e.g. control samples) are well-correlated.
- Check all pairwise scatter plots among samples

6. Statistical testing to identify differentially expressed miRNAs

- If necessary, convert Ct values into *C*-Ct, where *C* is a constant value (e.g. 40 or 45), to make a large intensity represents a high abundance.
- Moderated t-test using LIMMA (Bioconductor package in R environment)
- FDR (false discovery rate, usually Benjamini-Hochberg corrected) computation
- (Optional) Present/absent call – probes with sample-wise average intensity greater than global average intensity are considered to be *present* (i.e. reliably expressed)
- Identification of differentially expressed miRNAs using threshold cutoffs to p-value and/or FDR and/or fold-change and/or present/absent call
- Volcano plot, p-values (or FDRs) as a function of fold-changes may be helpful to quickly understand the miRNA expression changes in the dataset

7. Feature (gene) selection approach for biomarker discovery

- N times of K-fold cross validation (N×K times in total) by four different classification methods (two linear and two non-linear classifiers)
  i. Nearest Shrunken Centroid (PAM; linear)
  ii. LASSO-panelized logistic regression (LASSO; linear)
  iii. Random Forest with feature selection (RF; non-linear)
  iv. Recursive Feature Elimination combined with support vector machine (SVM-RFE; non-linear)
- Selection of miRNAs used more than (N×K)/2 cross validations (50%) for each method
- Selection of informative miRNAs
  i. commonly used in various classification methods
  ii. differentially expressed between PTSD patients and controls
  iii. amplifiable by specific primer

8. Diagnostic modeling

- Use of the classification method showing the lowest cross validation error at the previous step. In our study, SVM-RFE (kernlab; Bioconductor package in R) was adopted.
- Identification of optimal panel of miRNAs using SVM-RFE with
  i. C-svc algorithm
  ii. radial basis kernel function
  iii. automatically determined hyperparameter σ and manually selected marginal parameter C (C = 0.1, 1 and 10)
- Computation of performances through K-fold cross validation with training set composed of the identified optimal set of miRNAs (Output scores of

samples representing probability for being PTSD positive or negative will be produced from the model)

- Generation of confusion matrix (True Negative, True Positive, False Negative and False Positive) and Receiver Operating Characteristic curve using ROCR (Bioconductor packages in R)

9. Example script codes

```
## Missing value implementation
library(impute)
data                                                              =
read.table("Example.txt",header=T,sep="\t",check.names=F)
exprs = data[,2:ncol(data)]
nanrow = (rowSums(is.na(exprs))>floor(0.2*ncol(exprs)))
exprs2 = exprs[!nanrow,]
imputed = impute.knn(as.matrix(exprs2))
imputed = data.frame(imputed$data)

## LIMMA analysis
library(limma)
design            <-            cbind(CTR=c(1,1,1,1,0,0,0,0),
PTSD=c(0,0,0,0,1,1,1,1)); # example;
fit<-lmFit(imputed,design);
cont.matrix       <-       makeContrasts(CTRvsPTSD=PTSD-CTR,
levels=design);
fit <- contrasts.fit(fit, cont.matrix);
fit <- eBayes(fit);
result<-topTable(fit,n=nrow(imputed),adjust="BH",
sort="none")

##  Recursive  Feature  Elimination  with  Support  Vector
Machine
library('kernlab')
status                                                            =
read.table("ExampleStatus.txt",header=T,sep="\t",check.name
s=F)
x = t(imputed)
y = status
xrfe = x
folds = length(y)
i = 0
cverr = list()
while (ncol(xrfe)>1){
  i = i+1
  model = ksvm(xrfe, y, C=1, type="C-svc", kernel="rbfdot",
shrinking=F)
  cv  =  ksvm(xrfe,  y,  C=1,  type="C-svc",  kernel="rbfdot",
shrinking=F, cross=length(y))
```

```
  cverr[i] = cross(cv)
  svs = unlist(alphaindex(model))
  weight = abs(t(unlist(coef(model)))%*%xrfe[svs,])
  remove = colnames(xrfe)[order(weight)[1]]
  xrfe = xrfe[,setdiff(colnames(xrfe),remove),drop=F]
}
```