BRB-ArrayTools Workshop

• Overview of gene expression analysis (2hr)

• Individual consultation as needed
  – Biometric Research Branch statisticians
  – BRB-ArrayTools Development Team
http://linus.nci.nih.gov/brb

- http://linus.nci.nih.gov/brb
  - Powerpoint presentations and audio files
  - Reprints & Technical Reports
  - BRB-ArrayTools software
  - BRB-ArrayTools Data Archive
Assumptions

- You are somewhat familiar with BRB-ArrayTools
- You have brought your own laptop
- You have installed BRB-ArrayTools
- You have imported (collated) your data into BRB-ArrayTools
BRB-ArrayTools 3.5 alpha

• Available on cd for you to try if you’d like

• Contains
  – Data import wizzard
  – Data analysis wizzard
  – Enhanced survival risk-group prediction tool
Take Time to Clarify Your Specific Objectives

• Study Design
• Analysis Strategy
Good Microarray Studies Have Clear Objectives

• Class Comparison
  – Find genes whose expression differs among predetermined classes

• Class Prediction
  – Prediction of predetermined class (phenotype) using information from gene expression profile

• Class Discovery
  – Discover clusters of specimens having similar expression profiles
  – Discover clusters of genes having similar expression profiles
Class Comparison and Class Prediction

• Not clustering problems
  – Global similarity measures generally used for clustering arrays may not distinguish classes
  – Don’t control multiplicity or for distinguishing data used for classifier development from data used for classifier evaluation

• Supervised methods

• 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.
Microarray Platforms for Developing Predictive Classifiers

• Single label arrays
  – Affymetrix GeneChips

• Dual label arrays
  – Common reference design
  – Other designs
Common Reference Design

A_i = \text{ith specimen from class A}
B_i = \text{ith specimen from class B}
R = \text{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

RED

GREEN

Array 1

Array 2

Array 3

Array 4

\[ A_i = \text{ith specimen from class A} \]

\[ B_i = \text{ith specimen from class B} \]
This procedure finds genes differentially expressed among classes of samples. The classes are pre-defined based on columns of the experiment descriptor file. Each array should represent one sample, either as a single-label experiment or as a dual-label experiment using a common reference. For non-reference designs, consider using the tool for class comparison between red and green samples.

**Experimental design:**
- Column defining classes:
- Unpaired samples:
  - Block by:
  - Average over replicates of:
- Paired samples:
  - Pair samples by:

**Find gene lists determined by:**
- Significance threshold of univariate tests: 0.001
- Restriction on proportion of false discoveries:
  - Maximum proportion of false discoveries: 0.1
  - Confidence level (between 0 and 100%): 80
- Restriction on number of false discoveries:
  - Maximum number of false discoveries: 10
  - Confidence level (between 0 and 100%): 80

**Variance model:**
- Use random variance model for univariate tests.

NOTE: This analysis is currently set to run on all genes passing the filter.

Select gene subsets
Class Comparison Blocking

• Paired data
  – Pre-treatment and post-treatment samples of same patient
  – Tumor and normal tissue from the same patient

• Blocking
  – Multiple animals in same litter
  – Any feature thought to influence gene expression
    • Sex of patient
    • Batch of arrays
Technical Replicates

• Multiple arrays on alloquots of the same RNA sample
• Select the best quality technical replicate or
• Average expression values
Simple Control for Multiple Testing

• If each gene is tested for significance at level \( \alpha \) and there are \( n \) genes, then the expected number of false discoveries is \( n \alpha \).
  – e.g. if \( n=1000 \) and \( \alpha=0.001 \), then 1 false discovery
  – To control \( E(FD) \leq u \)
  – Conduct each of \( k \) tests at level \( \alpha = u/k \)
False Discovery Rate (FDR)

• FDR = *Expected* proportion of false discoveries among the tests declared significant

• Studied by Benjamini and Hochberg (1995):
<table>
<thead>
<tr>
<th></th>
<th>Not rejected</th>
<th>Rejected</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>True null hypotheses</td>
<td>890</td>
<td>10 True false discoveries</td>
<td>900</td>
</tr>
<tr>
<td>False null hypotheses</td>
<td>10</td>
<td>90 True discoveries</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1000</td>
</tr>
</tbody>
</table>
If you analyze n probe sets and select as “significant” the k genes whose $p \leq p^*$

- FDR $\sim n \frac{p^*}{k}$
Limitations of Simple Procedures

• p values based on normal theory are not accurate in the extreme tails of the distribution
• Difficult to achieve extreme quantiles for permutation $p$ values of individual genes
• Multiple comparisons controlled by adjustment of univariate (single gene) $p$ values may not take advantage of correlation among genes
Additional Procedures

- **“SAM”** - Significance Analysis of Microarrays
  - Tusher et al., *PNAS*, 2001
  - Estimate FDR
  - Statistical properties unclear

- **Multivariate permutation tests**
  - Korn et al., 2001 (http://linus.nci.nih.gov/brb)
  - Control number or proportion of false discoveries
  - Can specify confidence level of control
Multivariate Permutation Procedures
(Korn et al., 2001)

Allows statements like:

**FD Procedure**: We are 95% confident that the (actual) number of false discoveries is no greater than 5.

**FDP Procedure**: We are 95% confident that the (actual) proportion of false discoveries does not exceed 0.10.
t-test Comparisons of Gene Expression for gene j

- $x_j \sim N(\mu_{j1}, \sigma_j^2)$ for class 1
- $x_j \sim N(\mu_{j2}, \sigma_j^2)$ for class 2

- $H_{0j}: \mu_{j1} = \mu_{j2}$
Estimation of Within-Class Variance

• Estimate separately for each gene
  – Limited degrees-of-freedom (precision) unless number of samples is large
  – Gene list dominated by genes with small fold changes and small variances

• Assume all genes have same variance
  – Poor assumption

• Random (hierarchical) variance model
  – Variances are independent samples from a common distribution; Inverse gamma distribution used
  – Results in exact F (or t) distribution of test statistics with increased degrees of freedom for error variance
  – For any normal linear model
This procedure finds genes differentially expressed among classes of samples. The classes are pre-defined based on columns of the experiment descriptor file. Each array should represent one sample, either as a single-label experiment or as a dual-label experiment using a common reference. For non-reference designs, consider using the tool for class comparison between red and green samples.

**Class Comparison Options**

- **Perform univariate permutation tests:**
  - Number of permutations for univariate test: 10000

- **Perform GO Observed vs. Expected analysis.**

- **Name to use for output files:**
  - ClassComparison

- **Unpaired samples**
  - Block by:
    - Number of permutations for multivariate test: 1000

- **Paired samples**
  - Pair samples by:

- **Options**
  - Set to run on all genes passing the filter.
**Significance Analysis of Microarrays (SAM)**

SAM finds genes differentially expressed among classes of samples. The classes are pre-defined based on columns of the experiment descriptor file.

### Experimental design:
- **Column defining classes:**

### Parameters:
- **Target proportion of false discoveries:** 0.1
- **Number of Permutations:** 100
- **Percentile:** 80
- **Perform Gene Ontology Observed vs. Expected analysis**
- **Name to use for output files:** SAM

**NOTE:** This analysis is currently set to run on all genes passing the filter.

**descriptors**
- Filtered log ratio
- Gene annotations
- Gene identifiers
The "Gene Ontology" option finds Gene Ontology categories that have higher than expected number of genes differentially expressed among classes of samples. The number of comparisons is the number of GO categories and hence the multiple testing problem is reduced. The "Pathway" option finds pathways that have higher than expected number of genes differentially expressed among classes of samples. The number of comparisons is the number of pathways represented in the dataset and hence the multiple testing problem is reduced. The "User gene lists" option finds Gene Lists that have higher than expected number of genes differentially expressed among classes of samples. All classes are pre-defined based on columns of the experiment descriptor file.

**Experimental design:**
- Column defining classes:
- Unpaired samples:
  - Average over replicates of:
- Paired samples:
  - Pair samples by:

**Gene set determined by:**
- Gene Ontology
- Pathways
- User gene lists

**Find pathway lists determined by:**
- Human:
  - BoCarta Pathways
  - KEGG Pathways
  - Broad/MIT Pathways and signatures
- Mouse:
  - BoCarta Pathways

**Variance model:**
- Use random variance model for univariate tests.

**Name to use for output files:**
- GOComparison

**Significance threshold of permutation tests:**
- 0.005

NOTE: This analysis is currently set to run on all genes passing the filter.
Gene Set Expression Comparison

• Compute p value of differential expression for each gene in a gene set (k=number of genes)
• Compute a summary (S) of these p values
• Determine whether the value of the summary test statistic S is more extreme than would be expected from a random sample of k genes (probe-sets) on that platform
• Two types of summaries provided
  – Average of log p values
  – Kolmogorov-Smirnov statistic; largest distance between the cumulative distribution of the p values and the uniform distribution expected if none of the genes were differentially expressed
Gene Set Expression Comparison

• p value for significance of summary statistic need not be as extreme as .001 usually, because the number of gene sets analyzed is usually much less than the number of individual genes analyzed

• Conclusions of significance are for gene sets in this tool, not for individual genes
Comparison of Gene Set Expression Comparison to O/E Analysis in Class Comparison

• Gene set expression tool is based on all genes in a set, not just on those significant at some threshold value

• O/E analysis does not provide statistical significance for gene sets
Class comparison between red and green samples

This tool is for finding genes differentially expressed among two classes for dual-label arrays in which each array contains a sample from one class and a sample from the other class. The samples from one class need not be labeled with the same label on all arrays; generally it is best to have complete balance of labels and class. This tool requires that each biological sample appear either on only one array or else always paired with the same sample from the other class. As a special case, this tool allows to compare samples of one class with the reference samples. In this case, reference samples should contain the key word 'reference' in the Red-labeled or Green-labeled sample ID column of the Experiment Descriptors worksheet.

Experimental design:
- Red-labeled sample ID column:
- Green-labeled sample ID column:
- Red-labeled sample class column:
- Green-labeled sample class column:

Find gene lists determined by:
- Significance threshold of univariate tests: 0.001
- Restriction on proportion of false discoveries:
  - Maximum proportion of false discoveries: 0.1
  - Confidence level (between 0 and 100%): 80
- Restriction on number of false discoveries:
  - Maximum number of false discoveries: 10
  - Confidence level (between 0 and 100%): 80

Variance model:
- Use random variance model for univariate tests.

NOTE: This analysis is currently set to run on all genes passing the filter.

Select gene subsets
## Quantitative Trait Analysis

This tool finds genes that are significantly correlated with a specified quantitative variable (trait).

### Experimental design:
- **Quantitative trait column:** [Dropdown]
- **Use Spearman Correlation Test**
- **Use Pearson Correlation Test**
- **Average over replicates of:** [Dropdown]

### Find gene lists determined by:
- **Significance threshold of univariate tests:** 0.001
- **Restriction on proportion of false discoveries:**
  - Maximum proportion of false discoveries: 0.1
  - Confidence level (between 0 and 100%): 80
- **Restriction on number of false discoveries:**
  - Maximum number of false discoveries: 10
  - Confidence level (between 0 and 100%): 80

**NOTE:** This analysis is currently set to run on all genes passing the filter.

[Select gene subsets]

### Descriptors:
- Filtered log ratio
- Gene annotations
- Gene identifiers
Find Genes Correlated with Survival

This procedure tests for genes which are significantly associated with survival.

**Experimental design:**
- Status column: (0 - censored, 1 - death)
- Column defining survival time:

**Find gene lists determined by:**
- Significance threshold of univariate tests: 0.001
- Restriction on proportion of false discoveries:
  - Maximum proportion of false discoveries: 0.1
  - Confidence level (between 0 and 100%): 80
- Restriction on number of false discoveries:
  - Maximum number of false discoveries: 10
  - Confidence level (between 0 and 100%): 80

NOTE: This analysis is currently set to run on all genes passing the filter.

Select gene subsets

OK  Cancel  Options  Reset  Help
Statistical Methods Appropriate for Prediction are Different than Those Appropriate for Gene Finding

- Demonstrating statistical significance of prognostic factors is not the same as demonstrating predictive accuracy.
- Demonstrating goodness of fit of a model to the data used to develop it is not a demonstration of predictive accuracy.
- Statisticians are used to inference, not prediction
- Most statistical methods were not developed for $p \gg n$ prediction problems
Components of Class Prediction

• Feature (gene) selection
  – Which genes will be included in the model

• Select model type
  – E.g. Diagonal linear discriminant analysis, Nearest-Neighbor, ...

• Fitting parameters (regression coefficients) for model
  – Selecting value of tuning parameters
Feature Selection

• Genes that are differentially expressed among the classes at a significance level $\alpha$ (e.g. 0.01)
  – The $\alpha$ level is selected only to control the number of genes in the model
Feature Selection

• Small subset of genes which together give most accurate predictions
  – Combinatorial optimization algorithms
    • Genetic algorithms

• Little evidence that complex feature selection is useful in microarray problems
  – Failure to compare to simpler methods
  – Some published complex methods for selecting combinations of features do not appear to have been properly evaluated
Linear Classifiers for Two Classes

\[ l(x) = \sum_{i \in F} w_i x_i \]

\( x \) = vector of log ratios or log signals
\( F \) = features (genes) included in model
\( w_i \) = weight for i'th feature
decision boundary \( l(x) > \) or \( < d \)
Linear Classifiers for Two Classes

- Fisher linear discriminant analysis
  - Requires estimating correlations among all genes selected for model
- Diagonal linear discriminant analysis (DLDA) assumes gene expressions are uncorrelated
- Compound covariate predictor (Radmacher) 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

- Compound covariate predictor

\[ w_i \propto \frac{\bar{x}_i^{(1)} - \bar{x}_i^{(2)}}{\hat{\sigma}_i} \]

Instead of for DLDA

\[ w_i \propto \frac{\bar{x}_i^{(1)} - \bar{x}_i^{(2)}}{\hat{\sigma}_i^2} \]
Linear Classifiers for Two Classes

• Support vector machines with inner product kernel are linear classifiers with weights determined to separate the classes with a hyperplane that minimizes the length of the weight vector.
Support Vector Machine

\[
\text{minimize } \sum_i w_i^2 \\
\text{subject to } y_j \left( w' x^{(j)} + b \right) \geq 1 \\
\text{where } y_j = \pm 1 \text{ for class 1 or 2.}
\]
When $p \gg n$

- It is always possible to find a set of features and a weight vector for which the classification error on the training set is zero.
- Why consider more complex models?
Myth

• Complex classification algorithms such as neural networks perform better than simpler methods for class prediction.
• Artificial intelligence sells to journal reviewers and peers who cannot distinguish hype from substance when it comes to microarray data analysis.

• Comparative studies have shown that simpler methods work as well or better for microarray problems because they avoid overfitting the data.
Other Simple Methods

- Nearest neighbor classification
- Nearest k-neighbors
- Nearest centroid classification
- Shrunken centroid classification
Nearest Neighbor Classifier

• To classify a sample in the validation set, determine it’s nearest neighbor in the training set; i.e. which sample in the training set is its gene expression profile is most similar to.
  – Similarity measure used is based on genes selected as being univariately differentially expressed between the classes
  – Correlation similarity or Euclidean distance generally used

• Classify the sample as being in the same class as it’s nearest neighbor in the training set
Nearest Centroid Classifier

• For a training set of data, select the genes that are informative for distinguishing the classes
• Compute the average expression profile (*centroid*) of the informative genes in each class
• Classify a sample in the validation set based on which centroid in the training set it’s gene expression profile is most similar to.
Other Methods

• Top-scoring pairs
  – Claim that it gives accurate prediction with few pairs because pairs of genes are selected to work well together

• Random Forrest
  – Very popular in machine learning community
  – Complex classifier
When There Are More Than 2 Classes

• Nearest neighbor type methods

• Decision tree of binary classifiers
Decision Tree of Binary Classifiers

- Partition the set of classes \{1,2,...,K\} into two disjoint subsets \(S_1\) and \(S_2\)
- Develop a binary classifier for distinguishing the composite classes \(S_1\) and \(S_2\)
  - Compute the cross-validated classification error for distinguishing \(S_1\) and \(S_2\)
- Repeat the above steps for all possible partitions in order to find the partition \(S_1\) and \(S_2\) for which the cross-validated classification error is minimized
- If \(S_1\) and \(S_2\) are not singleton sets, then repeat all of the above steps separately for the classes in \(S_1\) and \(S_2\) to optimally partition each of them
This procedure computes a classifier which can be used for predicting the class of a new sample.

Column defining classes:

Use random variance model for univariate tests.

Average over replicates of:

Pair samples by:

Arrays are paired between classes.

Prediction methods:
- Compound covariate predictor
- Diagonal linear discriminant analysis
- K-nearest neighbors (for K=1 and 2)
- Nearest centroid
- Support vector machines

Significant univariately at alpha level: 0.001

Optimize over the grid of alpha levels (and cross-validate optimization)

With univariate misclassification rate below: 0.25

With fold-ratio of geometric means between two classes exceeding: 2

Individual genes:

Number of pairs selected by the "Greedy pairs" method: 25

NOTE: This analysis is currently set to run on all genes passing the filter.
### Class Prediction

This procedure computes a classifier which can be used for predicting the class of a new sample.

#### Class Prediction Options

- **Cross-validation method:**
  - Leave-one-out validation
  - 10-fold validation (Repeated 1 times)
  - 0.632 bootstrap validation

- **Do statistical significance test of cross-validated misclassification rate.**
  - Number of permutations for significance test of cross-validated misclassification rate: 100

- **Use separate test set:**
  - Column containing "training", "predict", "exclude" labels:

- **Name to use for output files:**
  - ClassPrediction

- **Support vector machines**
This procedure computes a classifier which can be used for predicting the class of a new sample.

Class Prediction Options

- Cross-validation method:
  - Leave-one-out
  - Repeated
  - 0.632 bootstrap

Support vector machine parameters:
- Cost (tuning parameter):
- Weight of misclassifications in Class 1 relative to Class 2 (where Class 1 denotes the class label which would come first in an alphanumeric sorting of the class labels):

Options:
- Use separate test set for validation
- Use internal fixed random seed

OK Cancel Options Reset Help
### Prediction Analysis of Microarrays (PAM)

This tool is an interface to the Prediction Analysis of Microarrays (PAM) Package developed by R. Tibshirani, T. Hastie, B. Narasimha and G. Chu. Shrunken centroids algorithm is used for class predictions.

#### Column defining classes:

- [ ] Use separate test set
- [ ] Column containing "training", "predict", "exclude" labels:

#### Name to use for output files:

- PAM

#### Average over replicates of:

- [ ] Select gene subsets

---

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>Stage</td>
<td>SurvStatus</td>
<td>T Stage</td>
<td>Age Code</td>
<td>Medullo vs Glio vs PNET</td>
<td>Rhabdo vs Medullo vs AT/RT</td>
<td>AT/RT</td>
<td>AT/RT</td>
<td>AT/RT</td>
<td>AT/RT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**event descriptors** | **Gene annotations** | **Filtered log intensity** | **Gene identifiers** | **Scatterplot** | **Cluster view**
Binary tree class prediction

This tool computes a binary tree classifier which can be used for predicting the class of a new sample. At each stage (tree node), classes are divided into two groups. Cross-validation mis-classification rate is used to characterize the quality of the division. A division with the lowest mis-classification rate is used as a node of the tree. Then, procedure is repeated for each branch with two or more classes.

Column defining classes:

Prediction method:
- Compound covariate predictor
- K-nearest neighbors (for K=1)
- K-nearest neighbors (for K=3)
- Nearest centroid
- Support vector machines
- Diagonal linear discriminant analysis

Use separate test set
- Column containing "training", "predict", "exclude" labels:

Average over replicates of:

Predictors should only include genes:
- Significant univariately at level: 0.001
- With univariate misclassification rate below: 0.25
- With fold-ratio of geometric means between two classes exceeding: 2

NOTE: This analysis is currently set to run on all genes passing the filter.

Select gene subsets

OK  Cancel  Options  Reset  Help
This tool computes a binary tree classifier which can be used for predicting the class of a new sample. At each stage (tree node), classes are divided into two groups. Cross-validation misclassification rate is used to characterize the quality of the division. A division with the lowest misclassification rate is used as a node of the tree. Then, procedure is repeated for each branch with two or more classes.

**Binary Tree class prediction options**

- **Binary Tree options:**
  - Value of K (defining K-fold): 10
  - Do not split classes if the best achievable error rate is more than: 0.5

- **Support vector machine parameters:**
  - Cost (tuning parameter): 1
  - Weight of misclassifications in Class 1 relative to Class 2 (where Class 1 denotes the class label which would come first in an alphanumeric sorting of the class labels): 1

- **Name to use for output files:**
  - BinaryTreePrediction

- **Perform GC Observed vs. Expected analysis.**

**Correlation selection**

- Select gene subsets
This tool is used for Survival Risk Prediction based on the Supervised Principal Components method.


**Experimental design:**
- **Status column:**
  - 0 = censored, 1 = death
- **Column defining survival time:**

**Find gene lists determined by:**
- **Significance threshold of Cox Model:** 0.001
- **Number of Principal Components (1-10):** 2

**Covariates**
- **Clinical Covariates**
  - Column defining Covariate1:
  - Column defining Covariate2:
  - Column defining Covariate3:

**NOTE:** This analysis is currently set to run on all genes passing the filter.

[Select gene subsets]
Analysis of Variance

Column of exper descriptor sheet for factor A
Column of exper descriptor sheet for factor B
Column of exper descriptor sheet for factor C
Column of exper descriptor sheet for factor D
Column of exper description sheet for indicator of included arrays
Column of exper descriptor sheet for technical replicates
Threshold p value for testing effects
Threshold p value for testing the model
Threshold false discovery rate for testing effects
Threshold false discovery rate for testing the model
Model Formulas
Blocking factor(s)
Use Random Variance Model

Submit
View README
Analysis of variance (ANOVA) for each gene.

This plug-in performs an analysis of variance for relating the log-ratio or log-signal expression to specified factors. A separate ANOVA model is fitted for each gene. All factors are considered fixed effects.

The F-test used for statistical significance testing is based on the likelihood ratio test or type III sum of squares in SAS terminology. That means that the significance of each factor is adjusted for all other factors of the model.

To run this function, the user should input the following:

- Column of exper descriptor sheet for factor A
- Column of exper descriptor sheet for factor B (can be empty)
- Column of exper descriptor sheet for factor C (can be empty)
- Column of exper descriptor sheet for factor D (can be empty)
- Column of exper descriptor sheet for indicator of included arrays (can be empty)
- Column of exper descriptor sheet for technical replicates (can be empty)
- Threshold p value for testing effects
- Threshold p value for testing the model
- Threshold false discovery rate (Benjamini & Hochberg, 1995) for testing effects
- Threshold false discovery rate for testing the model
- Model formula (e.g. A+B:A:B). See next section for details how to specify a model formula.
- Blocking factors (e.g. B or B,C)
- Use Random variance Model (a checkbox)

The "Column of exper descriptor sheet for indicator of included arrays" is used to include arrays we are only interested in. For arrays we don't want them to be included in analyses, we should leave blank value in this column. We can put any value other than blank in this column for arrays we are interested in. If nothing is specified in the dialog, all arrays with non-empty factor labels will be used.
## ANOVA for Mixed-Effects Model

<table>
<thead>
<tr>
<th>Column of exper descriptor sheet for factor A</th>
<th>Column of exper descriptor sheet for factor B</th>
<th>Column of exper descriptor sheet for factor C</th>
<th>Column of exper descriptor sheet for random factor D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>Dx</td>
<td>Medub Type</td>
<td>Medub Stage</td>
</tr>
<tr>
<td>Age at Dx</td>
<td>Surv (months)</td>
<td>Sex</td>
<td></td>
</tr>
<tr>
<td>Threshold p value for testing fixed effects</td>
<td>.001</td>
<td>Threshold p value for testing the model</td>
<td>.001</td>
</tr>
<tr>
<td>Threshold false discovery rate for testing fixed effects</td>
<td>.1</td>
<td>Threshold false discovery rate for testing the model</td>
<td>.1</td>
</tr>
<tr>
<td>Model Formula</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Submit

View README
** NAME OF THIS PLUG-IN:**

** PURPOSE OF THIS PLUG-IN:**

This plug-in fits data to a linear mixed-effects model for each gene and computes ANOVA for a user’s specified model. Only one random factor can be specified in the mixed-effects model. This model is useful for time series data with specimens collected from each subject at multiple time points. The subjects may have different characteristics of interest, such as diseased or normal, treated or non-treated, male or female, but the genes differentially expressed among the subjects not related to those specified factors may not be of interest. Although a fixed effects model such as in the basic ANOVA provided in another plug-in could be used, if there are many subjects, then considering the subjects as a random factor can provide more degrees of freedom for error estimation and potentially greater statistical power for testing the effects of interest. This mixed-effects model is, however, much more computationally intensive than the standard fixed effects model provided in the other plug-in.

The F-test is based on the likelihood ratio test or type III sum of squares in SAS’s terminology. That means that the significance of each factor is adjusted for all other factors of the model.

** USING THIS PLUG-IN:**

To run this function, the user should input the following:
- Column of exper descriptor sheet for Factor A
- Column of exper descriptor sheet for Factor B (can be empty)
- Column of exper descriptor sheet for Factor C (can be empty)
- Column of exper descriptor sheet for random factor
- Column of exper descriptor sheet for indicator of included arrays (can be empty)
- Column of exper descriptor sheet for technical replicates (can be empty)
- Threshold p value for testing fixed effects
- Threshold p value for testing the model
- Threshold false discovery rate (Benjamini & Hochberg, 1995) for testing fixed effects
- Threshold false discovery rate for testing the model
- Model formula (e.g. A+B+A:B).
<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>M</td>
<td>Stage</td>
<td>SurvStatus</td>
<td>T Stage</td>
<td>Age</td>
<td>Code</td>
<td>Medulo vs PN</td>
<td>Glio vs PN</td>
<td>Medulo vs AT/RT</td>
<td>Medulo vs AT/RT</td>
<td>Medulo vs AT/RT</td>
<td>Medulo vs AT/RT</td>
</tr>
<tr>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
</tr>
</tbody>
</table>

### Time Series Analysis

- Column of exper descriptor sheet for time:
  - Time

- Column of exper descriptor sheet for class:
  - Array
  - Dx
  - Medulo Type
  - Medulo Stage
  - Sex
  - Age or Dx
  - Survival (months)

- Column of exper description sheet for indicator of included arrays:
  - Threshold p value for testing effects
  - Threshold false discovery rate for testing effects

- Additional data:
  - 1 Medulloblastoma
  - 0 Medulloblastoma
  - 1 Medulloblastoma
  - 0 Medulloblastoma
  - 1 Medulloblastoma
  - 1 Medulloblastoma

### Buttons
- Submit
- View README
This plug-in can be used for regression analysis of time series expression data. In its simplest form (model A), the genes whose expression are varying over time are identified. A quadratic function is fit to the expression data of each gene and the hypothesis is that the linear and quadratic coefficients are simultaneously zero. The genes for which this hypothesis is rejected are identified. The tests are performed at a significance level specified by the user and also at a false discovery rate (FDR) specified by the user. Two lists of significant genes are produced, one for the specified significance level threshold and one for the FDR threshold. To fit this model, the user must provide a column in the experiment descriptor worksheet specifying the time point for each array. This column should be strictly numeric and should not contain alphabetic characters. The entry in the column should be blank if the array is to be excluded from the analysis. The arrays at the same time points can represent either technical or biological replicates, but the two kinds of replicates should not be combined in the same analysis. This plug-in is not appropriate for nested data where the same subject is sampled at different time points.

Model B is for identifying genes that are changing over time, but where there is a class variable to adjust for. For example, there could be two strains of mice included in the experiment or arrays were from two different print set batches. For model B it is assumed that the variation in gene expression over time is the same for each class. The output also indicates which genes are differentially expressed among the classes uniformly over time.

Model C is similar to model B but the variation in gene expression over time is permitted to differ among the classes. The output of model C identifies these genes for which the variation over time is different for different levels of the class variable. These genes are identified based on the user-specified significance level and based on the user-specified p-value. For genes whose expression does not significantly vary among classes, model B is fit to determine whether the gene is varying over time uniformly for each classes. Model C is useful for experiments where the class variable represents a treatment indicator.

For data without a class variable, the ANOVA model takes the form:
There are two steps for running ANOVA in this plugin. The first step is to normalize log intensity for each channel and the second step is to do ANOVA on normalized log intensity.

**Step 1: Normalization**

The following normalization model is used:

\[ y_{(adcg)} = \mu + A \cdot a + AD \cdot (ad) + e1_{(adcg)} \]  \[ (1) \]

where:
- \( y_{(adcg)} \) is the log intensity,
- \( \mu \) is the overall mean log intensity,
- \( A \) is the effect of the array a,
- \( AD \cdot (ad) \) is the interaction of array a and dye d,
- \( e1_{(adcg)} \) is the random noise,
- \( c \) is the index of class (variant).

We assume each effect is fixed. So \( A \cdot a \) and \( AD \cdot (ad) \) satisfy some identification conditions.

After fitting the normalization model, the residuals (normalized log intensity) \( r_{(adcg)} \) are obtained:

\[ r_{(adcg)} = y_{(adcg)} - \hat{\mu} - \hat{A} \cdot a - \hat{AD} \cdot (ad) \]  \[ (2) \]

**Step 2: ANOVA**

Fit the normalized log intensity by the following ANOVA model:

\[ r_{(adcg)} = \mu_{(g)} + \alpha_{(ag)} + \beta_{(dg)} + class_{(cg)} + e2_{(adcg)} \]  \[ (3) \]

where:
- \( \mu_{(g)} \) is the gene-specific average log intensity,
- \( \alpha_{(ag)} \) is the gene-specific array effect (spot effect),
- \( \beta_{(dg)} \) is the gene-specific dye effect,
- \( class_{(cg)} \) is the gene-specific class (variant) effect,
- \( e2_{(adcg)} \) is the random noise.

Again, we assume each effect in model (3) is fixed.
<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Site M Stage</td>
<td>SurvStatus</td>
<td>T Stage</td>
<td>Age C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rhabdo vs PNET</td>
<td>Rhabdo vs PNET</td>
<td>Rhabdo vs AT/RT</td>
<td>Rhabdo vs AT/RT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td>PNET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>&gt;0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T Site</td>
<td>M Stage</td>
<td>SurvStatus</td>
<td>T Stage</td>
<td>Age C</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td>CNS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td>Medulloblastoma</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Dataset:**
- **Rhabdo vs PNET**

**Columns:**
- **T Site**: Type of site
- **M Stage**: M stage
- **SurvStatus**: Survival status
- **T Stage**: T stage
- **Age C**: Age category

**Data Description:**
- Each row represents a patient's record.
- Columns provide specific data points for classification and analysis.

**Analysis Tools:**
- Multidimensional scaling
- Class comparison
- Class prediction
- Survival analysis
- Quantitative trait analysis
- Filter and subset the data
- Plugins
- Utilities

**Support Options:**
- Getting started
- Manuals
- Email support
- ListServ
Evaluating a Classifier

• “Prediction is difficult, especially the future.”
  – Neils Bohr

• Fit of a model to the same data used to develop it is no evidence of prediction accuracy for independent data.
Evaluating a Classifier

• Fit of a model to the same data used to develop it is no evidence of prediction accuracy for independent data
  – Goodness of fit vs prediction accuracy
• Demonstrating statistical significance of prognostic factors is not the same as demonstrating predictive accuracy
• Demonstrating stability of identification of gene predictors is not necessary for demonstrating predictive accuracy
Evaluating a Classifier

• The classification algorithm includes the following parts:
  – Determining what type of classifier to use
  – Gene selection
  – Fitting parameters
  – Optimizing with regard to tuning parameters

• If a re-sampling method such as cross-validation is to be used to estimate predictive error of a classifier, all aspects of the classification algorithm must be repeated for each training set and the accuracy of the resulting classifier scored on the corresponding validation set
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
• Cross validation is only valid if the test set is not used in any way in the development of the model. Using the complete set of samples to select genes violates this assumption and invalidates cross-validation.

• 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
• $P_i \sim \text{MVN}(0, \mathbf{I}_{6000})$
• Can we distinguish between the first 7 specimens (Class 1) and the last 7 (Class 2)?

Prediction Method
• Compound covariate prediction (discussed later)
• Compound covariate built from the log-ratios of the 10 most differentially expressed genes.
Number of misclassifications

Proportion of simulated data sets

Cross-validation: none (resubstitution method)
Cross-validation: after gene selection
Cross-validation: prior to gene selection

Number of misclassifications
Prediction Error Estimation: A Comparison of Resampling Methods

Annette M. Molinaro \textsuperscript{a,b}; Richard Simon \textsuperscript{b}; Ruth M. Pfeiffer \textsuperscript{b}

\textsuperscript{a}Biodiostatistics Branch, Division of Cancer Epidemiology and Genetics, NCI, NIH, Rockville, MD 20852, \textsuperscript{b}Department of Epidemiology and Public Health, Yale University School of Medicine, New Haven, CT 06520, \textsuperscript{c}Biometric Research Branch, Division of Cancer Treatment and Diagnostics, NCI, NIH, Rockville, MD 20852

ABSTRACT

Motivation: In genomic studies, thousands of features are collected on relatively few samples. One of the goals of these studies is to build classifiers to predict the outcome of future observations. There are three inherent steps to this process: feature selection, model selection, and prediction assessment. With a focus on prediction assessment, we compare several methods for estimating the true prediction error of a prediction model in the presence of feature selection.

Results: For small studies where features are selected from thousands of candidates, the resubstitution and simple split-sample estimates are seriously biased. In these small samples, leave-one-out (LOOCV), 10-fold cross-validation (CV), and the 532+ bootstrap have the smallest bias for linear discriminant analysis, nearest neighbor, and classification tree. LOOCV and 10-fold CV have the smallest bias for linear discriminant analysis. Additionally, LOOCV, 5- and 10-fold CV, and the 532+ bootstrap have the lowest mean square error. The 532+ bootstrap is quite biased in small sample sizes with strong signal to noise ratios. Differences in performance among resampling methods are reduced as the number of specimens available increase.

Availability: A complete compilation of results in tables and figures is available in Molinaro et al. (2005). A code for simulations and analyses is available from the authors.

Contact: annette.molinaro@yale.edu

1 INTRODUCTION

In genome experiments one frequently encounters high dimensional data and small sample sizes. Microarrays simultaneously monitor expression levels for several thousands of genes. Proteomic profiling studies using SELDI-TOF (surface-enhanced laser desorption and ionization time-of-flight) measure size and charge of proteins and protein fragments by mass spectroscopy, and result in up to 15,000 intensity levels at prespecified mass values for each spectrum. Sample sizes in such experiments are typically less than 109.

In many studies observations are known to belong to predetermined classes and the task is to build predictors or classifiers for new observations whose class is unknown. Estimating which genes or proteomic measurements to include in the prediction is called feature selection and is a crucial step in developing a class predictor including too many noisy variables reduces accuracy of the prediction and may lead to over-fitting of data, resulting in promising but often non-reproducible results (Ruan & Hoff, 2004).

Another difficulty is model selection with numerous classification models available. An important step in reporting results is assessing the chosen model's error rate, or generalizability. In the absence of independent validation data, a common approach to estimating predictive accuracy is based on some form of resampling the original data, e.g., cross-validation. These techniques divide the data into a training set and a test set and range in complexity from the popular learning-test split to k-fold cross-validation, Monte-Carlo k-fold cross-validation, and bootstrap resampling. Few comparisons of standard resampling methods have been performed to date, and all of them exhibit limitations that make their conclusions inapplicable to most genomic settings. Early comparisons of resampling techniques in the literature are focused on model selection as opposed to prediction error estimation (Breslow & Stanger, 1999; Hutter, 1989). In two recent assessments of resampling techniques for error estimation (Stinger & Doughterty, 2004; Elton, 2004), feature selection was not included as part of the resampling procedures, causing the conclusions to be inappropriate for the high-dimensional setting.

We have performed an extensive comparison of resampling methods to estimate prediction error using simulated (large signal to noise ratio), microarray (intermediate signal to noise ratio) and proteomic data (low signal to noise ratio), encompassing increasing sample sizes with large numbers of features. The impact of feature selection on the performance of various error estimation methods is highlighted. The results elucidate the 'best' resampling techniques for...
Simulated Data
40 cases, 10 genes selected from 5000

<table>
<thead>
<tr>
<th>Method</th>
<th>Estimate</th>
<th>Std Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>.078</td>
<td></td>
</tr>
<tr>
<td>Resubstitution</td>
<td>.007</td>
<td>.016</td>
</tr>
<tr>
<td>LOOCV</td>
<td>.092</td>
<td>.115</td>
</tr>
<tr>
<td>10-fold CV</td>
<td>.118</td>
<td>.120</td>
</tr>
<tr>
<td>5-fold CV</td>
<td>.161</td>
<td>.127</td>
</tr>
<tr>
<td>Split sample 1-1</td>
<td>.345</td>
<td>.185</td>
</tr>
<tr>
<td>Split sample 2-1</td>
<td>.205</td>
<td>.184</td>
</tr>
<tr>
<td>.632+ bootstrap</td>
<td>.274</td>
<td>.084</td>
</tr>
</tbody>
</table>
## DLBCL Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Bias</th>
<th>Std Deviation</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOOCV</td>
<td>-.019</td>
<td>.072</td>
<td>.008</td>
</tr>
<tr>
<td>10-fold CV</td>
<td>-.007</td>
<td>.063</td>
<td>.006</td>
</tr>
<tr>
<td>5-fold CV</td>
<td>.004</td>
<td>.07</td>
<td>.007</td>
</tr>
<tr>
<td>Split 1-1</td>
<td>.037</td>
<td>.117</td>
<td>.018</td>
</tr>
<tr>
<td>Split 2-1</td>
<td>.001</td>
<td>.119</td>
<td>.017</td>
</tr>
<tr>
<td>.632+ bootstrap</td>
<td>-.006</td>
<td>.049</td>
<td>.004</td>
</tr>
</tbody>
</table>
## Simulated Data

### 40 cases

<table>
<thead>
<tr>
<th>Method</th>
<th>Estimate</th>
<th>Std Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>.078</td>
<td></td>
</tr>
<tr>
<td>10-fold</td>
<td>.118</td>
<td>.120</td>
</tr>
<tr>
<td>Repeated 10-fold</td>
<td>.116</td>
<td>.109</td>
</tr>
<tr>
<td>5-fold</td>
<td>.161</td>
<td>.127</td>
</tr>
<tr>
<td>Repeated 5-fold</td>
<td>.159</td>
<td>.114</td>
</tr>
<tr>
<td>Split 1-1</td>
<td>.345</td>
<td>.185</td>
</tr>
<tr>
<td>Repeated split 1-1</td>
<td>.371</td>
<td>.065</td>
</tr>
</tbody>
</table>
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 in the real data
Common Problems With Internal Classifier Validation

• Pre-selection of genes using entire dataset

• Failure to consider optimization of tuning parameter part of classification algorithm
  – Varma & Simon, BMC Bioinformatics 2006

• Erroneous use of predicted class in regression model
Incomplete (incorrect) Cross-Validation

- Publications are using all the data to select genes and then cross-validating only the parameter estimation component of model development
  - Highly biased
  - Many published complex methods which make strong claims based on incorrect cross-validation.
    - Frequently seen in complex feature set selection algorithms
    - Some software encourages inappropriate cross-validation
Incomplete (incorrect) Cross-Validation

- Let $M(b,D)$ denote a classification model developed on a set of data $D$ where the model is of a particular type that is parameterized by a scalar $b$.
- Use cross-validation to estimate the classification error of $M(b,D)$ for a grid of values of $b$; $Err(b)$.
- Select the value of $b^*$ that minimizes $Err(b)$.
- Caution: $Err(b^*)$ is a biased estimate of the prediction error of $M(b^*,D)$.
- This error is made in some commonly used methods.
Complete (correct) Cross-Validation

- Construct a learning set \( D \) as a subset of the full set \( S \) of cases.
- Use cross-validation restricted to \( D \) in order to estimate the classification error of \( M(b,D) \) for a grid of values of \( b \); \( \text{Err}(b) \).
- Select the value of \( b^* \) that minimizes \( \text{Err}(b) \).
- Use the model \( M(b^*,D) \) to predict for the cases in \( S \) but not in \( D \) (\( S-D \)) and compute the error rate in \( S-D \).
- Repeat this full procedure for different learning sets \( D_1 \), \( D_2 \) and average the error rates of the models \( M(b_i^*,D_i) \) over the corresponding validation sets \( S-D_i \).
Does an Expression Profile Classifier Predict More Accurately Than Standard Prognostic Variables?

• Not an issue of which variables are significant after adjusting for which others or which are independent predictors
  – Predictive accuracy and inference are different

• The two classifiers can be compared with regard to predictive accuracy

• The predictiveness of the expression profile classifier can be evaluated within levels of the classifier based on standard prognostic variables
External Validation

- Should address clinical utility, not just predictive accuracy
  - Therapeutic relevance
- Should incorporate all sources of variability likely to be seen in broad clinical application
  - Expression profile assay distributed over time and space
  - Real world tissue handling
  - Patients selected from different centers than those used for developing the classifier
Survival Risk Group Prediction

- Evaluate individual genes by fitting single variable proportional hazards regression models to log signal or log ratio for gene
- Select genes based on p-value threshold for single gene PH regressions
- Compute first k principal components of the selected genes
- Fit PH regression model with the k pc’s as predictors. Let $b_1, \ldots, b_k$ denote the estimated regression coefficients
- To predict for case with expression profile vector $x$, compute the k supervised pc’s $y_1, \ldots, y_k$ and the predictive index $\lambda = b_1 y_1 + \ldots + b_k y_k$
Survival Risk Group Prediction

• LOOCV loop:
  – Create training set by omitting i’th case
• Develop supervised pc PH model for training set
• Compute cross-validated predictive index for i’th case using PH model developed for training set
• Compute predictive risk percentile of predictive index for i’th case among predictive indices for cases in the training set
Survival Risk Group Prediction

• Plot Kaplan Meier survival curves for cases with cross-validated risk percentiles above 50% and for cases with cross-validated risk percentiles below 50%
  – Or for however many risk groups and thresholds is desired

• Compute log-rank statistic comparing the cross-validated Kaplan Meier curves
Survival Risk Group Prediction

• Repeat the entire procedure for all (or large number) of permutations of survival times and censoring indicators to generate the null distribution of the log-rank statistic
  – The usual chi-square null distribution is not valid because the cross-validated risk percentiles are correlated among cases

• Evaluate statistical significance of the association of survival and expression profiles by referring the log-rank statistic for the unpermuted data to the permutation null distribution
Survival Risk Group Prediction

- Other approaches to survival risk group prediction have been published
- The supervised pc method is implemented in BRB-ArrayTools
- BRB-ArrayTools also provides for comparing the risk group classifier based on expression profiles to one based on standard covariates and one based on a combination of both types of variables
Sample Size Planning

References

• K Dobbin, R Simon. Sample size determination in microarray experiments for class comparison and prognostic classification. Biostatistics 6:27-38, 2005

• K Dobbin, R Simon. Sample size planning for developing classifiers using high dimensional DNA microarray data. Biostatistics (In Press)