L14

Mass Spec Quantitation
MS applications
Microarray analysis
Counting transcripts

- cDNA from the cell hybridizes to complementary DNA fixed on a ‘chip’.
- The intensity of the signal is a ‘count’ of the number of copies of the transcript
Our Goal is to construct a matrix as shown for proteins, and RNA, and use it to identify differentially expressed transcripts/proteins.
Gene Expression

• Measuring expression at transcript level is done by micro-arrays and other tools.
• Expression at the protein level is being done using mass spectrometry.
• Two problems arise:
  – Data: How to populate the matrices on the previous slide? (‘easy’ for mRNA, difficult for proteins)
  – Analysis: Is a change in expression significant? (Identical for both mRNA, and proteins).
• We will consider the data problem here. The analysis problem will be considered when we discuss micro-arrays.
The Biological Problem

• Two conditions that need to be differentiated, (Have different treatments).
  • EX: ALL (Acute Lymphocytic Leukemia) & AML (Acute Myelogenous Leukima)
• Possibly, the set of expressed genes is different in the two conditions
Supplementary fig. 2. Expression levels of predictive genes in independent dataset. The expression levels of the 50 genes most highly correlated with the ALL-AML distinction in the initial dataset were determined in the independent dataset. Each row corresponds to a gene, with the columns corresponding to expression levels in different samples. The expression level of each gene in the independent dataset is shown relative to the mean of expression levels for that gene in the initial dataset. Expression levels greater than the mean are shaded in red, and those below the mean are shaded in blue. The scale indicates standard deviations above or below the mean. The top panel shows genes highly expressed in ALL, the bottom panel shows genes more highly expressed in AML.
Gene Expression Data

- Gene Expression data:
  - Each row corresponds to a gene
  - Each column corresponds to an expression value

- Can we separate the experiments into two or more classes?

- Given a training set of two classes, can we build a classifier that places a new experiment in one of the two classes.
Three types of analysis problems

- Cluster analysis/unsupervised learning
- Classification into known classes (Supervised)
- Identification of “marker” genes that characterize different tumor classes
Supervised Classification: Basics

- Consider genes $g_1$ and $g_2$
  - $g_1$ is up-regulated in class A, and down-regulated in class B.
  - $g_2$ is up-regulated in class A, and down-regulated in class B.
- Intuitively, $g_1$ and $g_2$ are effective in classifying the two samples. The samples are linearly separable.
• With 3 genes, a plane is used to separate (linearly separable samples). In higher dimensions, a hyperplane is used.
Non-linear separability

- Sometimes, the data is not linearly separable, but can be separated by some other function.
- In general, the linearly separable problem is computationally easier.
Formalizing of the classification problem for micro-arrays

- Each experiment (sample) is a vector of expression values.
  - By default, all vectors $\mathbf{v}$ are column vectors.
  - $\mathbf{v}^T$ is the transpose of a vector
- The genes are the dimension of a vector.
- Classification problem: Find a surface that will separate the classes
Formalizing Classification

- Classification problem: Find a surface (hyperplane) that will separate the classes.
- Given a new sample point, its class is then determined by which side of the surface it lies on.
- How do we find the hyperplane? How do we find the side that a point lies on?

<table>
<thead>
<tr>
<th>$g_1$</th>
<th>1</th>
<th>.9</th>
<th>.8</th>
<th>.1</th>
<th>.2</th>
<th>.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_2$</td>
<td>.1</td>
<td>0</td>
<td>.2</td>
<td>.8</td>
<td>.7</td>
<td>.9</td>
</tr>
</tbody>
</table>

CSE182
Basic geometry

• What is $||\mathbf{x}||_2$?
• What is $\mathbf{x}/||\mathbf{x}||$?
• Dot product?

$x=(x_1,x_2)$

$$x^T y = x_1 y_1 + x_2 y_2 = ||x|| \cdot ||y|| \cos \theta_x \cos \theta_y + ||x|| \cdot ||y|| \sin \theta_x \sin \theta_y + ||x|| \cdot ||y|| \cos (\theta_x - \theta_y)$$
Dot Product

• Let $\beta$ be a unit vector.
  – $||\beta|| = 1$
• Recall that
  – $\beta^T x = ||x|| \cos \theta$
• What is $\beta^T x$ if $x$ is orthogonal (perpendicular) to $\beta$?
Hyperplane

• How can we define a hyperplane L?

• Find the unit vector that is perpendicular (normal to the hyperplane)
Points on the hyperplane

- Consider a hyperplane $L$ defined by unit vector $\beta$, and distance $\beta_0$
- Notes:
  - For all $x \in L$, $x^T \beta$ must be the same, $x^T \beta = \beta_0$
  - For any two points $x_1, x_2$,
    - $(x_1 - x_2)^T \beta = 0$
Hyperplane properties

- Given an arbitrary point $x$, what is the distance from $x$ to the plane $L$?
  \[ D(x,L) = (\beta^T x - \beta_0) \]
- When are points $x_1$ and $x_2$ on different sides of the hyperplane?
Separating by a hyperplane

- Input: A training set of +ve & -ve examples
- Goal: Find a hyperplane that separates the two classes.
- Classification: A new point $x$ is +ve if it lies on the +ve side of the hyperplane, -ve otherwise.
- The hyperplane is represented by the line
- $\{x:-\beta_0+\beta_1x_1+\beta_2x_2=0\}$
Error in classification

• An arbitrarily chosen hyperplane might not separate the test. We need to minimize a misclassification error.
• Error: sum of distances of the misclassified points.
• Let $y_i = -1$ for +ve example $i$, $y_i = 1$ otherwise.

$$D(\beta, \beta_0) = \sum_{i \in M} y_i \left( x_i^T \beta + \beta_0 \right)$$

• Other definitions are also possible.
EXTRA SLIDES
Gradient Descent

- The function $D(\beta)$ defines the error.
- We follow an iterative refinement. In each step, refine $\beta$ so the error is reduced.
- Gradient descent is an approach to such iterative refinement.

$$\beta \leftarrow \beta - \rho \cdot D'(\beta)$$
Rosenblatt’s perceptron learning algorithm

\[ D(\beta, \beta_0) = \sum_{i \in M} y_i (x_i^T \beta + \beta_0) \]

\[ \frac{\partial D(\beta, \beta_0)}{\partial \beta} = \sum_{i \in M} y_i x_i \]

\[ \frac{\partial D(\beta, \beta_0)}{\partial \beta_0} = \sum_{i \in M} y_i \]

⇒ Update rule: 

\[
\begin{pmatrix}
\beta \\
\beta_0
\end{pmatrix} = \begin{pmatrix}
\beta \\
\beta_0
\end{pmatrix} - \rho \begin{pmatrix}
\sum_{i \in M} y_i x_i \\
\sum_{i \in M} y_i
\end{pmatrix}
\]
Classification based on perceptron learning

• Use Rosenblatt’s algorithm to compute the hyperplane $L=(\beta,\beta_0)$.
• Assign $x$ to class 1 if $f(x) \geq 0$, and to class 2 otherwise.
Perceptron learning

• If many solutions are possible, it does no choose between solutions
• If data is not linearly separable, it does not terminate, and it is hard to detect.
• Time of convergence is not well understood
Linear Discriminant analysis

- Provides an alternative approach to classification with a linear function.
- Project all points, including the means, onto vector $\beta$.
- We want to choose $\beta$ such that
  - Difference of projected means is large.
  - Variance within group is small
\[ \tilde{m}_1 = \frac{1}{n_1} \sum_x \beta^T x = w^T m_1 \]

Scatter between samples:
\[ |\tilde{m}_1 - \tilde{m}_2|^2 = \left| \beta^T (m_1 - m_2) \right|^2 \]
\[ |\tilde{m}_1 - \tilde{m}_2|^2 = \beta^T S_B \beta \]

scatter within sample: \( \tilde{s}_1^2 + \tilde{s}_2^2 \)

where,
\[ \tilde{s}_1^2 = \sum_y (y - \tilde{m}_1)^2 = \sum_{x \in D_1} (\beta^T (x - m_1))^2 = \beta^T S_1 \beta \]
\[ \tilde{s}_1^2 + \tilde{s}_2^2 = \beta^T (S_1 + S_2) \beta = \beta^T S_w \beta \]

**Fisher Criterion**
\[ \max_{\beta} \frac{\beta^T S_B \beta}{\beta^T S_w \beta} \]
Maximum Likelihood discrimination

- Suppose we knew the distribution of points in each class.
  - We can compute $\Pr(x|\omega_i)$ for all classes $i$, and take the maximum
ML discrimination

- Suppose all the points were in 1 dimension, and all classes were normally distributed.

\[ P(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)} \]

\[ \Pr(\omega_i \mid x) = \frac{\Pr(x \mid \omega_i) \Pr(\omega_i)}{\sum_j \Pr(x \mid \omega_j) \Pr(\omega_j)} \]

\[ g_i(x) = \ln(\Pr(x \mid \omega_i)) + \ln(\Pr(\omega_i)) \]

\[ \cong \frac{-(x - \mu_i)^2}{2\sigma_i^2} + \ln(\Pr(\omega_i)) \]
ML discrimination recipe

• We know the distribution for each class, but not the parameters
• Estimate the mean and variance for each class.
• For a new point \( x \), compute the discrimination function \( g_i(x) \) for each class \( i \).
• Choose \( \text{argmax}_i g_i(x) \) as the class for \( x \)