Introduction to gene expression microarray data analysis
Outline

• Brief introduction:
  – Technology and data.
  – Statistical challenges in data analysis.

• Preprocessing – data normalization and transformation.

• Useful Bioconductor packages.
A short history

• Evolved from Southern blotting, which is a procedure to detect and quantify a specific DNA sequence.
• Gene expression microarray can be thought as parallelized Southern blotting experiments.
  – study the expression of 45 Arabidopsis genes.
• Very popular for the past 20 years. Searching “gene expression microarray” on PubMed returns 60,000+ hits.
Still microarray?

• Microarray is still widely used because of lower costs, easier experimental procedure and more established analysis methods.

• Similar problems are presented in newer technologies such as RNA-seq, and similar statistical techniques can be borrowed.
Introduction to GE microarray technology and design
Goal: measure mRNA abundance

DNA (2 copies)

mRNA (multiple copies)

Protein (multiple copies)

The amount of these is easy to measure. And it is positively correlated with the protein amount!

The amount of these matters! But they are difficult to measure.
Gene expression microarray design

- A collection of DNA spot on a solid surface.
- Each spot contains many copies of the same DNA sequence (called “probes”).
  - Probe sequences are designed to target specific genes.
- Genes with part of its sequence complementary to a probe will hybridized on (stick to) that probe.
- The amount of hybridization on each probe measures the amount of mRNA for its target gene.
Experimental procedure

wet lab: perform experiment

dry lab: data analysis
Available platforms

• Affymetrix
• Agilent
• Nimblegene
• Illumina
• ABI
• Spotted cDNA
Affymetrix Gene expression arrays

The Affymetrix platform is one of the most widely used.

http://www.affymetrix.com/
Affymetrix GeneChip array design

Use U133 system for illustration:

- Around 20 probes per gene.
- Not necessarily evenly spaced: sequence property matters.
- The probes are located at random locations on the chip to average out the effects of the array surface.

**Target sequence**

TTAAGTCGTACCCGTGTACGGGCGC

**Perfect match (PM) probe:** measure signals

AATTCAGCATGGGCACATGCCCGCG

**Mis-match (MM) probe:** measure background

AATTCAGCATGGACACATGCCCGCG
One-color vs. two-color arrays

- Two-color (two-channel) arrays hybridize two samples on the same array with different colors (red and green).
  - Each spot produce two numbers.
  - Agilent, Nimblegen
- One-color (single-channel) arrays hybridize one sample per array.
  - Easier when comparing multiple groups.
  - Have to use twice as many arrays.
  - Affymetrix, Illumina.
Data from microarray

- Data are fluorescent intensities:
  - extracted from the images with artifacts (e.g., cross-talk) removed, which involves many statistical methods.
  - Final data are stored in a matrix: row for probes, column for samples.
  - For each sample, each probe has one number from one-color arrays and two numbers for two-color arrays.

<table>
<thead>
<tr>
<th></th>
<th>sample1</th>
<th>sample2</th>
<th>sample3</th>
<th>sample4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1007_s_at</td>
<td>8.575758</td>
<td>8.915618</td>
<td>9.150667</td>
<td>8.967870</td>
</tr>
<tr>
<td>1053_at</td>
<td>6.959002</td>
<td>7.039825</td>
<td>6.898245</td>
<td>7.136316</td>
</tr>
<tr>
<td>117_at</td>
<td>7.738714</td>
<td>7.618013</td>
<td>7.499127</td>
<td>7.610726</td>
</tr>
<tr>
<td>121_at</td>
<td>10.114529</td>
<td>10.018231</td>
<td>10.003332</td>
<td>9.809068</td>
</tr>
<tr>
<td>1255_g_at</td>
<td>5.056204</td>
<td>4.759066</td>
<td>4.629297</td>
<td>4.673458</td>
</tr>
<tr>
<td>1294_at</td>
<td>8.009337</td>
<td>7.980694</td>
<td>8.343183</td>
<td>8.025335</td>
</tr>
<tr>
<td>1316_at</td>
<td>6.899290</td>
<td>7.045843</td>
<td>6.976185</td>
<td>7.063050</td>
</tr>
<tr>
<td>1320_at</td>
<td>7.218898</td>
<td>7.600437</td>
<td>7.433031</td>
<td>7.201984</td>
</tr>
<tr>
<td>1405_i_at</td>
<td>6.861933</td>
<td>6.042179</td>
<td>6.165090</td>
<td>6.200671</td>
</tr>
<tr>
<td>1431_at</td>
<td>5.073265</td>
<td>5.114023</td>
<td>5.159933</td>
<td>5.063821</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Microarray data measure the “relative” levels of mRNA abundance

• Expression levels for different genes on the same array are not directly comparable.
• Expression levels for the same genes from different arrays can be compared, after proper normalization.
• All statistical inferences are for relative expressions, e.g., “the expression of gene X is higher in cancer compared to normal”.
Statistical challenges

• Data normalization: remove systematic technical artifacts.
  – Within array: variations of probe intensities are caused by:
    • cross-hybridization: probes capture the “wrong” target.
    • probe sequence: some probes are “sticker”.
    • others: spot sizes, smoothness of array surface, etc.
  – Between array: intensity-concentration response curve can be different from different arrays, caused by variations in sample processing, image reader, etc.

• Summarization of gene expressions:
  – summarize values for multiple probes belonging to the same gene into one number.

• Differential expression detection:
  – Find genes that are expressed differently between different experimental conditions, e.g., cases and controls.
Gene expression microarray data normalization
Normalization

• Artifacts are introduced at each step of the experiment:
  – Sample preparation: PCR effects.
  – Array itself: array surface effects, printing-tip effects.
  – Hybridization: non-specific binding, GC effects.
  – Scanning: scanner effects.

• Normalization is necessary before any analysis to ensure differences in intensities are due to differential expressions, not artifacts.
Within- and between-array normalization

- Within-array: normalization at each array individually to remove array-specific artifacts.
- Between-array: to adjust the values from different arrays and put them at the same baseline, so that numbers are comparable.
- The methods are different for one- and two-color arrays.
Within array normalization, two-color

• Most common problem is intensity dependent effect: log ratios of intensities from two channels depends on the total intensity.
• Most popular: loess normalization.
MA plot

• Widely used diagnostic plot for microarray data (Yang et al. 2002, *Nucleic Acids Research*).
• Also used for different types of sequencing data.
• For spot $i$, let $R_i$ and $G_i$ be the intensities, define:
  – $M_i = \log_2 R_i - \log_2 G_i$, $A = (\log_2 R_i + \log_2 G_i)/2$.
  – $M$ measures relative expression, $A$ measures total expression.
• Visualize relative vs. total expression dependence.
Loess normalization

• Based on the assumptions that: (1) most genes are not DE (with M=0) and (2) M and A are independent, MA plot should be flat and centered at 0.

• Normalization procedure:
  – Fit a smooth curve of M vs. A using loess, e.g., $M = f(A) + \epsilon$, $f(.)$ is smooth.
  – $M_{\text{norm}} = M - f(A)$
  – loess (lowess): locally weighted scatterplot smoothing.
    – method to fit a smooth curve between two variables.
Loess normalization: before and after
Within array normalization: one-color

• RMA (Robust Multi-array Average) background model (Irizarry et al. 2003, *Biostatistics*).

• Idea: observed intensity $Y$ is composed of the true intensity $S$ (exponentially distributed) and a random background noise $B$ (normally distribute).

• For each array, assume:

  $$Y = S + B$$

  Signal: $S \sim \text{Exp}(\lambda)$

  Background: $B \sim N(\mu, \sigma^2)$ left-truncated at zero
Simple derivation

• Observed: $Y$; of interest: $S$.

• The idea is to predict $S$ from $Y$ using $E[S|Y]$:

$$E[S|Y] = \int s f(s|Y = y) ds = \int s \frac{f(s,Y = y)}{f_Y(y)} ds = \frac{1}{f_Y(y)} \int s f(s,Y = y) ds$$

• The joint: $f(s,Y = y) = f(s,B = y - s) = f_S(s)f_B(y - s)$

• Marginal distribution of $Y$ $f_Y(y)$ can be derived.
An extension to consider probe sequence effects: GCRMA

\[ Y_{gi} = O_{gi} + N_{gi} + S_{gi} \]

\[ = O_{gi} + \exp(\mu_{gi} + \epsilon_{gi}) + \exp(s_{g} + \delta_{g}X_i + a_{gi} + b_i + \xi_{gi}). \]

Here \( Y_{gi} \) is the PM intensity for the probe \( j \) in probeset \( g \) on array \( i \), \( \epsilon_{gi} \) is a normally distributed error that account for NSB for the same probe behaving differently in different arrays, \( s_{g} \) represents the baseline log expression level for probeset \( g \), \( a_{gi} \) represents the signal detecting ability of probe \( j \) in gene \( g \) on array \( i \), \( b_i \) is a term used to describe the need for normalization, \( \xi_{gi} \) is a normally distributed term that accounts for the multiplicative error, and \( \delta_{g} \) is the expected differential expression for every unit difference in covariate \( X \). Notice \( \delta_{g} \) is the parameter of interest. As described by Naef and Magnasco (2003) \( a_{gi} \) is a function of \( \alpha \).

Wu et al. (2005) JASA
Probe sequence effects

- Probe affinity is modeled as:
  \[ \alpha = \sum_{k=1}^{25} \sum_{j \in \{A,T,G,C\}} \mu_{j,k} 1_{b_k=j} \text{ with } \mu_{j,k} = \sum_{l=0}^{3} \beta_{j,l} k^l, \]

- This kind of modeling is widely used in other microarrays and sequencing data!
Summary: within array normalization

- To remove the unwanted artifacts and obtain true signals.
- Performed at each array individually.
- Both MA-plot based normalization and background error models (e.g., RMA) are popular in many other data (other microarrays, ChIP-seq, RNA-seq)
  - Use loess with caution because it assumes most genes are not DE.
  - The error model (additive background, multiplicative error) is very useful.
Between array normalization

- Data from arrays (intensity values) represent mRNA quantities, but the intensity-mRNA quantities response can be different from different arrays. So a number, say, 5, on arrays 1 doesn’t mean the same on array 2.
- This could be caused by:
  - Total amount of mRNA used
  - Properties of the agents used.
  - Array properties
  - Settings of laser scanners
  - etc.
- These artifacts cannot be removed by within array normalization.
- Goal: normalize so that data from different arrays are comparable!
Linear scaling method

- Used in Affymetrix software MAS:
  - Use a number of “housekeeping” genes and assume their expressions are identical across all arrays.
  - Shift and rescale all data so the average expression of these genes are the same across all arrays.
Non-linear smoothing based

• Implemented in dChip (Li and Wong 2001, *Genome Bio.*).
  – Find a set of genes invariant across arrays.
  – Find a “baseline” array.
  – For every other arrays fit a smooth curve on expressions of invariant genes.
  – Normalize based on the fitted curve.
dChip normalization

Figure 10: Similar plots as in Figure 9 for arrays hybridized to two different samples (array 24 and 36 of array set 5).

(a) CEL intensities; (b) same plot as in (a) with superimposed circles representing the invariant set; (c) after renormalization; (d) Q-Q plot of normalized probe intensities. Note that the smoothing spline in (a) is affected by several points at the lower-right corner, which might belong to differentially expressed genes. The invariant set, on the other hand, does not include these points when determining the normalization curve, leading to a different normalization relationship at the high end.
Quantile normalization

Proposed in Bolstad et al. 2003, Bioinformatics:

• Force the distribution of all data from all arrays to be the same, but keep the ranks of the genes.

• Procedures:
  1. Create a target distribution, usually use the average from all arrays.
  2. For each array, match its quantiles to that of the target. To be specific: \( x_{\text{norm}} = F_2^{-1}(F_1(x)) \):
     • \( x \): value in the chip to be normalized
     • \( F_1 \): distribution function in the array to be normalized
     • \( F_2 \): target distribution function
A simple example for quantile normalization

<table>
<thead>
<tr>
<th>Gene</th>
<th>sample1</th>
<th>Sample2</th>
<th>Sample3</th>
<th>Sample4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>15</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>
1. Find the Smallest Value for each sample

<table>
<thead>
<tr>
<th>Gene</th>
<th>sample1</th>
<th>Sample2</th>
<th>Sample3</th>
<th>Sample4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>15</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>

2. Average them

\[
\frac{(1+2+2+8)}{4}=3.25
\]
3. Replace Each Value by the Average

<table>
<thead>
<tr>
<th>Gene</th>
<th>Sample1</th>
<th>Sample2</th>
<th>Sample3</th>
<th>Sample4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>15</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>3.25</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>3.25</td>
</tr>
<tr>
<td>4</td>
<td>3.25</td>
<td>5</td>
<td>3.25</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>
4. Find the Next Smallest Values, then average

<table>
<thead>
<tr>
<th>Gene</th>
<th>sample1</th>
<th>Sample2</th>
<th>Sample3</th>
<th>Sample4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>15</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>3.25</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>3.25</td>
</tr>
<tr>
<td>4</td>
<td>3.25</td>
<td>5</td>
<td>3.25</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>

\[(3+5+5+9)/4=5.5\]
5. Replace Each Value by the Average

<table>
<thead>
<tr>
<th>Gene</th>
<th>sample1</th>
<th>sample2</th>
<th>sample3</th>
<th>sample4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>15</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>3.25</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>5.50</td>
<td>6</td>
<td>5.50</td>
<td>3.25</td>
</tr>
<tr>
<td>4</td>
<td>3.25</td>
<td>5.50</td>
<td>3.25</td>
<td>5.50</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>
6. Continue the process, we get the following matrix after finishing:

<table>
<thead>
<tr>
<th>Gene</th>
<th>sample1</th>
<th>sample2</th>
<th>sample3</th>
<th>sample4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.25</td>
<td>12.00</td>
<td>12.00</td>
<td>10.25</td>
</tr>
<tr>
<td>2</td>
<td>7.50</td>
<td>3.25</td>
<td>10.25</td>
<td>12.00</td>
</tr>
<tr>
<td>3</td>
<td>5.50</td>
<td>7.50</td>
<td>5.50</td>
<td>3.25</td>
</tr>
<tr>
<td>4</td>
<td>3.25</td>
<td>5.50</td>
<td>3.25</td>
<td>5.50</td>
</tr>
<tr>
<td>5</td>
<td>12.00</td>
<td>10.25</td>
<td>7.50</td>
<td>7.50</td>
</tr>
</tbody>
</table>

The result matrix has following properties:
- The values taken in each column are exactly the same.
- The ranks of genes in each column are the same as before normalization.
Before/after QN boxplot
Summary: between-array normalization

- Must do before comparing different arrays.
- Same problems exist in sequencing data.
- Quantile normalization is very strong and could remove the true signals, use with caution.
Microarray data summarization

• There are multiple probes targeting a gene. The task is to summarize the readings from these probes into one number to represent the gene expression.
• Naïve methods: mean, median.
• From MAS 5.0: use one-step Tukey Biweight (TBW) to obtain a robust weighted mean that is resistant to outliers.
  – Probes with intensities far away from median will have smaller weights in the average.
• dChip (Li & Wong, 2001): model based on $PM-MM$. 

RMA summarization

\[ Y_{ijn} = \mu_{in} + \alpha_{jn} + \epsilon_{ijn}, \quad i = 1, \ldots, I, j = 1, \ldots, J, n = 1, \ldots, n \]

log transformed PM intensities, denoted with \( Y \)
\( \mu_i \) representing the log scale expression level for array \( i \).
\( \alpha_j \) a probe affinity effect,
each probe set \( n \)

- Borrow information from multiple samples to estimate probe effects.
- **Model-fitting: Median Polish** (robust against outliers)
  - Iteratively removing the row and column medians until convergence
  - The remainder is the residual;
  - After subtracting the residual, the row medians are the estimates of the expression, and column medians are probe effects.

Irizarry et al. (2003) *Biostatistics*.
Bioconductor for microarray data

- There is a rich collection of bioc packages (hundreds) for microarrays. In fact, Bioconductor started for microarray analysis.

- Important ones include:
  - **affy**: one of the earliest bioc packages. Designed for analyzing data from Affymetrix arrays.
  - **oligo**: preprocessing tools for many types of oligonucleotide arrays. This is designed to replace affy package.
  - **limma** and **siggenes**: DE detection using limma and SAM-t model.
  - Many annotation data package to link probe names to genes.

- Data normalization and summarization can be done using oligo package (details next lecture).
Review

• We have covered microarray analysis, including:
  – Data preprocessing: within and between array normalization.
  – Summarization.

• Next lecture:
  – DE detection for microarray.