Introduction to Computational Biology
Lecture # 21: Microarray - cont’

Yonatan Vaizman
28/05/2007

1 What else can we do with Microarray technology?

In the previous lesson we discussed using Microarrays to measure or compare levels of RNA expression in a tissue. What else can we do with this technology?

2 Testing DNA levels instead of RNA levels

Instead of transcripting RNA from the tissue and incubating the cDNA with the array, we can use a whole genome, when the source is DNA and not RNA. We can produce the whole genome DNA from all the cells in the tissue, clean it from proteins, and slice it to short sequences. On the array we can place probes from exons, each appearing only once in the genome. That way we can produce genomic DNA from different people (or other organisms), and compare the number of genomic copies for each exon. This method is called aCGH - array complete genome hybridization. Do we expect to see unity?

2.1 Studying pathological cases

It is possible to use aCGH to compare a genome from a tumour to that of a healthy male (which has both X and Y chromosomes). Most cancers involve silencing of genes. This can either be caused by mutations, DNA modifications (such as Methylation, Acetilation) or extraction of the gene from the genome. Gene shortage can be noticed with aCGH.

A tumour is usually a mosaic of many variants competing over the resources, so it is possible to see many variations of gene shortage in samples of a certain kind of cancer. However, if we identify a repeating pattern of gene shortage in tumours from a certain cancer, this could help understand the cause of this cancer, by assuming that this pattern had some selective advantage in this type of tumour.

2.2 Genetic Polymorphism

Looking for SNPs. Suppose we know that a certain gene has two common versions in the population: one with A in a certain position, and the other with C in the same position. We can prepare probes for the two different versions. Because the difference is little, we should use short probes, about 25 bases long, such that a single base in the middle makes a difference. This kind of usage is very popular now. People have developed arrays representing 500,000 SNPs.

3 Tiling Array

In an expression array we look for exons, and that’s what we place in the spots. In a SNP array we look for sequences with SNPs, and that’s what we place in the spots. But we can also use the array in a different approach: not looking
for specific sequences that we choose and place in the spots, but rather looking for whatever is in the genome. This is called Tiling.
We can either place probes for places in the genome with 1000 bp intervals, or for places in the genome with 10 bp intervals (between their starting points) and have overlaps (see figure 1).

3.1 Tiling with expressed RNA

If we use this kind of array, and pour on it RNA expressed from a tissue, we can get an RNA map. This is beyond expecting certain RNA sequences. With this method we can also get transcripts that we didn’t know before. That way we can discover transcripts that don’t code to protein, or that are silenced by siRNA

3.2 Tiling with protein binding DNA

It is possible to find areas in the genome, that bind to a certain protein. CHIP - chromatin immunoprecipitation: the DNA-protein interactions in the culture are fixated, the mix is incubated with antibody for the protein, obtaining mostly DNA from sites where the certain protein is bound. When spreading this mixture on a tiling array, we can see the sites in the genome, where the protein has potential to bind.
This can be tested for proteins such as transcription factors, moduls of polymerase, histones or components of the splicosome.

3.3 RIP

RNA immunoprecipitation. The same as CHIP, only looking for RNA molecules that bind to some RNA binding protein, such as translation factors, RNA transport factors.

4 Double stranded DNA probes

We can create an array with double stranded DNA probes, and look for double stranded DNA molecules from the genome, that bind to some factor.
There is a combinatorical problem in creating an array with large coverage. Say we want to cover all the possible sequences 10 bp long. There are over a milion possibilities, and we want to include all of them in the collection of 25 bases long probes.
Notice that this kind of test doesn’t really reveal where in the genome the factor binds, but shows to what sequences (possibly double stranded) the factor can bind.

5 Handling noise

Several stages of the microarray usage process may add an element of noise to the results: the production of RNA or DNA, the hybridization etc. How can we overcome the uncertainty of the noisy results?

<table>
<thead>
<tr>
<th>Tiling probes with overlap</th>
<th>Tiling probes without overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Tiling array. Representing probes along the genome.
5.1 Deliberate redundancy

In tiling array, for instance, an exon can be represented by several probes (see figure 2). The amount of real signal they should get should be identical. These probes report about the same thing (the amount of that exon in the tissue), but they may show different levels of hybridization. Each probe has a different affinity to the mRNA (or cDNA molecule), but all of them will show larger signals for a higher expression level of this gene (see figure 3).

Assumption: for \( i \) - probe, \( j \) - experiment, there are two types of noise - multiplicative and additive.

\[
x_{i,j} = a_i \cdot l_j \cdot \epsilon_{i,j} + \xi_{i,j}
\]

Where \( x \) is the value gotten for probe \( i \) in experiment \( j \), \( a_i \) is the affinity of probe \( i \) to its target gene, and \( l_j \) is the expression level in experiment \( j \). For now we will disregard the additive element. In this assumed model, the number of free parameters is smaller than the data dimension. We would expect that:

\[
\frac{x_{i,j}}{x_{i,k}} \approx \frac{l_j}{l_k}
\]

\[
\frac{x_{i,j}}{x_{k,j}} \approx \frac{a_i}{a_k}
\]

We’ll turn the problem to additive by log:

\[
\log(x_{i,j}) = \log(x_i) + \log(l_j) + \log(\epsilon_{i,j})
\]

and we would want to minimize the noise, so the problem will be:

\[
\text{minimize}_{a_i, l_j} \| \log(x_{i,j}) - (\log(a_i) + \log(l_j)) \|^2
\]

5.2 A Classic Mathematical problem: Linear function with noise

We have a set of \( n \) instances. Each instance has \( m \) \( x \) values, and one \( y \) value:

\[
\forall \ 1 \leq i \leq n \quad y_i = \alpha_1 x_{1,i} + \cdots + \alpha_m x_{m,i} + \epsilon_i
\]

Vectorian form:

\[
\forall \ 1 \leq i \leq n
\]

Figure 2: An exon can be represented by several probes - deliberate redundancy.
\[ y_i = \alpha^T \cdot \overline{x}_i + \epsilon_i \]

Matrician form:
\[ \overline{Y} = X \cdot \overline{\alpha} + \overline{\epsilon} \]

We wish to find the parameters \( \alpha \), that will minimise some loss function:
\[ \min_{\overline{\alpha}} \sum_{i=1}^{n} L(y_i, \overline{x}_i \cdot \overline{\alpha}) \]

In the quadratic case of a loss function:
\[ L(y, z) = \frac{1}{2} (y - z)^2 \]

Finding the minimum by calculating the derivative, and equalize to zero:
\[
\frac{\partial L_{\text{total}}}{\partial \alpha_j} = \sum_{i=1}^{n} x_{j,i} \cdot \frac{\partial}{\partial \alpha_j} L(y_i, \overline{x}_i \cdot \overline{\alpha}) = 0
\]

We have \( m \) linear equations like this, with \( m \) variables.
For our problem:
\[ \log(x_{i,j}) = \log(a_i) + \log(l_j) + \log(\epsilon_{i,j}) \]

The matrix \( X \) is of indicators 0 or 1, and the \( \alpha \)s of the equation are the \( a \)s and \( l \)s.
Figure 3: We would expect each probe to have some affinity to its target, and to respond linearly to target level.