Bioinformatics – Lecture Notes

Announcements

Reference: Microarray Data Analysis and Visualization by Arun Jagota

2 days of classes including today.

Final Projects due May 2 (Thursday).

Class 29 – April 30, 2002

I. Identifying genes expressed differently in paired samples

We want to find our if a gene x is expressed sufficiently differently in a set of patients before and after treatment so that we can infer that gene x is expressed differently in the population before and after treatment.

Assume that we have n patients with $e_{x,b,i}$ and $e_{x,a,i}$ being the before and after treatment expression levels for a gene x in patient i. Let $d_{x,i} = e_{x,b,i} - e_{x,a,i}$ be the difference between these two levels for patient i. In the whole population the mean difference is $\overline{d}_x = 1/n \sum_i d_{x,i}$.

The question can be rephrased as is $\overline{d}_x$ sufficiently different from zero for us to conclude that at the desired significance level gene x is expressed differently before and after treatment in the population. As we have done before, we can apply the z-test or t-test if we assume that the distribution is normally distributed. If we do not make this assumption, we can use the sign test as before.

II. Identifying genes expressed differently in more than two populations

If we are trying to identify genes that are not expressed identically in all populations and we have more than two populations, we have to try other methods than presented earlier. For example, consider the question: Is gene x expressed differently in three groups for us to conclude at the desired significance level gene x is expressed differently in the three populations. To do this, we use statistical inference once again.

Let $\mu_{x,1}$, $\mu_{x,2}$, and $\mu_{x,3}$ be the unknown mean expression level of gene x in the three populations. The null hypothesis is H_0: $\mu_{x,1} = \mu_{x,2} = \mu_{x,3}$. This can be thought of as H_0: $\wedge_{i,j} m_{x,i} = m_{x,j}$ which is a compound hypothesis that consists of three simple hypotheses. Hence, we can use the methods described earlier for determining if any genes are expressed differently in two populations. The problem with this approach is that the compound test needs to use a more stringent confidence level for each of the simple tests than the desired level for the

compound test. Therefore, we can also use another method called ANOVA (analysis of variance).

You might remember ANOVA from your statistics class and can get quite complicated. We will focus on the most basic approach, i.e. the single-factor ANOVA to answer the question with which we started the lecture.

The first step is to arrange the data into a table.

Factor Level	Cancer Subtype 1	Cancer Subtype 2	Cancer Subtype 3	
	$e_{x,1,1}$	$e_{x,1,2}$	$e_{x,1,3}$	
	.	.	.	
Data	.	.	.	
	.	.	.	
	$e_{x,n_1,1}$	.	$e_{x,n_3,3}$	
		$e_{x,n_2,2}$		
	$E_{x,1}$	$E_{x,2}$	$E_{x,3}$	E_x

where $e_{x,i,j}$ is the expression level of gene x in sample i in group j which is defined by the cancer subtype. The three groups have different numbers of samples, n_1, n_2, n_3. The quantities $E_{x,j}$ are the column sums

$$E_{x,j} = \sum_{i=1}^{n_j} e_{x,i,j} \qquad \text{with j=1, 2, or 3}$$

E_x is the sum of the column sums i.e. $E_x = E_{x,1} + E_{x,2} + E_{x,3}$. We also define

$$E_x^2 = \sum_{j=1}^{3} \sum_{i=1}^{n_j} e_{x.i.j}^2$$

The inter-group variation is

$$\left(\frac{E_{x,1}^2}{n_1} + \frac{E_{x,2}^2}{n_2} + \frac{E_{x,3}^2}{n_3} \right) - \frac{(E_x^2)}{n}$$

and the intra-group variation is

$$E_x^2 - \left(\frac{E_{x,1}^2}{n_1} + \frac{E_{x,2}^2}{n_2} + \frac{E_{x,3}^2}{n_3} \right)$$

where $n = n_1 + n_2 + n_3$ is the total number of samples. The degrees of freedom are

$$\text{df(inter-group)} = \text{\# groups} - 1 = 2$$

$$df(\text{intra-group}) = n - \#\,\text{groups} = n - 3$$

The variances are given by

$$\text{inter-group variance} = \text{inter-group variation}/df(\text{inter-group})$$
$$\text{intra-group variance} = \text{intra-group variation}/df(\text{intra-group})$$

If inter-group variance is much larger than theintra-group variance, we can reject the null hypothesis (that the group means are identical) at a certain high level of confidence. This is done formally using the F-test, with the F-statisitic

$$F = \frac{\text{inter - group variance}}{\text{intra - group variance}}$$

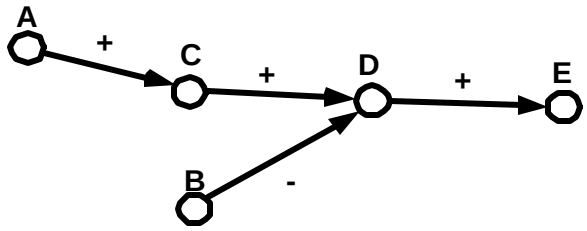
Thus, if $F \geq F(df(\text{inter-group}),df(\text{intra-group}),\alpha/2)$ or
if $1/F \geq F(df(\text{inter-group}),df(\text{intra-group}),\alpha/2)$ H_0 is rejected with confidence $1-\alpha$. Otherwise, we fail to reject H_0.

III. Gene regulation networks

Another application of microarray data is to try to infer possible gene regulation networks.

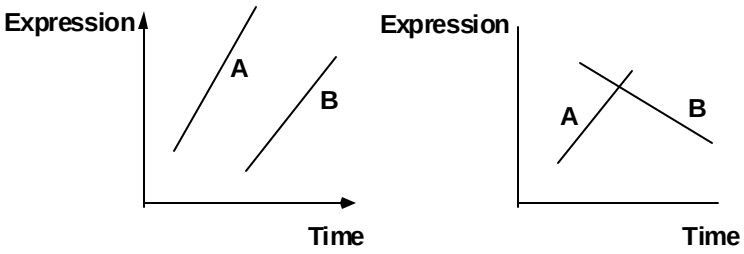
A. Combinatorial approach for parameter estimation

Let A and B be two genes. For example, A might be an activator for B, A might be an inhibitor of B, or neither. Consider the following problem. Given a set of genes {1, 2, …, n} and a set of measurements of their expression levels at various time points after some initial time at which there was some perturbation of the system (a stimulus), we want to infer the structure of the gene network. If we make a connected graph with the nodes being the different genes, the connections will be labeled with '+' or '-' to denote activation or inhibition, respectively



To do this we look for the following. If B's expression level starts to rise shortly after A's does, A might be an activator of B. If B's expression level starts to fall shortly after A's start's rising, then A might be an

inhibitor of B. Of course, it might just be coincidence in which case we will get a false positive.



To avoid the false positives, we need to impose constraints on the system such as requiring that the system be sufficiently sparse (which may or may not work – we cannot tell). Another possible constraint is to require that any one gene cannot act as both an activator and an inhibitor (Chen, Fikov, and Skiena). These authors justify this on the basis that activation and inhibition involve different biological mechanisms.

Chen and co-workers take this constraint and define an optimization problem that takes a network of many possible activation and inhibition paths and removes the edges to enforce the above constraint. They do this by assigning each vertex as an activator '+' or inhibitor '-' while maximizing the number of vertices with both and incoming '+' edge and an incoming '-' edge under the constraint that the edges leaving a vertex has the same sign as the vertex. This is a NP-hard problem and is intractable for a large number of genes (thousands)

To make this problem tractable, we need to reduce the number of genes considered by some preprocessing approach. One possible method to do so is called *varation filtering*. In this process, those genes whose expression levels to not change with time are removed from the data set. Another possible preprocessing method is to cluster the genes conservatively so that all genes in a cluster have very similar expression profiles. All the expression data of genes in this cluster would be replaced by one piece of expression data for the cluster. The network would be performed on the cluster data and give a network representing the regulation of clusters of genes.

B. Modeling dynamics for parameter estimation

Assume that we know that a small set of genes G = {1, 2, …, n} form a gene regulation network, but we do not know the network topology or the strengths of the interactions between the genes. This model can be fitted by a recurrent neural net governed by a set of n nonlinear coupled differential equations.

$$\tau_i \frac{dx_i}{dt} = g\left(h_i + \sum_j w_{ij} x_j \right) - \lambda_i x_i \qquad \text{i = 1, 2, … n}$$

where $x_i(t)$ is the expression level of gene i at time t, w_{ij} is the strength and nature of the influence of gene j on gene i, λ_i is the spontaneous decay rate of x_i, τ_i is a time scaling factor, h_i represents some time varying influence of an external perturbation to the system and g is a sigmoid function to scale its argument (as described earlier). The network will learn the parameters τ_i, λ_i, and w_{ij} from the microarray data.

Solving the model will give expression time courses for the different genes. These can be compared to the data gene expression time courses. The sum of the squared errors (or mean square error) should be minimized by varying the parameters until convergence is reached. An additional constraint to cause a sparse network would be to minimize $\sum_{ij} w_{ij}^2$.

C. Regression for parameter estimation

We can also consider a discrete version of the system of differential equations described above

$$x_i(t + \Delta t) = g\left(b_i + \sum_j w_{ij} x_j \right) \qquad \text{i = 1, 2, … n}$$

where b_i are the bias and w_{ij} are the weights. We solve $y = Wx + b$ for W and b by using $\{x = e(t)$ and $y = e(t+\Delta t)\}$. These two parameters are determined by minimizing using the least-squared error approach. To do this we need to have at least as many time points as genes considered to keep the problem from being under constrained.

D. Bayesian networks for parameter estimation

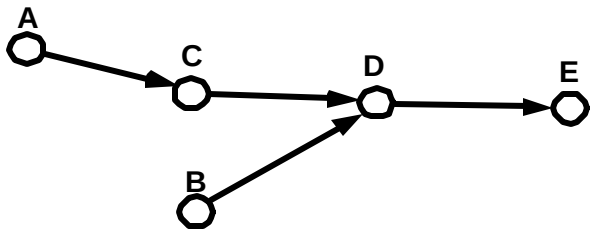
Assume that we have n genes with binary expression data, that is, $x_i = 1$ means a gene is up-regulated and $x_i = 0$ means that a gene is down regulated. Assume that we know the network and that the interactions are probabilistic. For example, if gene C is off and gene B is on, there is a higher probability of gene D being turned off than when gene C is on and gene B is off. Hence, each node of the Bayesian network will contain a conditional probability distribution

$$\Pr[X_i \mid \text{parents of i}]$$

where "parents of i" are the nodes with a directed edge coming into note i. The Bayesian network will be the joint distribution over all the variables in the graph

$$\Pr[X_1, X_2, \ldots, X_n] = \prod_{i=1}^{n} \Pr[X_i \mid \text{parents of } i]$$

From the network



$$\Pr[A, B, C, D, E] = \Pr[A]\Pr[C|A]\Pr[B]\Pr[D|C,B]\Pr[E|D]$$

where the node name denotes the 0,1 random variable.

The parameters that define the Bayesian network have to be determined. Assume that the expression data is of the form

$$\{e_i(t) \in [0, 1], \quad t = 1, 2, \ldots, m \text{ and } i = 1, 2, \ldots, n$$

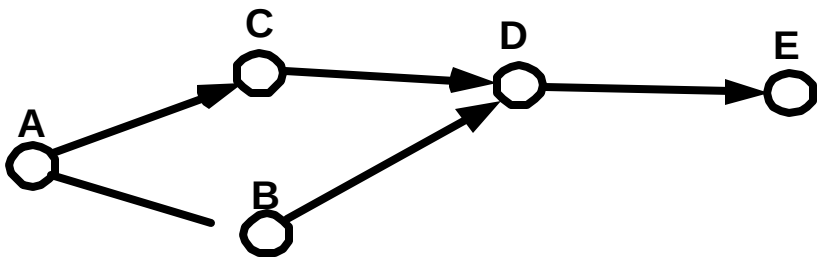
Assume that the parents of node i at time t affect node i's value at time t+1. We can now determine and table $\Pr[x_i(t+1)|$ parents of i at time t] by counting the occurrences of the relevant events in the data. For example, Pr[D(t+1) | B(t), C(t)]

B(t), C(t)→	00	01	10	11
D(t+1)↓				
0	$\frac{a = \#t : (B(t) = 0, C(t) = 0, D(t+1) = 0)}{a+b}$	.	.	.
1	$\frac{b = \#t : (B(t) = 0, C(t) = 0, D(t+1) = 1)}{a+b}$	.	.	.

A Bayesian network may be used to factor a joint distribution. *Causal* Bayesian networks can also be used to infer gene network structure. This has the added property that the parents of a node are its direct causes. If we consider a joint distributions Pr(X,Y) in a Bayesian network, it can either imply $X \to Y$ or $X \leftarrow Y$. If we want only $X \to Y$ we need to use a causal Bayesian network. In fact, the figure show before was a causal Bayesian network.

For example, assume that two genes X and Y have very similar expression patterns. We want to determine whether X regulates Y or Y regulates X but do not have the experimental data to differentiate between the two possibilities. One way of dealing with this is to gather more data. If we can force X on and see how it affects Y. Similarly, we can turn Y on and see how if affects X. Thus, if forcing X on has a great effect on Y but the opposite is not true, we can infer that X regulates Y.

However, this regulation might be direct or indirect. X might regulate Y through its effects on one or more other factors that each can directly or indirectly affect Y. In our example network below, turning C on can regulate both D and E.



$$\Pr[E = e \mid C = on] =$$
$$\sum_{a,b,d} \Pr[E = e \mid D = d] \Pr[D = d \mid B = b, C = on] \Pr[B = b \mid A = a] \Pr[A = a]$$

If we consider the network $X \rightarrow Y$, there is *forced* and *unforced conditioning*. By Bayes rule, for unforced conditioning on Y = on

$$\Pr[X = x \mid Y = on) = \frac{\Pr[Y = on \mid X = x] \Pr[X = x]}{\sum_{x'} \Pr[Y = on \mid X = x'] \Pr[X = x']}$$

For forced conditioning

$$\Pr[X = x \mid Y = on) = \Pr[X = x]$$

The right hand side quantities are easily computable from the parameters stored in the network.