Microarray Data Analysis (V)

Preprocessing (i): two-color spotted arrays
Preprocessing

- **Probe-level data**: the intensities read for each of the components.
- **Genomic-level data**: the measures being used in real research.
Preprocessing Data

- Preprocessing: A procedure that extracts meaningful data characteristics and eliminates unrelated system biases.
  - Image analysis
  - Data import
  - Normalization
  - Summarization
  - Quality assessment
Preprocessing Two-Color Spotted Arrays
Image Analysis

• The raw data from a microarray experiment consist of pairs of image files, 16-bit TIFFs, one for each of the dyes.

• Image analysis is required to extract measures of the red and green fluorescence intensities for each spot on the array.
Steps in Image Analysis

1. **Addressing.** Estimate location of spot centers.

2. **Segmentation.** Classify pixels as foreground (signal) or background.

3. **Information extraction.** For each spot on the array and each dye
   - signal intensities;
   - background intensities;
   - quality measures.
• Bioconductor does NOT provide image analysis utilities and relies on other software.
• The marray package helps to import the resulting data from different software:
  – GenePix: .gpr
  – Spot: .spot
  – SMD: .xls
  – Agilent: .txt
Data Import

- **Illustrative data:** *Swirl Zebrafish*
  - *Swirl* is a point mutant in the BMP2 gene of Zebrafish that affects the dorsal/ventral body axis.
  - **Objective:** to identify genes differentially expressed in the *Swirl mutant* compared to wild-type zebrafish.

- **Microarray layout:**
  - 8448 probes.
  - 4 x 4 print-tips; each grid consisted of a 22 x 24 spots matrix.
22 x 24 x 4 x 4 = 8448
Data Import

• Information needed for effective statistical analysis:
  1) the sample target information
  2) the probe information
  3) the probe spot and background intensities
1) the sample target information describes which RNA samples were hybridized to each array; typically a tab-delimited text file (.txt)

```
<table>
<thead>
<tr>
<th></th>
<th>Names</th>
<th>slide number</th>
<th>experiment Cy3</th>
<th>experiment Cy5</th>
<th>date</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>swirl.1.spot</td>
<td>81</td>
<td>swirl</td>
<td>wild type</td>
<td>2001/9/20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>swirl.2.spot</td>
<td>82</td>
<td>wild type</td>
<td>swirl</td>
<td>2001/9/20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>swirl.3.spot</td>
<td>93</td>
<td>swirl</td>
<td>wild type</td>
<td>2001/11/8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>swirl.4.spot</td>
<td>94</td>
<td>wild type</td>
<td>swirl</td>
<td>2001/11/8</td>
<td></td>
</tr>
</tbody>
</table>
```

C:\Program Files\R\R-2.5.1\library\marray\swirldata\SwirlSample.txt
> read.marrayInfo(fname)

> swirl.samples = read.marrayInfo(file.path(datadir, "SwirlSample.txt"))
> swirl.samples
An object of class "marrayInfo"
@maLabels
[1] "swirl.1.spot" "swirl.2.spot" "swirl.3.spot" "swirl.4.spot"

@maInfo

<table>
<thead>
<tr>
<th></th>
<th>Names</th>
<th>slide number</th>
<th>experiment Cy3</th>
<th>experiment Cy5</th>
<th>date</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>swirl.1.spot</td>
<td>81</td>
<td>swirl</td>
<td>wild type</td>
<td>2001/9/20</td>
<td>NA</td>
</tr>
<tr>
<td>2</td>
<td>swirl.2.spot</td>
<td>82</td>
<td>wild type</td>
<td>swirl</td>
<td>2001/9/20</td>
<td>NA</td>
</tr>
<tr>
<td>3</td>
<td>swirl.3.spot</td>
<td>93</td>
<td>swirl</td>
<td>wild type</td>
<td>2001/11/8</td>
<td>NA</td>
</tr>
<tr>
<td>4</td>
<td>swirl.4.spot</td>
<td>94</td>
<td>wild type</td>
<td>swirl</td>
<td>2001/11/8</td>
<td>NA</td>
</tr>
</tbody>
</table>

@maNotes
2) the probe information describes the spotted probe sequences (gene names, annotations, etc.); if using GenePix or Spot, it is a tab-delimited text file (.gal) with rows corresponding to spotted probe sequences and columns containing various coordinates and annotations.

```
# fish.gal - 記事本
<table>
<thead>
<tr>
<th>Block</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

C:\Program Files\R\R-2.5.1\library\marray\swirldata\fish.gal
> read.Galfile(fname)

> # probe information
> galinfo = read.Galfile("fish.gal", path=datadir)
> galinfo

$gnames
An object of class "marrayInfo"
@maLabels
[1] "control" "control" "control" "control" "control"
8443 more elements ...

@maInfo

ID  Name
control control geno1
control control geno2
control control geno3
control control 3XSSC
control control 3XSSC
8443 more rows ...

@maNotes
[1] ""

$layout
An object of class "marrayLayout"
@maNgr
[1] 4
3) **the probe spot and background intensities:** microarray image processing results

```r
> read.Spot(fnames)    # Spot
> read.GenePix(fnames) # GenePix
> read.Agilent(fnames) # Agilent
```

The data is stored as *marrayRaw* class

```r
> data = read.Spot(path=datadir, targets=swirl.targets)
Reading ...  C:/PROGRA~1/R/R-25~1.1/library/marray/swirldat/a/swirl1.1.spot
Reading ...  C:/PROGRA~1/R/R-25~1.1/library/marray/swirldat/a/swirl1.2.spot
Reading ...  C:/PROGRA~1/R/R-25~1.1/library/marray/swirldat/a/swirl1.3.spot
Reading ...  C:/PROGRA~1/R/R-25~1.1/library/marray/swirldat/a/swirl1.4.spot
> slotNames(data)
[1] "maRf"     "maGf"     "maRb"     "maGb"     "maW"      "maLayout"
[7] "maGnames" "maTargets" "maNotes"
```
• For \textit{marrayRaw} and \textit{marrayNorm} objects, various methods are defined to extract stored information

\textbf{maM:} \quad M = \log_2 \frac{\text{Cy5}}{\text{Cy3}} = \log_2 (\text{Cy5}) - \log_2 (\text{Cy3})

\textbf{maA:} \quad A = \log_2 \sqrt{\text{Cy5} \cdot \text{Cy3}} = \frac{\log_2 (\text{Cy5}) + \log_2 (\text{Cy3})}{2}

\textbf{maLG:} \quad \text{the green log intensities}

\textbf{maLR:} \quad \text{the red log intensities}

\textbf{maGeneTable:} \quad \text{creates a \textit{data.frame} of spot coordinates and gene names}
> plot(maA(data), maM(data))

> maGeneTable(data)[1:4, 1:5]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>control</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1 control</td>
</tr>
<tr>
<td>control.1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2 control</td>
</tr>
<tr>
<td>control.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3 control</td>
</tr>
<tr>
<td>control.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4 control</td>
</tr>
</tbody>
</table>
Normalization

• Identify and remove **systematic sources of variation** in the measured fluorescence intensities, other than differential expression, for example
  – different labeling efficiencies of the dyes;
  – different amounts of Cy3- and Cy5- labeled mRNA;
  – different scanning parameters;
  – sector/print-tip, spatial, or plate effects, etc.

• Caution – **Bias-variance trade-off**: more complex normalization procedures tend to be able to remove more of the technical variation but they might also remove more of the biological signals.
Normalization

- The need for normalization can be seen most clearly in **self-self hybridizations** where the same mRNA sample is labeled with the Cy3 and Cy5 dyes.

- The imbalance in the red and green intensities is usually **not constant** across the spots within and between arrays, and can vary according to overall spot intensity, location, plate origin, etc.

- These factors should be considered in the normalization.
Self-Self Hybridization for Two-Channel Arrays
(Ideal case)

\[ A = \frac{\log Cy5 + \log Cy3}{2} \]

\[ M = \log Cy5 - \log Cy3 = \text{log-ratios} \]
Self-Self Hybridization for Two-Channel Arrays

Within-Array Normalization

- Dealing with absolute log intensities; to eliminate spot-to-spot variation
  - Location normalization:
    \[ M_{\text{norm}} = M - l \]
  - Scale normalization:
    \[ M_{\text{norm}} = M / s \]
  - Location-scale normalization:
    \[ M_{\text{norm}} = (M - l) / s \]
Location Normalization

- $M_{\text{norm}} = M - l$
  - It centers log-ratios around 0.
  - $l$ can be identical for all values of $M$:
    - Global median normalization: $l = \text{median}(M)$
  - $l$ can vary:
    - Global Lowess/Loess: $l = \text{the expected value after locally fitting a smooth regression curve on the global MA plot}$.
    - Print-tip Lowess/Loess: $l(i) = \text{the loess/lowess fit to the MA-plot for the } i\text{ th grid only.}$
LOWESS/LOESS

- The function fitted by LOWESS is a polynomial of the form:

\[ y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots \]

- The approach used by LOWESS/LOESS:

1. The degrees of the polynomials used are limited to 1 (LOWESS) or 2 (LOESS) to avoid over-fitting.

2. The data points that fall into this intervals will be used to fit the first polynomial in a weighted manner.
Loess Normalization Result

Global Loess
Scale Normalization

• Scale normalization can be applied if suspecting unequal variances.

• However, it is recommended to check the need of such normalization; there is a trade-off between the possible gain in bias achieved by scale normalization and the increase in variability introduced by this additional step.

• If the scale differences are relatively small, it is preferable to perform only a location normalization.
R: Within-Array Normalization

maNorm (limma): returns a marrayNorm object

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Description</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No normalization.</td>
<td>n</td>
</tr>
<tr>
<td>Median</td>
<td>Global median location normalization.</td>
<td>m</td>
</tr>
<tr>
<td>Loess</td>
<td>Global A-dependent normalization using the scatter-plot smoother loess.</td>
<td>l</td>
</tr>
<tr>
<td>Print-tip loess</td>
<td>A-dependent normalization using the scatter-plot smoother loess within print-tip groups.</td>
<td>p</td>
</tr>
<tr>
<td>2D loess</td>
<td>2D–spatial normalization using the loess function.</td>
<td>twoD</td>
</tr>
</tbody>
</table>

maNormScale (limma)
Between-Array Normalization

• Dealing with log-ratios; to make array replicates comparable.
  – Quantile normalization method
  – VSN-method
Quantile Normalization

• **Assumption:** The measurements from different arrays share the same underlying distribution.

• It forces arrays have an identical distribution:

\[ x_i' = F^{-1}(G_i(x_i)) \]

\( G_i \) = empirical distribution functions for the \( i \) th array
\( F \) = the empirical distribution function of the means of the quantiles over all arrays
\[
\begin{align*}
\text{mean} &= (-0.862 - 1.461 - 0.951)/3 \\
\end{align*}
\]
Quantile Normalization

• Variants:
  – Gquantile: ensures that the green (first) channel has the same empirical distribution across arrays
  – Rquantile: ensures that the red (second) channel has the same empirical distribution across arrays
  – Aquantile: ensures that the A-values (average intensities) have the same empirical distribution across arrays
  – Tquantile: performs quantile normalization separately for the groups indicated by 'targets'
Variance Stabilization and Normalization (VSN) Method

- VSN assumes that less than half of the genes on the arrays is differentially transcribed across the experiment.
- The variance of microarray data may depend on the signal intensity; after normalization the variance is approximately constant.
- $x_{ki} =$ expression level of the $k$th probe in the $i$th array.

$$x'_{ki} = g \log \left( \frac{x_{ki} - a_i}{b_i} \right)$$

$a_i =$ background offset
$b_i =$ scale parameter for array $i$
glog(.) = genealized logarithm,
R: Within-Array Normalization

`normalizeBetweenArrays` (limma)

**Usage:**

```r
normalizeBetweenArrays(object, method="Aquantile", targets=NULL, ...)
```

**Arguments:**

- `object`: a `matrix`, `RGList` or `MAList` object containing expression ratios for a series of arrays

- `method`: character string specifying the normalization method to be used. Choices are "none", "scale", "quantile", "Aquantile", "Gquantile", "Rquantile", "Tquantile" or "vsn". A partial string sufficient to uniquely identify the choice is permitted.
Data Output

• Export the normalized data into a file:

  > write.marray(withinNormData)
Quality Assessment

• Before and after normalization, it is important to consider and unsure the quality of the data. ⇒ Visualization tools.

• The package arrayQuality gives user a quick visual way to access the quality of individual arrays by providing per-slide diagnostic plots.
> maQualityPlots(data, dev="jpeg")

dev: A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".