♠ Outcomes and classification probabilities

- Expression level vector $\mathbf{x}_j$ on array (tumor sample, etc) j
- Binary outcome: $z_j = 0$ or 1
  - codes a clinical or physiological endpoint, state or outcome
  - e.g., primary breast cancer ER+ ($z_j = 1$) versus ER− ($z_j = 0$)
- Probability model estimates $\pi_j = Pr(z_j = 1)$ based on available information and data
  - Fit a model to estimate the π_j in training sample $j = 1, \ldots, n$
    classification, discrimination of cases in training sample
  - Evaluate/estimate π_j for validation cases $j = n+1, n+2, \ldots$
    predictive classification, validation, prognosis

♠ Binary regression models

- Linear regression model based on regression functions $\mu_j = \mathbf{x}_j'\boldsymbol{\beta}$
  - linear combinations, linear scoring, of expression levels of genes
  - $p-$vector of regression parameters $\boldsymbol{\beta}$, one for each gene
    (plus intercept term β_0)
- Idea: model π_j as a function of $\mathbf{x}_j$ in a similar fashion
  - But $0 < \pi_j < 1$ for probability, and μ_j is real-valued
  - Need truncation or transformation
  - Standard statistical models transform from real-value to $(0, 1)$ using a specified non-linear function: mapping μ_j to π_j
- Logistic regression:
$$\pi_j = 1/(1 + \exp(-\mu_j))$$
- Probit regression:
$$\pi_j = \Phi(\mu_j)$$
  where Φ is standard normal cumulative distribution function
- Others ... all similar in form (any continuous distribution function does the trick)

♠ Probit models

- One nice, and important, interpretation: *Latent threshold for 0/1*
- Multiple regression outcome $y_j = \mathbf{x}_j'\boldsymbol{\beta} + \epsilon_j$ on array j
  i.e., $y_j = \mu_j + \epsilon_j$
- with a standard Gaussian (or normal) error term $\epsilon_j \sim N(0, 1)$
- y_j is *latent* – i.e., not observed, unknown, hidden
- The probability that y_j is positive is $\pi_j = \Phi(\mu_j)$
- A "hidden" underlying threshold mechanism in which a (weighted, super-gene measure of) expression levels determine the probability of outcome
- Also, $z_j = 1$ if and only if $y_j > 0$
  - latent variable is positive for ER+ cases, negative for ER− cases
  - could precisely classify cases if we could observe the latent y_j, but we do not; result is the binary probability model

♠ SVD regression and Bayesian analysis

- Dimension problem: $p = 000$'s of genes, n =few microarrays. Ill-posed estimation problem – many more variables than data points. May use SVD regression ideas, to map to factor regression $\mathbf{x}_j'\boldsymbol{\beta} = \mathbf{f}_j'\boldsymbol{\theta}$ (see Note 6) using the SVD analysis of expression data matrix $\mathbf{X}$
- As in Note 6, $\boldsymbol{\theta} = \mathbf{B}'\boldsymbol{\beta}$ or $\boldsymbol{\theta} = \mathbf{DA}'\boldsymbol{\beta}$
- Multiple regression on the factor variables themselves as predictors
- n predictor variables, not p
- Regression parameter vector $\boldsymbol{\theta}$ to estimate
- Dimension reduction of inference/estimation problem when $p > n$, as is the case in gene expression analyses
- Formal inference and prediction can be based on Bayesian analysis and its implicit stochastic regularisation of the estimation problem
  - remove some of the "least variable" factors
  - apply Bayesian analysis to the rest
- Corresponding estimation of $\boldsymbol{\beta}$ via $\boldsymbol{\beta} = \mathbf{AD}^{-1}\boldsymbol{\theta}$

♠ Software, Computation and Summary

- Point estimate analysis: iterative computation of estimates of $\boldsymbol{\theta}$ that are Bayesian *posterior modes* (EM algorithms, MAP estimation)
  - Choose a subset of genes to use in $\mathbf{X}$
  - e.g., screen genes to choose the "top 100" in terms of raw sample correlation with ER or other binary outcome
  - Fit model on this reduced subset, using SVD regression
  - Point estimates of $\boldsymbol{\theta}$ and corresponding $\boldsymbol{\beta}$
- Full Bayesian analysis using stochastic simulation methods (Markov chain Monte Carlo simulation, Gibbs sampling)
  - iterative computation of *simulation samples* of values of $\boldsymbol{\theta}$ whose distribution can be summarised to represent the information in the data about $\boldsymbol{\theta}$ in terms of point estimates (the average sample value, for example), and probability intervals (taking fractiles/percentiles of the sample values). Map these values to corresponding values of $\boldsymbol{\beta}$ to summarise too – effects of individual genes on analysis
- Estimated or fitted classification probabilities in sample: $\pi_j = \Phi(\mu_j)$ with $\mu_j = \mathbf{x}_j'\boldsymbol{\beta}$
  - point estimate $\hat{\boldsymbol{\beta}}$ implies estimate $\hat{\mu}_j$ and hence estimates of probabilities $\hat{\pi}_j = \Phi(\hat{\mu}_j)$
  - simulation samples of values of $\boldsymbol{\beta}$ imply corresponding simulation samples of values of μ_j and hence of the probabilities π_j. Summarise by averages, percentiles for interval estimates, etc

♠ Cross-validation and Prediction

- Include "validation samples" to be predicted to assess the realistic utility of the model in "forecasting" the probabilities of 0/1 outcomes for new cases: inference on π_j for $j = n+1, n+2, ...$ etc
- Also useful to explore *One-at-a-time cross-validation* studies, also known as *Hold-one-out* studies
  - Hold out sample $i = 1$, and fit model to cases $2, \ldots, n$
  - Estimate/infer π_1 to to assess how well case 1 is predicted based on the others
  - Repeat for case $i = 2$, then $i = 3$, and so forth
- Formal and "honest" assessment of model fit and adequacy
- Identifies "interesting" cases, those that fit least well
- Reflects real-life context of application of models
- n.b., if screening genes to select a subset, must do so separately in each CV analysis