GS01 0163
Analysis of Microarray Data

Keith Baggerly and Kevin Coombes
Section of Bioinformatics
Department of Biostatistics and Applied Mathematics
UT M. D. Anderson Cancer Center
kabagg@mdanderson.org
kcoombes@mdanderson.org

16 November 2004

Lecture 22: Predicting clinical outcome

- A tale of two studies
- Gleason grade and Gleason score
- Combining the data from two studies
- Logistic Regression

A tale of two studies

We start by reviewing two published microarray studies of prostate cancer. We have looked previously at the first study:

Reference: Lapointe et al. Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proc Natl Acad Sci USA*. 2004; **101**: 811–816.

- 62 samples of prostate cancer
- 41 matched samples of normal prostate
- 9 samples of lymph node metastases from prostate cancer

Lapointe's gene filtering

This paper uses two-color microarrays produced at Stanford. They were processed with local background subtraction, global (mean) normalization, then taking log ratios with the reference channel.

After preprocessing the data, they filtered the genes in two steps:

- Intensity filter (intensity > 1.5-fold above background in both experimental samples and reference samples in at least 75% of experiments)
- Variation filter (≥ 3-fold variation from the mean in at least two samples)

Lapointe's clustering

Next, they performed hierarchical clustering. No information was supplied on the distance measure nor on the linkage rule. No validation of clusters was performed.

Claimed result: Can distinguish normals from tumors with two exceptions. (The two tumors clustered with normals are both unusual.) They also claim to find three subclusters of the tumor population.

Lapointe's tumor subtypes

Continuing with their analysis, they looked at how clinical features matched the putative subtypes. One of the three subtypes (subtype III) contained 8 of the 9 lymph node metastases.

Interestingly, they pooled subtypes II and III before performing chi-squared tests to look at how clinical information matched the subtypes. Based on these chi-squared tests, higher grade tumors and advanced stage tumors were more likely to be clustered in combined group II–III than in group I.

Lapointe's study of differential expression

Next, they looked for genes that were differentially expressed between various subgroups of cancer. Differntial expressed genes were selected using Significance Analysis of Microarrays (SAM). They compared tumors based on:

- Gleason grade ($\leq 3+4$ vs. $\geq 4+3$), finding 41 genes
- Stage ($\leq T2$ vs. $\geq T3$), finding 11 genes
- Recurrence, finding 23 genes

A second study

Reference: Singh et al. Geen expression correlates of clinical prostate cancer behavior. *Cancer Cell*. 2002; **1**: 203–209.

- 52 samples of prostate cancer
- 50 samples of apparently normal prostate

This study used Affymetrix U95Av2 oligonucleotide arrays. They were quantified using MAS4.0, and genes that varied less than 5-fold across the sample set were removed from consideration.

Singh's study of differential expression

Genes were ranked according to their differential expression between tumor and normal using a variant of the t-statistic (with a non-standard estimate of the pooled standard deviation). Significance was determined using a permutation test, and 317 genes were found to be up-regulated in prostate cancer, along with 139 genes that were down-regulated.

Singh's prediction of tumor status

Next, they built a classifier to predict tumor vs. normal using k-nearest neighbors. (The value of k is not specified in the paper.) Leave-one-out cross-validation was performed. (Feature selection by ranking genes was included in the cross-validation.) They constructed 4-gene and 16-gene models; the 4-gene model had a leave-one-out cross-validation accuracy of 90%.

Singh's investigation of clinical correlates

They looked for differntial expression with respect to a number of clincal factors; the only one that appeared significant was Gleason grade, where they found 29 differentially expressed genes.

They also tried to predict recurrence and survival,with limited success.

Gleason grade and Gleason score

Biopsies of prostate cancer are graded by pathologists using a scale developed by Gleason. The grade is based on the appearance of the tumor specimen under a microscope. An individual tumor focus is graded on a scale from 1 to 5, with 1 being essentially normal in appearance. The severity of the tumor increases (and it appears less differentiated) as the grade increases.

In general, prostate cancer appears in multiple distinct foci. The two most abundant foci are graded separately, and the Gleason score is often summarized by adding the two grades. A patient with prostate cancer can get a Gleason score of 7 represented as a $4 + 3$, which means that the largest tumor focus is grade 4 and the secondary tumor focus if grade 3. This may well represent a worse tumor than a Gleason 7 that arises from a $3 + 4$.

Gleason score and prognosis

In general, patients with a Gleason score of 6 or lower have a good prognosis and a low risk of recurrence after surgery. Standard clinical practice for these patients is to “watch-and-wait”.

By contrast, patients with a Gleason score of 8 or higher have a poor prognosis and a high risk of recurrence afer surgery. These patients are usually treated witrh chemotherapy, which is often effective in preventing recurrence.

Patients with a Gleason score of 7 have an intermediate risk, and it is unclear how best to proceed.

GOAL: Find a model that can predict whether a patient has good prognosis (defined as Gleason 6 or lower) or poor prognosis (Gleason 8 or higher), and then see what this model tells us about patients with Gleason 7 prostate cancer.

Combining the data from two studies

We get a much larger sample size if we combine the data:

Study	Low	Medium	High	Unknown	Total
Singh	19	29	4	0	52
Lapointe	24	22	15	1	62
Total	43	51	19	1	114

Re-processing

Both studies need to be pre-processed differently.

The Singh study used the old MAS4.0 algorithm, which gives inferior estimates of gene expression. We reprocessed the CEL files using the PM-only model in dChip version 1.3, which should do reasonably well with 52 cancer and 50 normal samples. After processing, we computed the log intensities.

The Lapointe study used global normalization. The M-vs-A plots suggest here that a loess normalization would be more appropriate, and so we applied loess before computing log ratios.

Gene joining

A critical task when combing two studies from different platforms is to determine which genes are measured on both platforms.

Using the GenBank identifiers and the Affymetrix probe-set identifiers, we updated the annotations on both platforms to use UniGene buiild 170 (July 2004). Probes targeting the same UniGene cluster were assumed to be measuring the same gene on both platforms.

We found 8054 probe-sets on the Affyemtrix U95A that matched with 11,596 clones onthe Stnaford glass mivroarrays, representing 6204 distinct UniGene clusters.

Everything is relative

At any rate, all microarray measurements are relative.

So, we can't directly combine glass array data (represented as log ratios to a reference sample) to Affymetrix data (represented as log intensities using different probes).

Although we're interested in comparing cancer subtypes, we have measurements of normal prostate on each platform. Thus, we can adjust the measurements on both platforms to be relative to the same thing.

Standardized gene expression

On each platform, we define

$$S_i = \frac{X_i - \mu_{normal}}{\sigma_{normal}}.$$

Here X_i represents the vector of gene measurements in sample i. We estimate the mean μ and standard deviation σ of the normal prostate samples on each platform, and we standardize trhe measurements to ensure that the normals have mean zero and standard deviation one. We refer to the resulting S_i values as standardized gene expression measurements.

Note: There are multiple probes for many UniGene clusters within each platform. After standardizing, we average these measurements and then re-standardize.

Catching our breath

At this point, we can cobine the prostate cancer data from the two platforms. We have a data matrix of size 6204×62 from the Lapointe study that includes 62 cancer samples, and a data matrix of size 6204×52 from the Singh study that includes 52 cancer samples. We also know the Gleason score of all 114 cancer samples.

How do we build a model to predict Gleason score from gene expression?

Logistic Regression

Our goal is to predict “good” or “poor” prognosis in terms of gene expression, where “good” means Gleason 6 or lower and “poor” means Gleason 8 or higher. So, we actually have a binary outcome variable to predict. Numerically, we can code this outcome as a value of 0 (good) or 1 (poor).

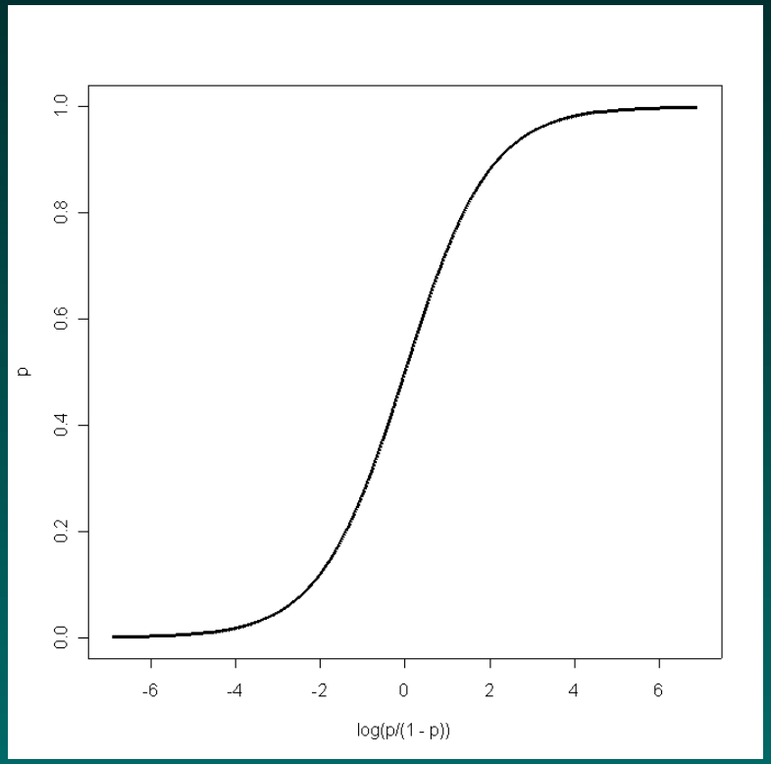
A favorite statistical tool for a wide variety of problems is to build a linear model. It’s hard, however, to get linear functions to restrict their output values to 0’s and 1’s.

So, we first generalize and think about the outcome as the probability of a poor outcome. That helps a little bit, since probabilities can be any continuous value between 0 and 1. But there is still the problem that linear functions will, sooner or later, extend outside of any fixed interval.

Logistic functions, or the logit transformation

Next,we transform the outcome so it spreads out across the entire range of real numbers:

$$y = log(\frac{p}{1-p}).$$



Almost there

Now given any set of covariates X_1, X_2, ..., X_n, we can build a predictive linear model of the form

$$y = log(\frac{p}{1-p}) = X_1\beta_1 + \cdots X_n\beta_j.$$

There is, of course, one minor technical glitch here. The value $p = 0$ corrsponding to an observed "good" prognosis gets mapped to negative infinity, and the value $p = 1$ corresponding to an observed "poor" prognosis gets mapped to positive infinity. This difficulty prevents us from using the standard techniques to fit the model from the data. Instead, we have to resort to using iteratrives methods to find the maximum likelihood estimates.

Logistic regression in R

Fortunately, we don't have to concern ourselves with the details of finding the maximum likelihood estimates, since the iterative procedure is already coded in an R function. Let's assume we have put together a data frame (`my.data`) containing three columns:

X The standardized expression levels of a single gene for all samples

G A binary indicator of Gleason score (0 = good, 1 = poor)

S A binary indicator of the study (0 = Lapointe, 1 = Singh)

Logistic regression in R

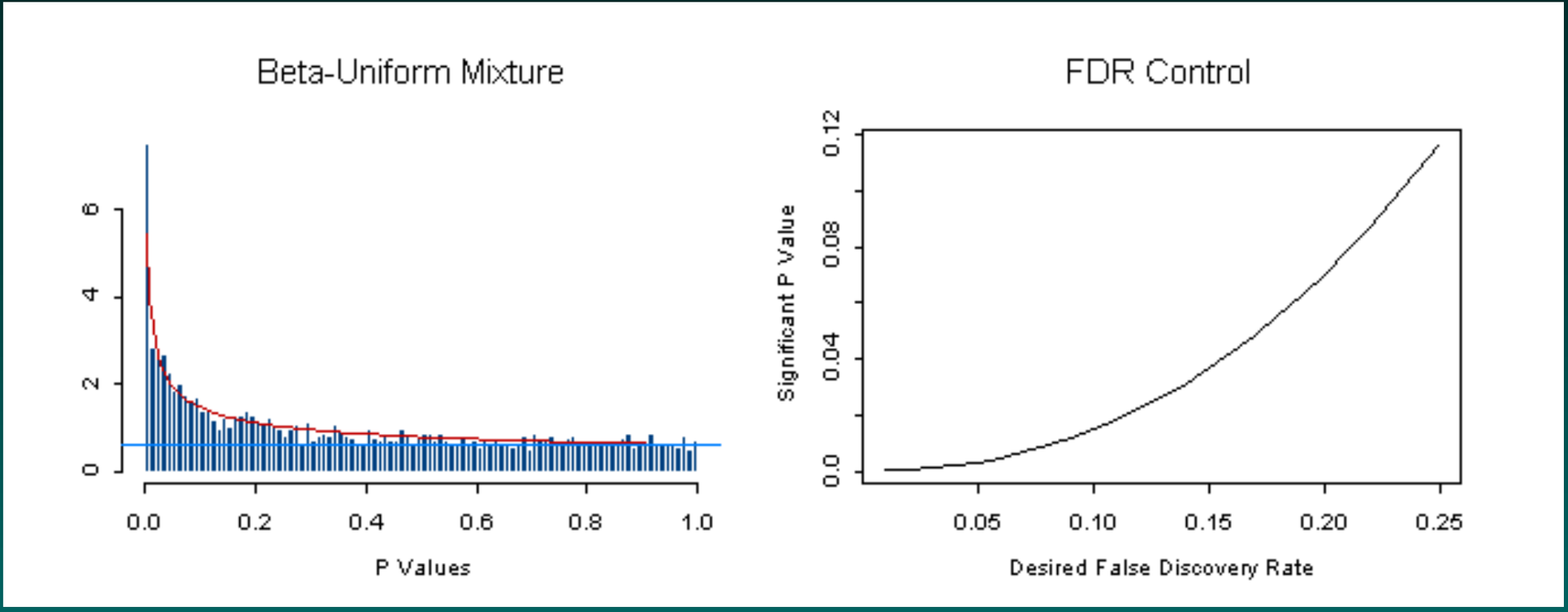
Then we can fit a logistic regression model that predicts Gleason score from gene expression, allowing for a study effect, by

```
> logreg <- glm(G ~ S + X, data = my.data,
+   family = binomial)
```

A p-value associated with this model tells us how well it describes the data; separate p-values can be computed for the significance of the study effect or of the gene expression as a contributor to the prediction of the Gleason score. (Note: study had no efffect on Gleason score.) We computed logistic regression models for each of the 6402 UniGene clusters measured on both studies, and computed a p-value for the significance of each model.

BUM and logistic regression

To account for multiple testing, we modeled the distribution of all the p-values using a beta-uniform mixture (BUM), the same method we used for differential expression.



Combining multiple genes in a single model

So, we now have some evidence that there are genes that make a nontrivial contribution to our ability to predict Gleasoin score. How do we put together a single model that combines information across genes?

First, we arbitrarily restricted ourselves to the 20 genes that had the most significant p-values from the logistic regression models using only one gene at a time. We put together a data frame (`top.twenty`) that combined the Gleason score, the study effect, and the standardized expression values of the 20 genes. We also constructed a logistic regression model that included all twenty genes:

```
> logreg <- glm(G ~ ., data = top.twenty,
+   family = binomial)
```

Akaike information criterion

Nested models using different explanatory variable can be compared using the Akaike Information Criterion:

$$AIC = -2\text{max log likellihood} + 2\text{No. parameters.}$$

A smaller value of the AIC is better. The AIC uses the number of parameters as a penalty term. If two models explain the data equally well, then the model with fewer parameters (or fewer explanatory variables) is preferred.

R includes an automated procedure to discard genes that are not contributing to the model, based on the AIC:

```
> best.model <- step(logreg)
```

The seven-gene predictor

We ran this procedure on the compined prostate cancer data sets to predict Gleason good (6 or lower) or Gleason poor (8 or higher).

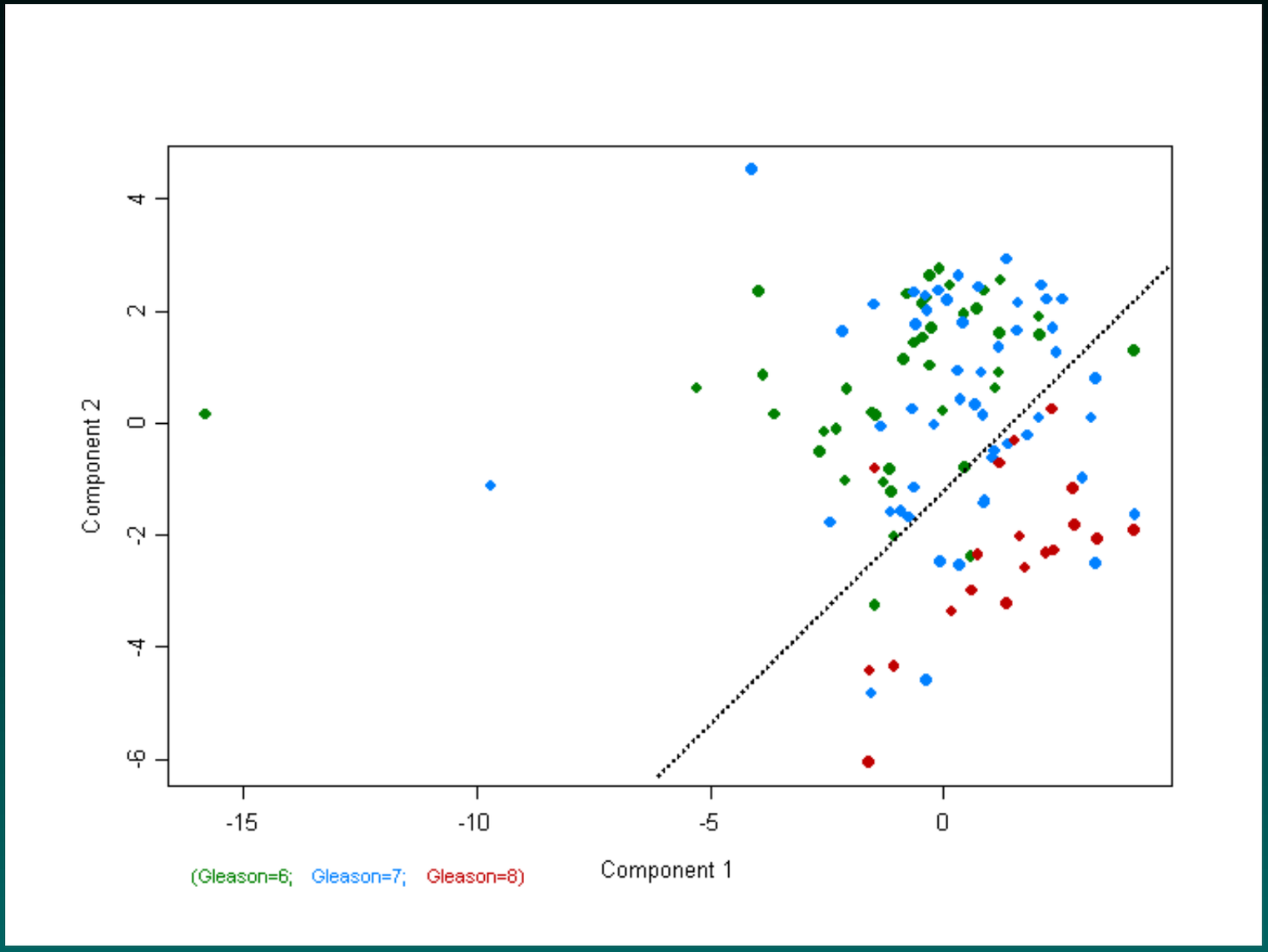
- Got a seven-gene model:

  LTBP2 latent transforming growth factor beta binding protein 2
  TIMP2 tissue inhibitor of metalloproteinase 2
  CDH11 cadherin 11, type 2, OB-cadherin (osteoblast)
  RAP140 retinoblastoma-associated protein 140
  ProSAPiP2 ProSAPiP2 protein
  CXCR4 chemokine (C-X-C motif) receptor 4
  SEPT6 Septin 6

Three of these genes (TIMP2, CDH11, CXCR4) have been

previously reported to be related to prognosis in prostate cancer.

PCA



Prediction probabilities