

Key Aspects of the Design & Analysis of DNA Microarray Studies

Richard Simon, D.Sc.
Chief, Biometric Research Branch
National Cancer Institute
<http://linus.nci.nih.gov/brb>

<http://linus.nci.nih.gov/brb>

- Powerpoint presentation
- Bibliography
 - Publications providing details and proofs of assertions
- Reprints & Technical Reports
- BRB-ArrayTools software
 - Performs all analyses described

- *Design and Analysis of DNA Microarray Investigations*

- R Simon, EL Korn, MD Radmacher, L McShane, G Wright, Y Zhao. Springer (2003)

Myth

- That microarray investigations should be unstructured data-mining adventures without clear objectives

- Good microarray studies have clear objectives, but not generally gene specific mechanistic hypotheses
- Design and analysis methods should be tailored to study objectives

Common Types of Objectives

- Class Comparison
 - Identify genes differentially expressed among predefined classes
 - tissue types
 - experimental groups
 - Response groups
 - Prognostic groups
- Class Prediction
 - Develop multi-gene predictor of class for a sample using its gene expression profile
- Class Discovery
 - Discover clusters among specimens or among genes

Do Expression Profiles Differ for Two Defined Classes of Arrays?

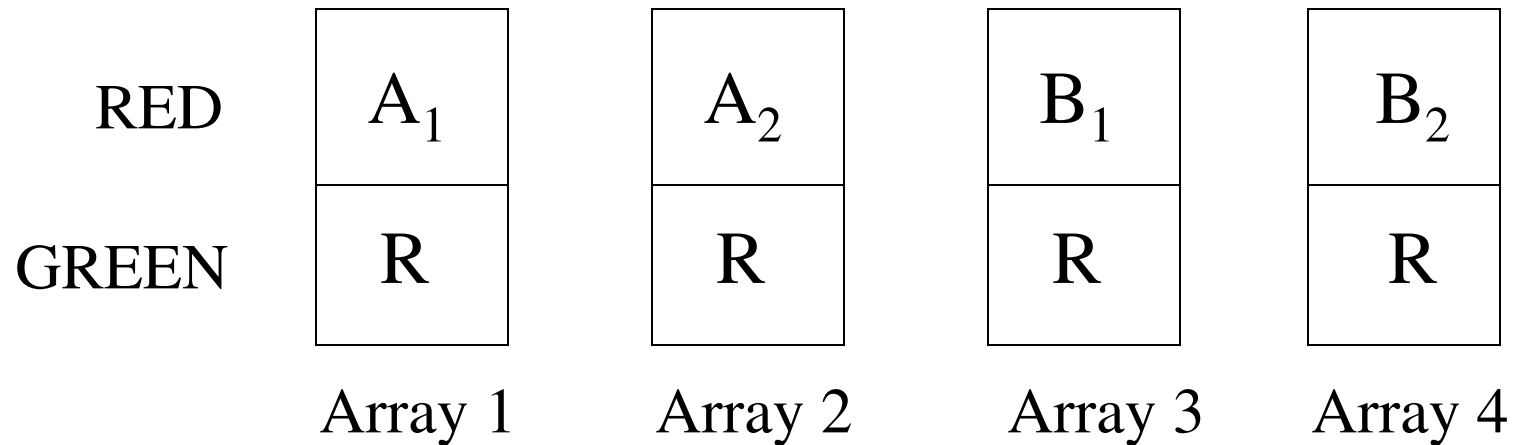
- Not a clustering problem
 - Supervised methods
- Generally requires multiple biological samples from each class

Levels of Replication

- Technical replicates
 - RNA sample divided into multiple aliquots and re-arrayed
- “Biological” replicates
 - Multiple subjects
 - Replication of the tissue culture experiment

- Biological conclusions generally require independent biological replicates. The power of statistical methods for microarray data depends on the number of biological replicates.
- Technical replicates are useful insurance to ensure that at least one good quality array of each specimen will be obtained.
- Some of the microarray experimental design literature is applicable only to experiments without biological replication

Common Reference Design



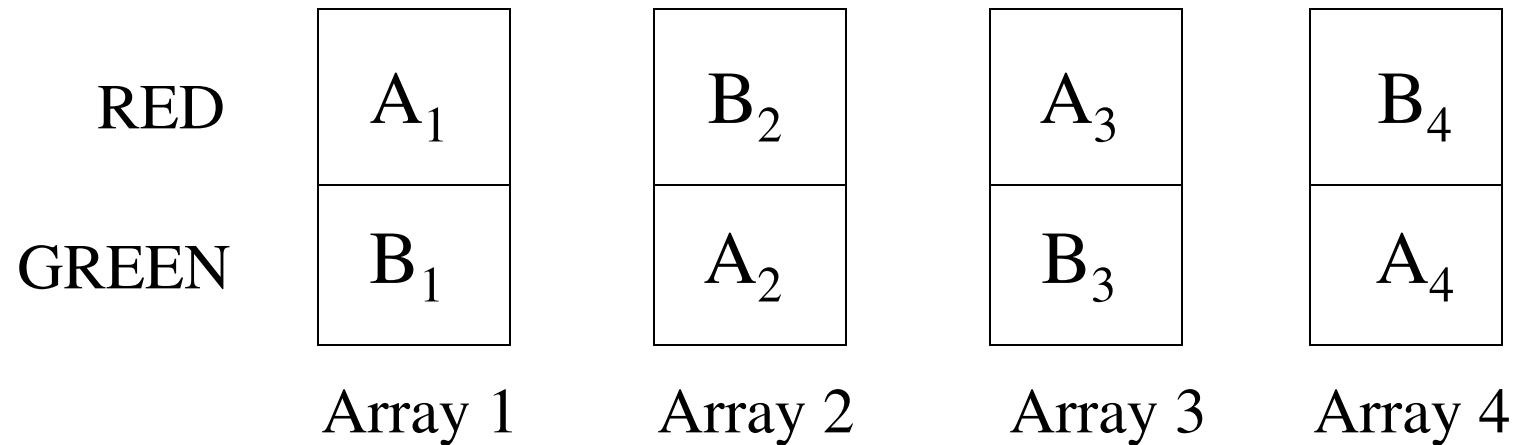
A_i = *i*th specimen from class A

B_i = *i*th specimen from class B

R = aliquot from reference pool

- The reference generally serves to control variation in the size of corresponding spots on different arrays and variation in sample distribution over the slide.
- The reference provides a relative measure of expression for a given gene in a given sample that is less variable than an absolute measure.
- The reference is not the object of comparison.
- The relative measure of expression will be compared among biologically independent samples from different classes.

Balanced Block Design



A_i = i th specimen from class A

B_i = i th specimen from class B

- Detailed comparisons of the effectiveness of designs:
 - Dobbin K, Simon R. Comparison of microarray designs for class comparison and class discovery. *Bioinformatics* 18:1462-9, 2002
 - Dobbin K, Shih J, Simon R. Statistical design of reverse dye microarrays. *Bioinformatics* 19:803-10, 2003
 - Dobbin K, Simon R. Questions and answers on the design of dual-label microarrays for identifying differentially expressed genes, *JNCI* 95:1362-1369, 2003

- Common reference designs are very effective for many microarray studies. They are robust, permit comparisons among separate experiments, and permit many types of comparisons and analyses to be performed.
- For simple two class comparison problems, balanced block designs require many fewer arrays than common reference designs.
 - Efficiency decreases for more than two classes
 - Are more difficult to apply to more complicated class comparison problems.
 - They are not appropriate for class discovery or class prediction.
- Loop designs can be useful for multi-class comparison problems, but are not-robust to bad arrays and are not suitable for class prediction or class discovery.

Myth

- For two color microarrays, each sample of interest should be labeled once with Cy3 and once with Cy5 in dye-swap pairs of arrays.

Dye Bias

- Average differences among dyes in label concentration, labeling efficiency, photon emission efficiency and photon detection are corrected by normalization procedures
- Gene specific dye bias may not be corrected by normalization

- Dye swap technical replicates of the same two rna samples are rarely necessary.
- Using a common reference design, dye swap arrays are not necessary for valid comparisons of classes since specimens labeled with different dyes are never compared.
- For two-label direct comparison designs for comparing two classes, it is more efficient to balance the dye-class assignments for independent biological specimens than to do dye swap technical replicates

Can I reduce the number of arrays by pooling specimens?

- Pooling all specimens is inadvisable because conclusions are limited to the specific RNA pool, not to the populations since there is no estimate of variation among pools
- With multiple biologically independent pools, some reduction in number of arrays may be possible but the reduction is generally modest and may be accompanied with a large increase in the number of independent biological specimens needed
 - Dobbin & Simon, Biostatistics (In Press).

Sample Size Planning

- GOAL: Identify genes differentially expressed in a comparison of two pre-defined classes of specimens on dual-label arrays using reference design or single label arrays
- Compare classes separately by gene with adjustment for multiple comparisons
- Approximate expression levels (log ratio or log signal) as normally distributed
- Determine number of samples and arrays to give power $1-\beta$ for detecting mean difference δ at level α

$$n = 4m \left[\frac{z_{\alpha/2} + z_{\beta}}{\delta} \right]^2 \left(\tau_g^2 / k + \gamma^2 / m \right)$$

- m = number of technical reps per sample
- k = number of samples per pool
- n = total number of arrays
- δ = mean difference between classes in log signal
- τ^2 = biological variance within class
- γ^2 = technical variance
- α = significance level e.g. 0.001
- $1-\beta$ = power
- z = normal percentiles (use t percentiles for better accuracy)

Number of samples pooled per array	Number of arrays required	Number of samples required
1	25	25
2	17	34
3	14	42
4	13	52

$\alpha=0.001, \beta=0.05, \delta=1, \tau^2+2\sigma^2=0.25, \tau^2/\sigma^2=4$

$$\alpha=0.001 \quad \beta=0.05 \quad \delta=1$$
$$\tau^2+2\gamma^2=0.25, \quad \tau^2/\gamma^2=4$$

m technical reps	n arrays required	samples required
1	25	25
2	42	21
3	60	20
4	76	19

Class Prediction

- Most statistical methods were developed for inference, not prediction.
- Most statistical methods for were not developed for $p \gg n$ settings

Components of Class Prediction

- Feature (gene) selection
 - Which genes will be included in the model
- Select model type
 - E.g. LDA, Nearest-Neighbor, ...
- Fitting parameters (regression coefficients) for model

Univariate Feature Selection

- Genes that are univariately differentially expressed among the classes at a significance level α (e.g. 0.01)
 - The α level is selected to control the number of genes in the model, not to control the false discovery rate
 - The accuracy of the significance test used for feature selection is not of major importance as identifying differentially expressed genes is not the ultimate objective

Linear Classifiers for Two Classes

$$l(\underline{x}) = \sum_{i \in F} w_i x_i$$

\underline{x} = vector of log ratios or log signals

F = features (genes) included in model

w_i = weight for i'th feature

decision boundary $l(\underline{x}) >$ or $<$ d

Linear Classifiers for Two Classes

- Fisher linear discriminant analysis
- Diagonal linear discriminant analysis (DLDA) assumes features are uncorrelated
 - Naïve Bayes classifier
- Compound covariate predictor (Radmacher et al.) and Golub's method are similar to DLDA in that they can be viewed as weighted voting of univariate classifiers

Linear Classifiers for Two Classes

- Support vector machines with inner product kernel are linear classifiers with weights determined to minimize errors subject to regularization condition
 - Can be written as finding hyperplane with separates the classes with a specified margin and minimizes length of weight vector
- Perceptrons are linear classifiers

When $p > n$

- For the linear model, an infinite number of weight vectors w can always be found that give zero classification errors for the training data.
 - $p \gg n$ problems are almost always linearly separable
- Why consider more complex models?

Myth

- That complex classification algorithms perform better than simpler methods for class prediction
 - Many comparative studies indicate that simpler methods work as well or better for microarray problems

Evaluating a Classifier

- Fit of a model to the same data used to develop it is no evidence of prediction accuracy for independent data

Split-Sample Evaluation

- Training-set
 - Used to select features, select model type, determine parameters and cut-off thresholds
- Test-set
 - Withheld until a *single* model is *fully* specified using the training-set.
 - Fully specified model is applied to the expression profiles in the test-set to predict class labels.
 - Number of errors is counted
 - Ideally test set data is from different centers than the training data and assayed at a different time

Leave-one-out Cross Validation

- Omit sample 1
 - Develop multivariate classifier from scratch on training set with sample 1 omitted
 - Predict class for sample 1 and record whether prediction is correct

Leave-one-out Cross Validation

- Repeat analysis for training sets with each single sample omitted one at a time
- e = number of misclassifications determined by cross-validation
- Subdivide e for estimation of sensitivity and specificity

- With proper cross-validation, the model must be developed *from scratch* for each leave-one-out training set. This means that feature selection must be repeated for each leave-one-out training set.
- The cross-validated estimate of misclassification error is an estimate of the prediction error for model fit using specified algorithm to full dataset
- If you use cross-validation estimates of prediction error for a set of algorithms indexed by a tuning parameter and select the algorithm with the smallest cv error estimate, you do not have a valid estimate of the prediction error for the selected model

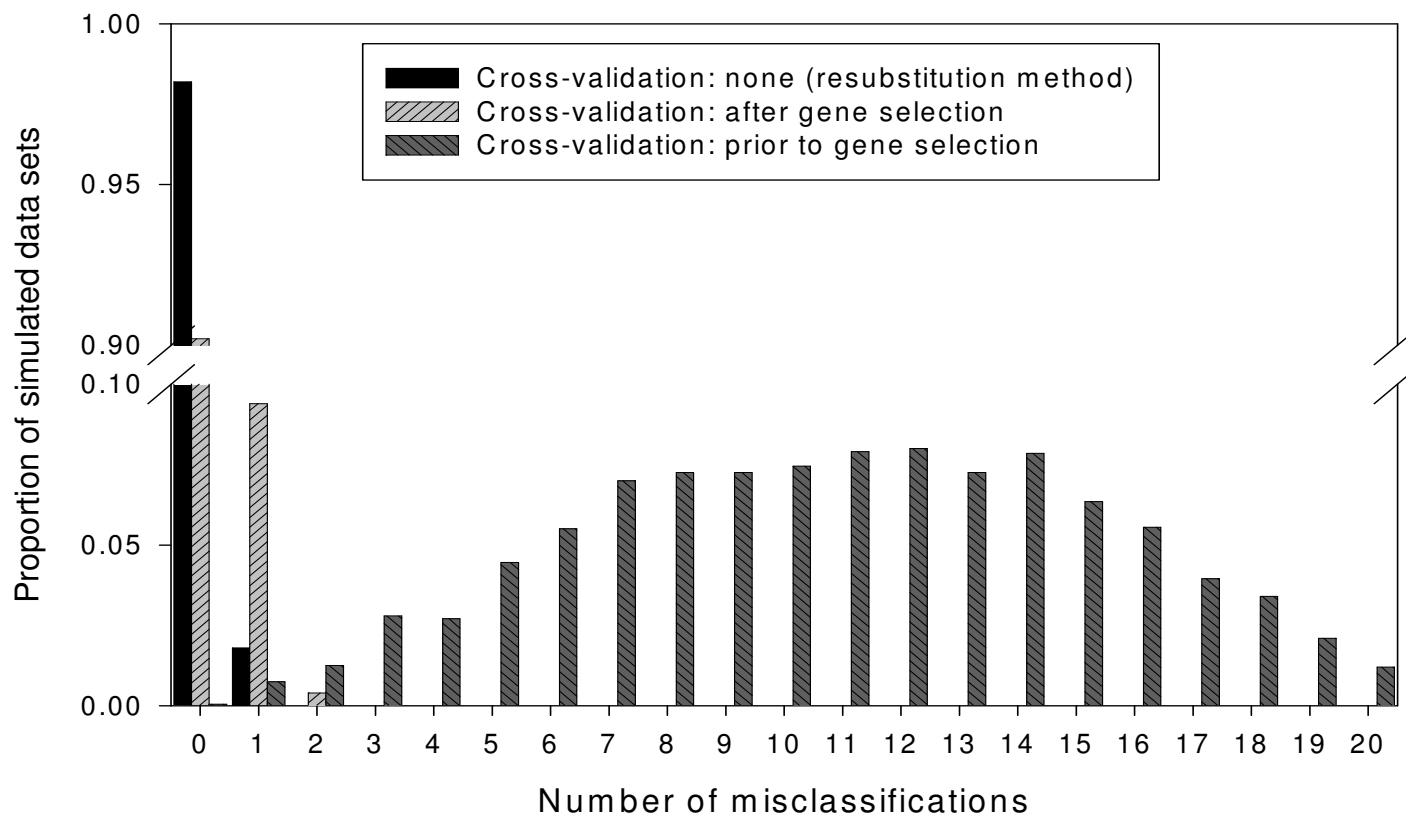
Prediction on Simulated Null Data

Generation of Gene Expression Profiles

- 14 specimens (P_i is the expression profile for specimen i)
- Log-ratio measurements on 6000 genes
- $P_i \sim \text{MVN}(\mathbf{0}, \mathbf{I}_{6000})$
- Can we distinguish between the first 7 specimens (Class 1) and the last 7 (Class 2)?

Prediction Method

- Compound covariate prediction
- Compound covariate built from the log-ratios of the 10 most differentially expressed genes.



Permutation Distribution of Cross-validated Misclassification Rate of a Multivariate Classifier

- Randomly permute class labels and repeat the entire cross-validation
- Re-do for all (or 1000) random permutations of class labels
- Permutation p value is fraction of random permutations that gave as few misclassifications as e in the real data

Invalid Criticisms of Cross-Validation

- “You can always find a set of features that will provide perfect prediction for the training and test sets.”
 - There is often many sets of features that provide zero training errors. Cross validation will provide an unbiased estimate of the generalization error for a specified algorithm that selects a specific model on a training set.

Cross-Validation

- Estimates prediction error for future data
 - For prediction using model developed using full current dataset
- Cross-validation is used to estimate prediction error of a defined algorithm, not as part of a model building algorithm
- If you use the results of cross-validation for model building, then a double nested cross-validation is needed to obtain a valid estimate of prediction error for the resulting model

Comparison of Internal Validation Methods

Molinaro, Pfiffer & Simon

- For small sample sizes, LOOCV is much more accurate than split-sample validation
 - Split sample validation is highly positively biased
- For small sample sizes, LOOCV is preferable to 10-fold, 5-fold cross-validation or repeated k-fold versions

BRB-ArrayTools

- Integrated software package using Excel-based user interface but state-of-the art analysis methods programmed in R, Java & Fortran
- Publicly available for non-commercial use

<http://linus.nci.nih.gov/brb>

Selected Features of BRB-ArrayTools

- Multivariate permutation tests for class comparison to control number and proportion of false discoveries with specified confidence level
 - Permits blocking by another variable, pairing of data, averaging of technical replicates
- SAM
 - Fortran implementation 7X faster than R versions
- Extensive annotation for identified genes
 - Internal annotation of NetAffx, Source, Gene Ontology, Pathway information
 - Links to annotations in genomic databases
- Find genes correlated with quantitative factor while controlling number or proportion of false discoveries
- Find genes correlated with censored survival while controlling number or proportion of false discoveries
- Analysis of variance

Selected Features of BRB-ArrayTools

- Gene enhancement analysis.
 - Find Gene Ontology groups and signaling pathways that are differentially expressed among classes
- Class prediction
 - DLDA, CCP, Nearest Neighbor, Nearest Centroid, Shrunk Centroids, SVM, Random Forests
 - Complete LOOCV, k-fold CV, repeated k-fold, .632 bootstrap
 - permutation significance of cross-validated error rate

Selected Features of BRB-ArrayTools

- Clustering tools for class discovery with reproducibility statistics on clusters
 - Internal access to Eisen's Cluster and Treeview
- Visualization tools including rotating 3D principal components plot exportable to Powerpoint with rotation controls
- Extensible via R plug-in feature
- Tutorials and datasets

Acknowledgements

- Kevin Dobbin
- Ed Korn
- Amy Peng Lam
- Lisa McShane
- Michael Radmacher
- Sudhir Varma
- George Wright
- Yingdong Zhao