Pre-processing

Martin Morgan (mtmorgan@fhcrc.org)
Fred Hutchinson Cancer Research Center

28 January 2010

Questions & platforms

- Expression
  - Single channel (e.g., Affymetrix) – *affy*; *xps*, *aroma.affymetrix*
  - Two channel (e.g., Agilent / Genepix) – *limma*
  - Bead array (e.g., Illumina) – *lumi*, *beadarray*
  - Long oligo (Nimblegen) – *oligo*
- Array CGH
- Exon
- Methylation
- Genotyping, e.g., SNP

Full analysis possibilities: `http://bioconductor.org/packages/release/Software.html`

Work flow – expression arrays

Prior to analysis

- Biological experimental design
- Expression experimental design – especially two-channel

Analysis

1. Pre-processing (normalization); quality assessment; exploratory analysis
2. Differential expression; machine learning (clustering and classification)
3. Annotation
4. Gene set enrichment analysis
5. ...

Pre-processing

Background correction

- One-channel: PM / MM probes
- Two-channel: background vs. foreground intentisities

Normalization

- Key assumption: most probes not differentially expressed; distribution of intensities approxiamtely equal across arrays

Summarization

- One-channel: from probes to probesets (approxiamtely, genes)

One channel Affymetrix 3' expression arrays

- In practice:

```
> ## assume phenoData is an AnnotatedDataFrame
> ## "/celfile/directory" contains CEL files
> setwd("/celfile/directory")
> library(affy)
> eset <- just.rma(phenoData=phenoData)
```

- Also: `just.gcrma`
- `expresso` for more flexible control; *affyPLM* for detailed probe models; *oligo* for recent arrays.
- `http://bioconductor.org/workflows` for common analyses.

Two channel expression arrays

- In practice, e.g., Genepix gpr files:

```
> ## create 'targets' from file names, phenotype data
> gpr <- list.files("/gpr/directory", "\\.gpr$",
+                   full=TRUE)
> targets <- data.frame(FileName=gprFiles)
> library(limma)
> rg <- read.maimages(targets, source="genepix")
> ma <- normalizeWithinArrays(rg)
```

- Considerable flexiblity in data input, background correction, within-array normalization.
- Default: 'subtract' background, 'printtiploess' normalization.
- Result: an `MAList`

Example: RMA (robust multi-chip average)

Background correction

- Observation: using MM probes is problematic when MM>PM.
- Model PM probes as expoentially distributed signal, plus normal noise, exp(α) + $N(\mu, \sigma^2)$.

Normalization

- Quantile normalization – force the *distribution* of background-corrected expression values of each array to have exactly the same distribution.

Summarization

- Estimate probeset effect by fitting a linear model to all probes in each probe set, across array.

Quality assessment

- In practice:

```
> library(arrayQualityMetrics)
> rpt <- arrayQualityMetrics(abatch)
> ## or, as appropriate,
> ## rpt <- arrayQualityMetrics(eset)
> ## rpt <- arrayQualityMetrics(rg)
> browseURL(rpt)
```

- QC summary statistics: acceptable ranges for 'control' probes
- Between-array distances: no unintended association with experimental conditions, e.g., run date.
- NUSE (normalized unscaled standard error) and RLE (relative log expression) plots: consistent expression and variablity across arrays.

Lab activity

- Chapter 3, sections 3.1 – 3.3.
- Goals: manipulating *AffyBatch* and *ExpressionSet* objects; become familiar with R packages, including obtaining help; understanding essentials of pre-processing and quality assessment.