Advanced Statistical Methods: Beyond Linear Regression

John R. Stevens
Utah State University

Notes 3. Statistical Methods II

Mathematics Educators Workshop
28 March 2009

http://www.stat.usu.edu/~jrstevens/pcmi
ALL Data

- “Preprocessed” gene expression data
  - 12625 genes (hgu95av2 Affymetrix GeneChip)
  - 128 samples (arrays)
  - a matrix of “expression values” – 128 cols, 12625 rows

- Phenotypic data on all 128 patients, including:
  - 95 B-cell cancer
  - 33 T-cell cancer

- Motivating question: Which genes are changing expression values systematically between B-cell and T-cell groups?

- Needle(s) in a haystack …
Basic idea of differential expression (DE)

- “Observe” gene expression in different conditions – healthy vs. diseased, e.g.
- Decide which genes’ expression levels are changing significantly between conditions
- Target those genes – to halt disease, e.g.
- Note: there are far too many ways to test for DE to present here – we will just look at major themes of most of them, and focus on implementing one
Miscellaneous statistical issues

- Test each gene individually
  - Dependence structure among genes not well-understood: (co-regulation or co-expression)
  - Ignore coregulation – first, one at a time

- Scale of data
  - Magnitude of change depends on scale
  - In general: log scale is “approximately right”
  - Variance stabilization transformation can help
Simple / Naïve test of DE

- Observe gene expression levels under two conditions

\[ Y_{ijk} = \text{log-scale expr. level of gene k in replicate j of "treatment" i} \]

- Calculate: average log “fold change”

\[ \overline{Y}_{i.k} = \text{ave. log expr. for gene k in treatment i} \]

\[ LFC_k = \overline{Y}_{2.k} - \overline{Y}_{1.k} = \text{ave. log fold change for gene k} \]

- Make a cut-off: R

Gene k is "significant" if \( |LFC_k| > R \)
What does naïve test do?

- Estimate degree of differential expression:
  - LFC > 0 for “up-regulated” genes
  - LFC < 0 for “down-regulated” genes

- Identifies genes with largest observed change

- A simple mean comparison ignores something…
  - cannot really test for “significance”
  - what if larger LFC have large variability?
    - then not necessarily significant
How to take variability into account?

- Build some test statistic on a per-gene basis

- How do we “usually” test for mean differences between two groups or samples?

- Test statistic:

\[
t_k = \frac{\bar{Y}_{2,k} - \bar{Y}_{1,k}}{S_k} = \frac{LFC_k}{S_k}
\]

pooled SD
How to use this to “test” for DE?

- What is being tested?
  
  Null: No change for gene $k$

- Under null, $t_k \sim t$ dist. with $n_k$ d.f.
  
  “parametric” assumption ...

- But what is needed to do this?
  
  - “Large” sample size
  - Estimate $\sigma_k = “pop. SD”$ for gene $k$
    (example: $s_k$)
What if we don’t have enough?

- Probably don’t – even dozens of arrays may not suffice

- Two main problems:
  1. Estimate $\sigma_k$ (especially for small sample size)
  2. Appropriate sampling distribution of test stat.

- Basic solutions:
  1. To estimate $\sigma_k$: Pool information across genes
  2. For comparison against ‘sampling distribution’:
     - use parametric assumption on “improved” test stat.
     - use non-parametric methods – resampling / permuting
Test for DE with limma / eBayes

- For gene k under treatment j on array i:

\[ Y_{ijk} = \beta_{k,0} + \beta_{k,1}T_{jk} + \varepsilon_{ijk}, \quad \text{Var}[\varepsilon_{ijk}] = \sigma_k^2 \]

(expression level (log scale) + treatment effect (DE) + treatment level (here, 0 or 1))

- What if there are more covariates than just treatment? – use matrix notation for convenience:

\[ E[Y_k] = X\beta_k \]

(log-scale expression vector + design matrix (n x p))
Assumptions in linear model

Obtain estimates $\hat{\beta}_k$ and $\hat{\sigma}_k$, and $\text{Var} \left[ \hat{\beta}_k \right] = V_k \hat{\sigma}_k^2$

For covariate $w$,

$$\hat{\beta}_{k,w} \mid \beta_{k,w}, \sigma_k^2 \sim N \left( \beta_{k,w}, V_{k,w,w} \hat{\sigma}_k^2 \right)$$

$$\hat{\sigma}_k^2 \mid \sigma_k^2 \sim \frac{\sigma_k^2}{d_k} \chi^2_{d_k}, \ d_k = \text{resid. d.f.} = n - p$$

Then $t_{k,w} = \frac{\hat{\beta}_{k,w}}{\hat{\sigma}_k \sqrt{V_{k,w,w}}} \sim t_{d_k}$

This is something we’ve already done …

Can we “improve” it?
Hierarchical model to borrow information across genes: eBayes

Assume prior distribution \( \frac{1}{\sigma_k^2} \sim \frac{1}{d_0s_0^2} X^2_{d_0} \)

\((s_0^2 \text{ and } d_0 \text{ estimated from data using empirical Bayes methods})\)

(\text{using all of the genes})

Consider the posterior mean \( \tilde{\sigma}_k^2 = E\left[\sigma_k^2 \mid \hat{\sigma}_k^2\right] = \frac{d_0s_0^2 + d_k\hat{\sigma}_k^2}{d_0 + d_k} \)

Then the "moderated" t-statistic \( \tilde{t}_{k,w} = \frac{\hat{\beta}_{k,w}}{\tilde{\sigma}_k \sqrt{V_{k,w,w}}} \sim t_{d_0 + d_k} \)

represents: added information (from using all genes)
Now – how many genes will be called significant?
Significance and P-values

- Usually, “small” P-value → claim significance
- Correct interpretation of P-value from a test of significance:
  “The probability of obtaining a difference at least as extreme as what was observed, just by chance when the null hypothesis is true.”

- Consider a t-test of $H_0: \mu_1 - \mu_2 = 0$, when in reality, $\mu_1 - \mu_2 = c$ (and SD=1 for both pop.)
- What P-values are possible, and how likely are they?
For each value of $c$, 1000 data sets (think of as 1000 genes) were simulated where two populations are compared, and the "truth" is $\mu_1 - \mu_2 = c$. For each data set, the t-test evaluates $H_0: \mu_1 - \mu_2 = 0$ (think of as no change in expression level). The resulting P-values for all data sets are summarized in the histograms.

What's going on here?
Histograms smoothed and overlayed

Note:

- Even when there is no difference (c=0), very small P-values are possible.

- Even for larger differences (c=0.2), very large P-values are possible.

- When we look at a histogram of P-values from our test of DE, we have a mixture of these distributions (because each gene has its own true value for c).
So this is a mixture of distributions.

A flat histogram would suggest that there really aren’t any DE genes.

The peak near 0 indicates that some genes are DE.

But which ones?
How to treat these P-values?

- Traditionally, consider some cut-off

  Reject null if P-value < $\alpha$, for example (often $\alpha = 0.05$)

- What does this mean?

  $\alpha$ is the acceptable level of Type I error:
  $\alpha = P(\text{reject null} \mid \text{null is true})$
Multiple testing

- We do this with many (thousands, often) genes simultaneously – say $m$ genes

<table>
<thead>
<tr>
<th></th>
<th>Fail to Reject Null</th>
<th>Reject Null</th>
<th>Total Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null True</td>
<td>$U$</td>
<td>$V$</td>
<td>$m_0$</td>
</tr>
<tr>
<td>Null False</td>
<td>$T$</td>
<td>$S$</td>
<td>$m-m_0$</td>
</tr>
</tbody>
</table>

- # of Type I errors: $V$
- # of Type II errors: $T$
- # of correct “decisions”: $U+S$
Error rates

- Think of this as a family of m tests or comparisons
- Per-comparison error rate: $\text{PCER} = E[V/m]$
- Family-wise error rate: $\text{FWER} = P(V \geq 1)$
- What does the $\alpha$-cutoff mean here?
  Testing each hypothesis (gene) at level $\alpha$
  guarantees:

  $\text{PCER} \leq \alpha$

  - let’s look at why
What are P-values, really?

Suppose T is the test stat., and t is the observed T.

\[ Pval = P(T > t \mid H_0) \]

Assume \( H_0 \) is true. Let \( F \) be the cdf of T and \( f \) be pdf:

\[ F(t) = P(T \leq t) = \int_{-\infty}^{t} f(t) dt = 1 - Pval \]

What is the distribution of \( Y = F(t) \)? Let \( g \) be pdf of \( Y \):

\[ \frac{dy}{dt} = F'(t) = f(t), \quad g(y) = f(t) \frac{dt}{dy} = f(t) \frac{1}{f(t)} = 1 \]

So \( Y = 1 - Pval \) is Uniform[0,1].

Then when \( H_0 \) is true, \( Pval \sim U[0,1] \).
P-values and $\alpha$ cut-off

- Suppose null is true for all $m$ genes - (so none of the genes are differentially expressed)

- Look at histogram of $m=1000$ P-values with $\alpha=0.05$ cut-off
  
  - about 50 “significant” just by chance
  these can be “expensive” errors

(Here, $V/m \approx 50/1000 = 0.05$.)
How to control this error rate?

Look at controlling the FWER:
Testing each hypothesis (gene) at $\alpha/m$ instead of $\alpha$ guarantees:
FWER $\leq \alpha$

This is called the Bonferroni correction

This is far too conservative for large $m$
A more reasonable approach

- Consider these corrections sequentially:

  Let $P_i$ be the P-value for testing gene $i$, with null $H_i$. Let $P(1) \leq P(2) \leq \ldots \leq P(m)$ be the ordered P-values.

  Let $k$ be the largest $i$ for which $P(i) \leq \frac{i}{m} \alpha$.

  Reject all $H(i)$ for $i = 1, 2, \ldots, k$.

- Then for independent test statistics and for any configuration of false null hypotheses, this procedure guarantees:

  $E[V/R] \leq \alpha$
What does this mean?

- \( V = \# \) of “wrongly-rejected” nulls
- \( R = \) total \# of rejected nulls
- Think of rejected nulls as “discovered” genes of significance
- Then call \( E[V/R] \) the FDR
  - False Discovery Rate
- This is the Benjamini-Hochberg FDR correction – sometimes called the marginal FDR correction
Benjamini-Hochberg adjusted $P$-values

Let $P_1 \leq P_2 \leq \ldots \leq P_m$ be the ordered $P$-values.

Let $P_{(i)}^{(adj)} = P_{(i)} \cdot \frac{m}{i}$.

If any $P_{(i)}^{(adj)} > 1$, reset it to 1.

If any $P_{(i)}^{(adj)} > P_{(i+1)}^{(adj)}$, reset it to $P_{(i+1)}^{(adj)}$ (starting at the end of the list, checking backwards).

Then $P_{(1)}^{(adj)} \leq P_{(2)}^{(adj)} \leq \ldots \leq P_{(m)}^{(adj)}$ are the ordered BH - FDR - adjusted $P$-values.
An extension: the q-value

• p-value for a gene:
  the probability of observing a test stat.
  more extreme when null is true

• q-value for a gene:
  the expected proportion of false positives
  incurred when calling that gene significant

• Compare (with slight abuse of notation):

\[ pval = P(T > t \mid H_0 \text{ true}) \quad qval = P(H_0 \text{ true} \mid T > t) \]
Useful visualization technique: the volcano plot

Good for visualizing importance of variability

Loss of information?
Smoothed Color Density Representation

- **Interpretation:**
  - color
    represents
    density around corresponding point

- **How is this better?**
  - visualize overlayed points
Another visualization tool: heatmap

- Expression (matrix) on color scale (here, dark red to dark blue)
- Rows and columns clustered … “dendrogram”
Hierarchical clustering: agnes

- **Agglomerative Nesting** – (hierarchical clustering)

- Start with “singleton” clusters – each vector is its own group
- Find the two “closest” vectors and “merge” them – distance usually Euclidean; form a cluster
- Then recalculate distances: Linkage – distance between clusters
  - Average linkage: average of dissimilarities between clusters
  - Single linkage: dissimilarity between “nearest neighbors”
  - Complete linkage: “farthest neighbors”
Side note: color matters!

www.vischeck.com/vischeck/vischeckImage.php
Gene Profiling / Selection

- “Observe” gene expression in different conditions – healthy vs. diseased, e.g.

- Use simultaneous expression “profiles” of thousands of genes (what are the genes doing across arrays)

- Look at which genes are “important” in “separating” the two conditions; i.e., what determines the conditions’ “signatures”
Machine Learning

- Computational & statistical inference processes: observed data → reusable algorithms for prediction
- Why “machine”? 
  want minimal human involvement
- Why “learning”? 
  develop ability to predict
- Here, supervised learning: 
  use knowledge of condition type
Machine Learning Methods

- Neural Network
- SVM (Support Vector Machine)
- RPART (Recursive PArtitioning and Regression Trees)
- CART (Classification and Regression Trees)
- Ensembling Learning (average of many trees)
  - Boosting
  - Bagging
  - RandomForests
CART: Classification and Regression Trees

- Each individual (array) has data on many predictors (genes) and one response (disease state)

- Think of a tree, with splits based on levels of specific predictors

- Choose predictors and split levels to maximize “purity” in new groups; the best split at each node

- Prediction made by: passing test cases down tree
CART generalized: Random Forests

- Rather than using all predictors and all individuals to make a single tree, make a forest of many \( n_{tree} \) trees, each one based on a random selection of predictors and individuals.

- Each tree is fit using a bootstrap sample of data (draw with replacement) and ‘grown’ until each node is ‘pure’.

- Each node is split using the best among a subset (of size \( m_{try} \)) of predictors randomly chosen at that node (default is sqrt. of # of predictors) (special case using all predictors: bagging).

- Prediction made by aggregating across the forest (majority vote or average).
How to measure “goodness”? 

• Each tree fit on a “training” set (bootstrap sample), or the “bag”

• The left-over cases (“out-of-bag”) can be used as a “test” set for that tree (usually 1/3 of original data)

• The “out-of-bag” (OOB) error rate is the: % misclassification
What does RF give us?

- Kind of a “black box”
  - but can look at “variable importance”
- For each tree, look at the OOB data:
  - Permute values of predictor $j$ among all OOB cases
  - Pass OOB data down the tree, save the predictions
  - For case $i$ of OOB and predictor $j$, get:
    - OOB error rate with variable $j$ permuted – OOB error rate before permutation
- Average across forest to get overall variable importance for each predictor $j$
Why “variable importance”? 

- Recall: identifying conditions’ signatures 
- Sort genes by some criterion 
- Want smallest set of genes to achieve good diagnostic ability
ALL subset: 3,024 genes (FDR sig. at .05), 30 arrays (15 B, 15 T)
Look at top 3 most “important” genes from Random Forest.
On the subset of 30 arrays:

Over all 128 arrays:
Then what?

- List of candidate genes: needles in a haystack

- Further study:
  - common pathways and functions
  - validation: qRT-PCR, etc.

- Hypothesis generation