Microarray Analysis

The Basics

Thomas Girke

December 7, 2012

Technology

Challenges

Data Analysis

Data Depositories

Microarray Analysis with R and Bioconductor

Outline

Microarray and Chip Technology

Definition

- Hybridization-based technique that allows simultaneous analysis of thousands of samples on a solid substrate.

Applications

- Transcriptional Profiling
- Gene copy number
- Resequencing
- Genotyping
- Single-nucleotide polymorphism
- DNA-protein interaction (e.g.: ChIP-on-chip)
- Gene discovery (e.g.: Tiling arrays)
- Identification of new cell lines
- Etc.

Related technologies

- Protein arrays
- Compound arrays

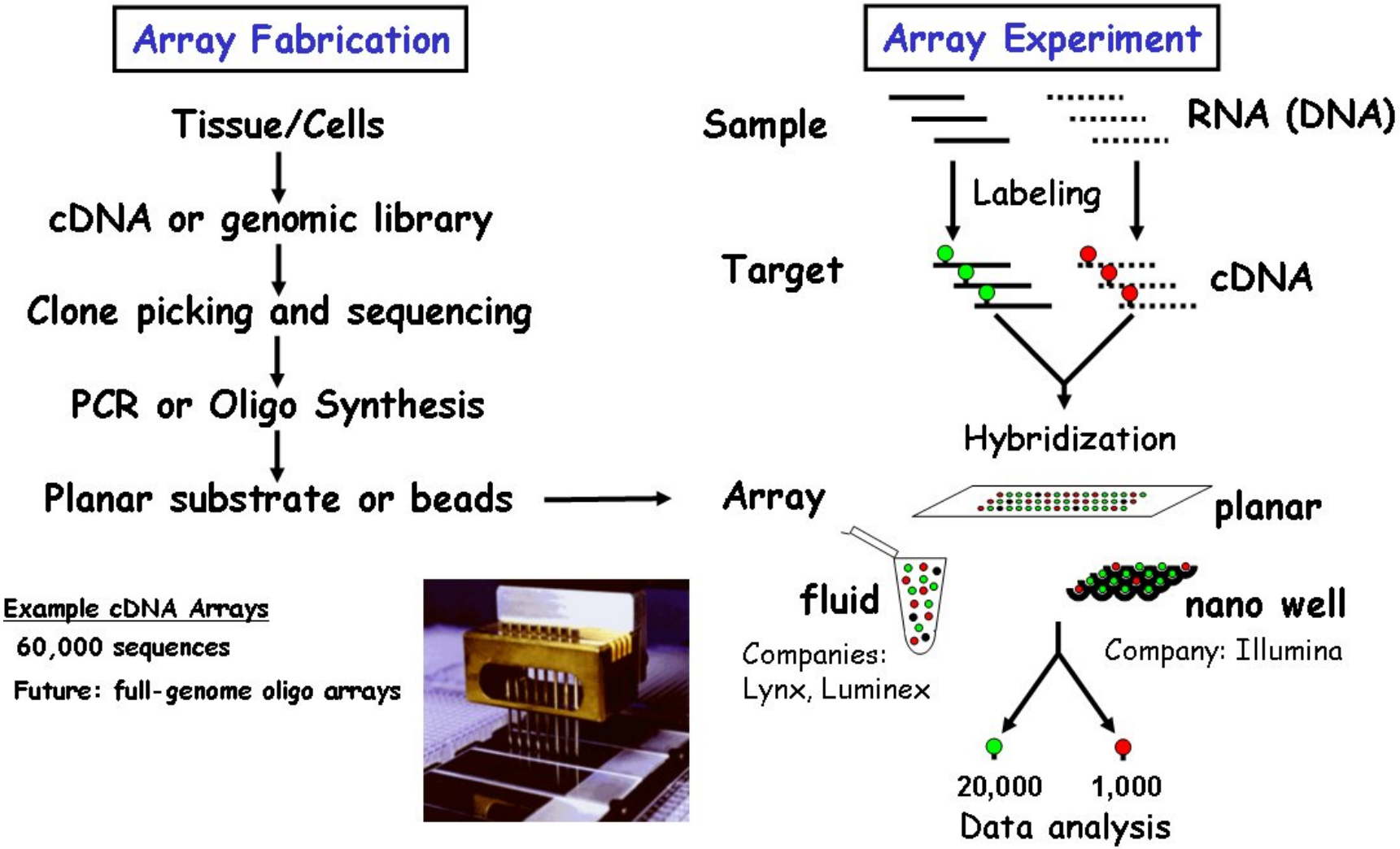
Why Microarrays?

- Simultaneous analysis of thousands of genes
- Discovery of gene functions
- Genome-wide network analysis
- Analysis of mutants and transgenics
- Identification of drug targets
- Causal understanding of diseases
- Clinical studies and field trials

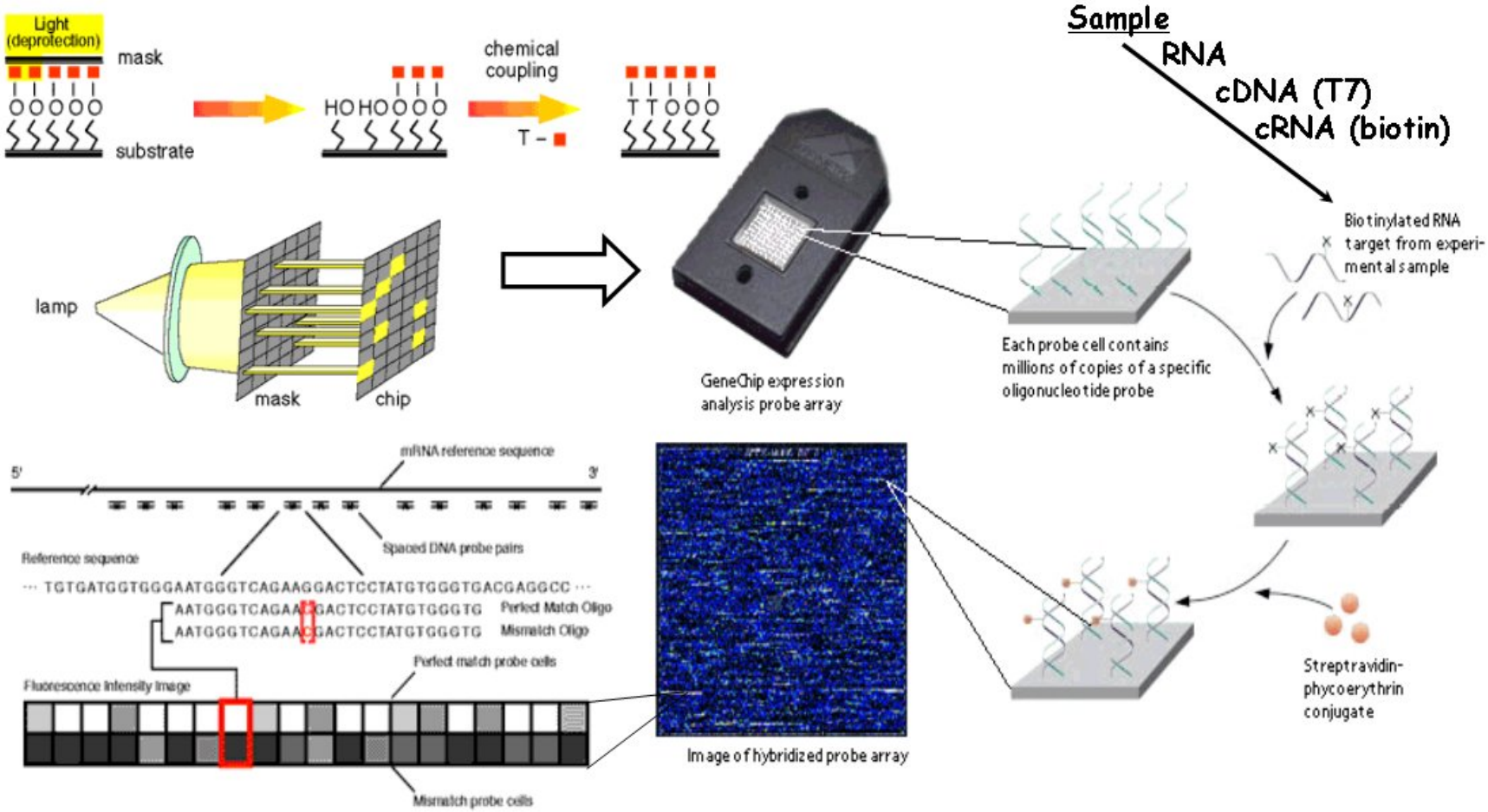
Different Types of Microarrays

- Single channel approaches
  - Affymetrix gene chips
  - Macroarrays
- Multiple channel approaches
  - Dual color (cDNA) microarrays
- Specialty approaches
  - Bead arrays: Lynx, Illumina, ...
  - PCR-based profiling: CuraGen, ...

Dual Color Microarrays

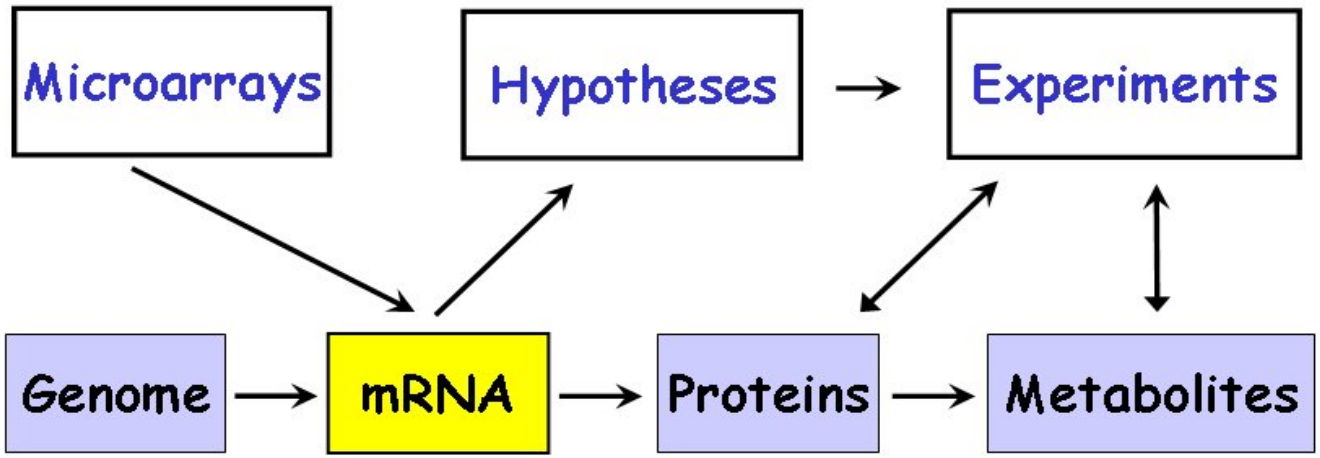


Affymetrix DNA Chips



Outline

Profiling Chips Monitor Differences of mRNA Levels



Efficient strategy for down-stream follow-up experiments important!

Strategies to Validate Array Hits

- Real-time PCR, Northern, etc.
- Transgenic tests
- Knockout plants and/or activation tagged lines
- Protein profiling
- Metabolic profiling
- Other tests: in situ hybs, biochemical and physiological tests
- Integration with sequence, proteomics and metabolic databases

Sources of Variation in Transcriptional Profiling Experiments

- Every step in transcriptional profiling experiments can contribute to the inherent 'noise' of array data.
- Variations in biosamples, RNA quality and target labeling are normally the biggest noise introducing steps in array experiments.
- Careful experimental design and initial calibration experiments can minimize those challenges.

Experimental Design

- Biological questions:
  - Which genes are expressed in a sample?
  - Which genes are differentially expressed (DE) in a treatment, mutant, etc.?
  - Which genes are co-regulated in a series of treatments?
- Selection of best biological samples and reference
  - Comparisons with minimum number of variables
  - Sample selection: maximum number of expressed genes
  - Alternative reference: pooled RNA of all time points (saves chips)
- Develop validation and follow-up strategy for expected expression hits
  - e.g. real-time PCR and analysis of transgenics or mutants
- Choose type of experiment
  - common reference, e.g.: S1 x S1+T1, S1 x S1+T2
  - paired references, e.g.: S1 x S1+T1, S2 x S2+T1
  - loop & pooling designs
  - many other designs
- At least three (two) biological replicates are essential
  - Biological replicates: utilize independently collected biosamples
  - Technical replicates: utilize often the same biosample or RNA pool

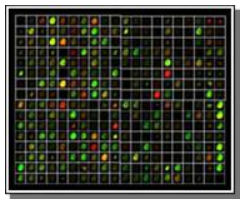
Outline

Basic Data Analysis Steps

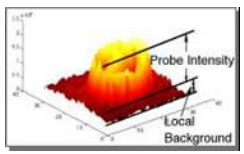
- Image Processing: transform feature and background pixel into intensity values
- Transformations
  - Removal of flagged values (optional)
  - Detection limit (optional)
  - Background subtraction
  - Taking logarithms
- Normalization
- Identify EGs and DEGs
  - Which genes are expressed?
  - Which genes are differentially expressed?
- Cluster analysis (time series)
  - Which genes have similar expression profiles?
- Promoter analysis
- Integration with functional information: pathways, etc.

Image Analysis

- Overall slide quality
- Grid alignment (linkage between spots and feature IDs)



- Signal quantification: mean, median, threshold, etc.



- Local background
- Manual spot flagging
- Export to text file

Image analysis software (selection)

ScanAlyze (http://rana.lbl.gov/EisenSoftware.htm)

TIGR SpotFinder (http://www.tigr.org/software/)

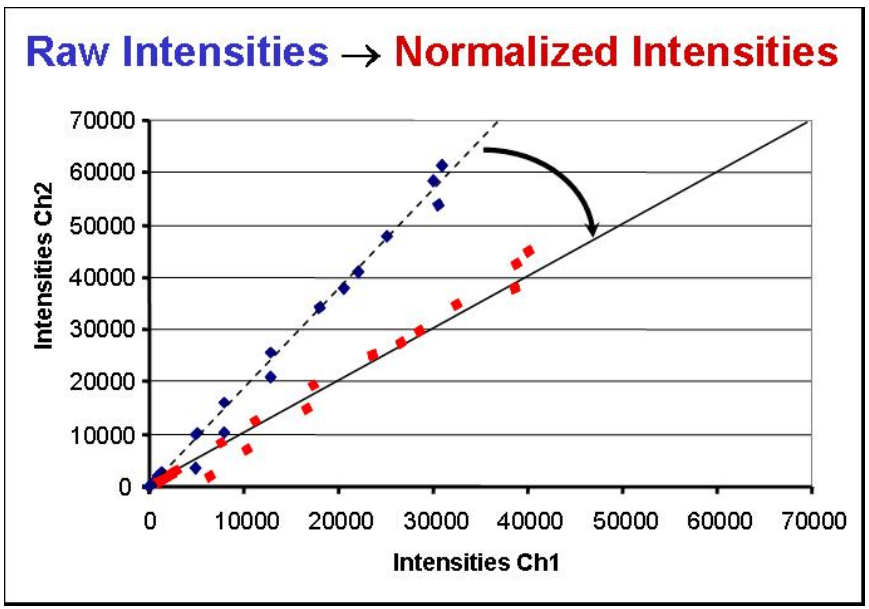
Background Correction

- Filtering (optional)
  - Intensities below detection limit
  - Negative intensities
  - Spacial quality issues
- Background correction
  - BG consists of non-specific hybridization and background fluorescence
  - If BG is higher than signal: (1) remove values, (2) set signal to lowest measured intensity, (3) many other approaches
  - BG subtraction
    - Local background
    - Global background
    - No background subtraction

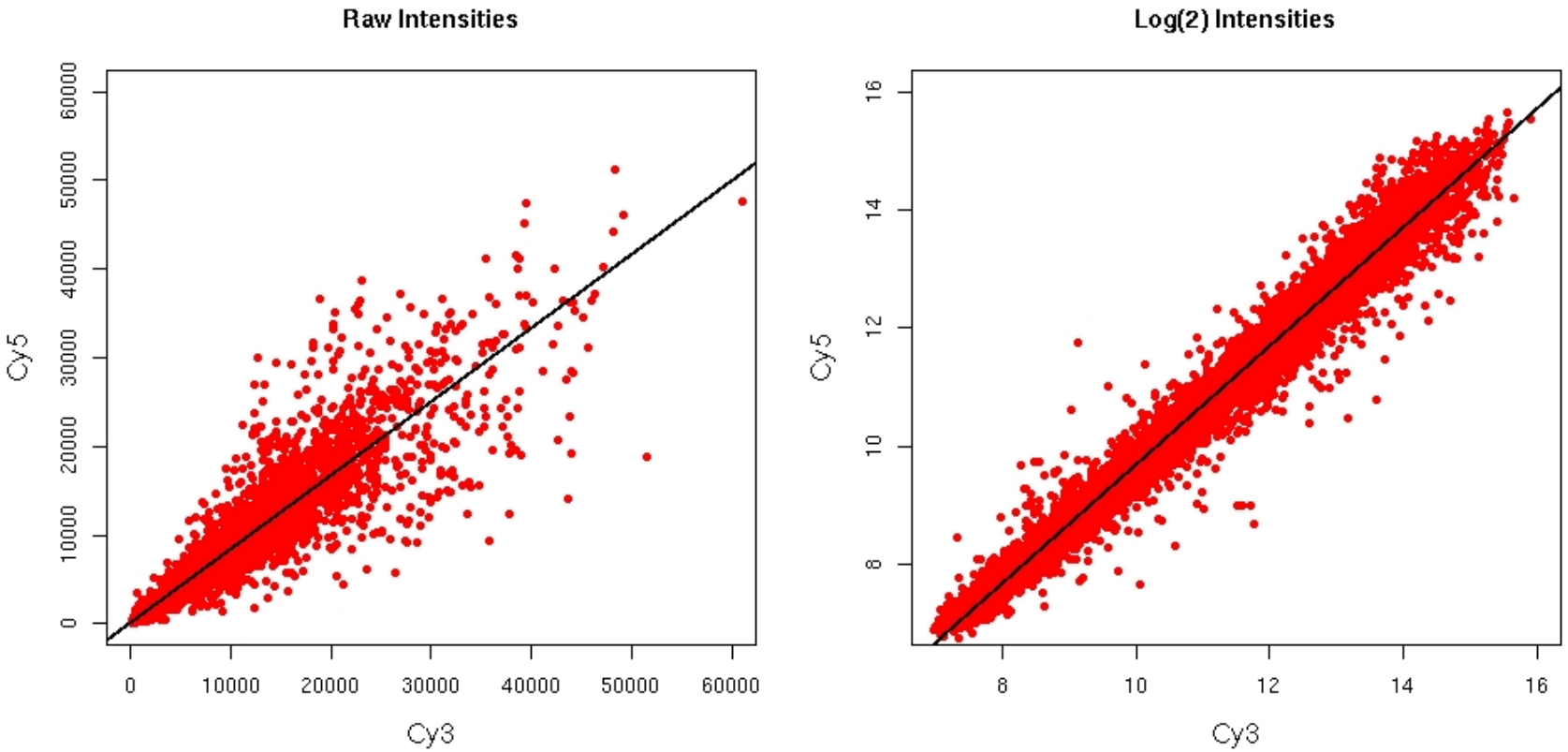
  Background subtraction can cause ratio inflation, therefore background corrected intensities below threshold are often set to threshold or similar value.

Normalization

Normalization is the process of balancing the intensities of the channels to account for variations in labeling and hybridization efficiencies. To achieve this, various adjustment strategies are used to force the distribution of all ratios to have a median (mean) of 1 or the log-ratios to have a median (mean) of 0.



Log Transformation: Scatter Plots



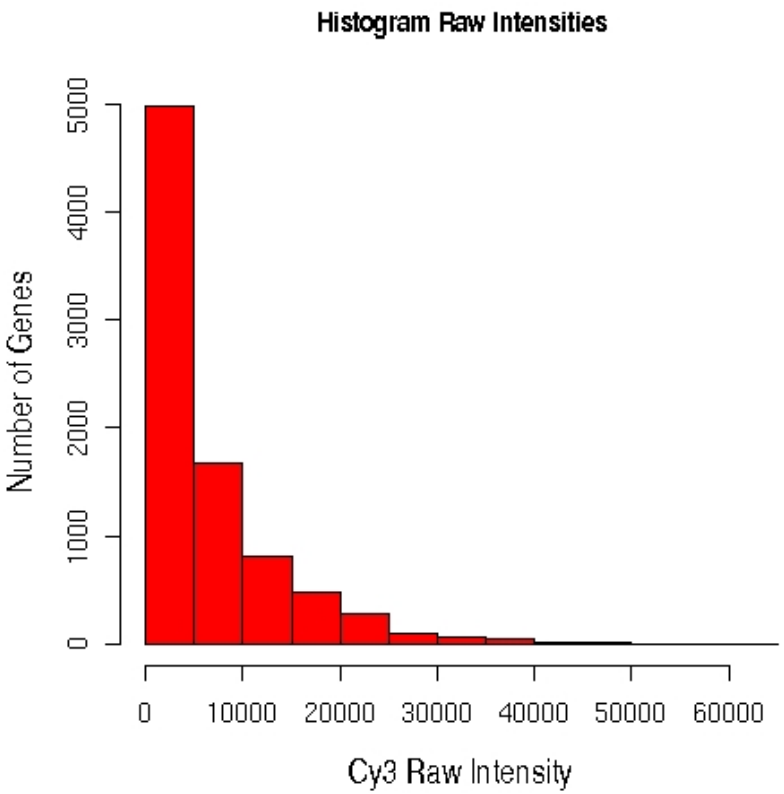
Reasons for working with log-transformed intensities and ratios

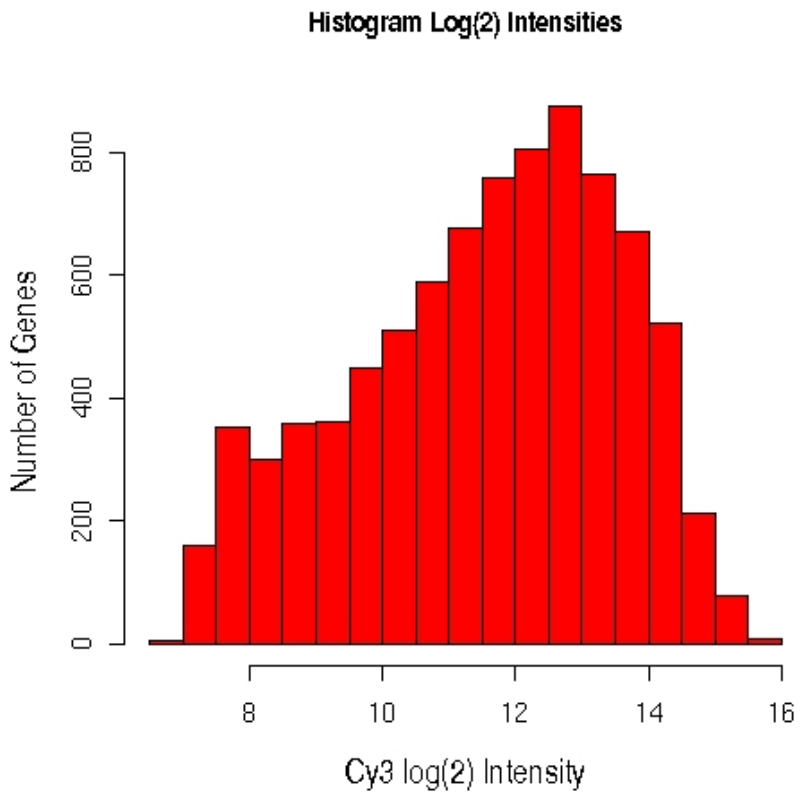
(1) spreads features more evenly across intensity range

(2) makes variability more constant across intensity range

(3) results in close to normal distribution of intensities and experimental errors

Log Transformation: Histograms





Distribution of log transformed data is closer to being bell-shaped

Normalization If Large Fraction of Genes IS DE

Minimize normalization requirements (dynamic range limits)

- Pre-scanning: hybridize equal amounts of label
- During scanning: balance average intensities through laser power and PMP adjustments

Normalization if large fraction of genes is DE

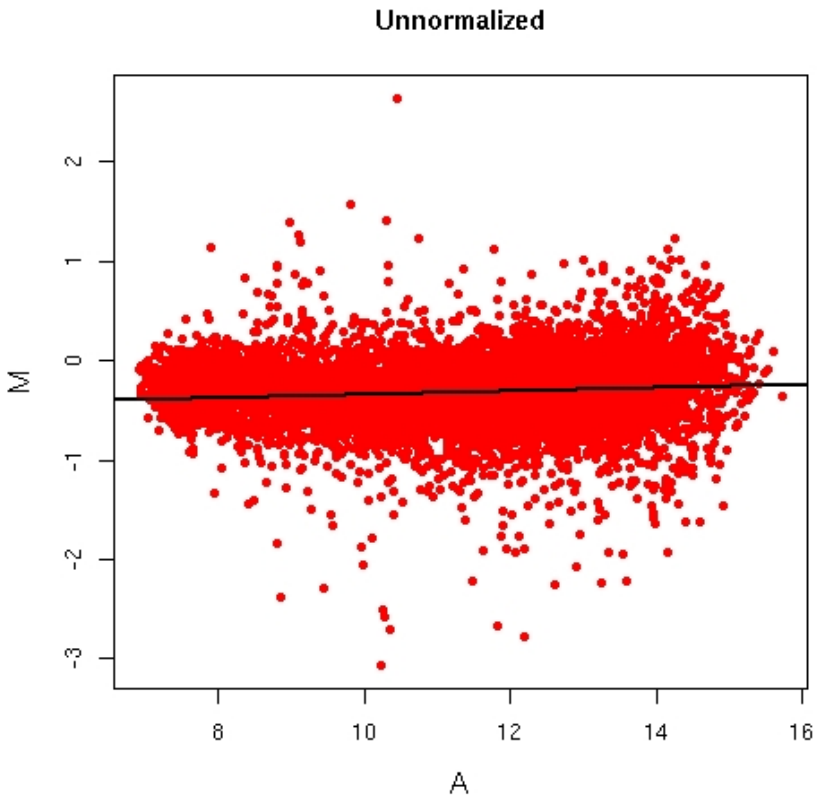
- Spike-in controls
- Housekeeping controls
- Determine constant feature set

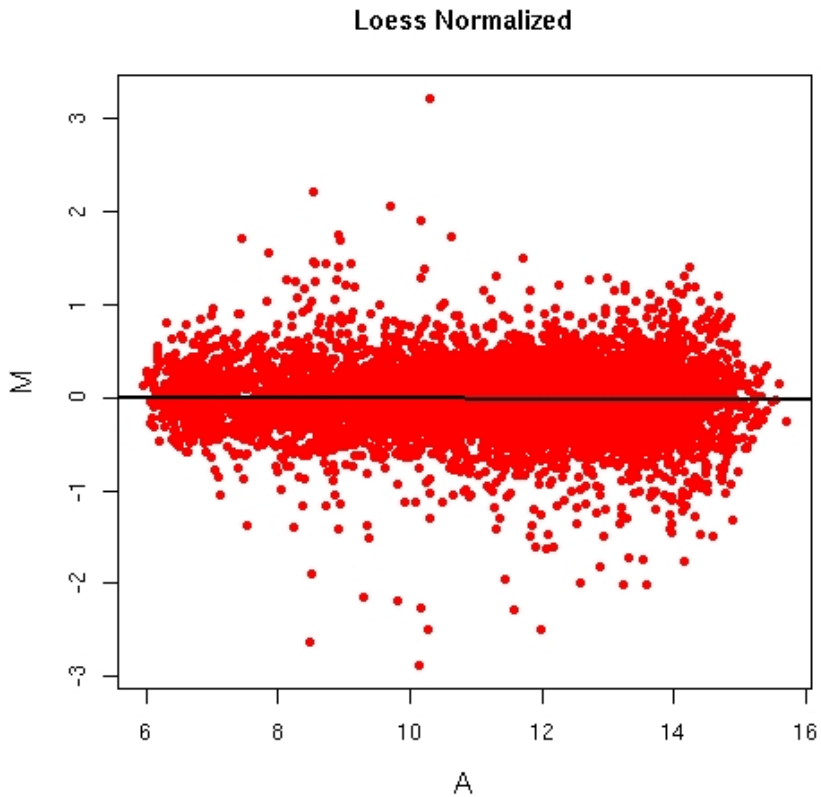
Normalization If Large Fraction of Genes IS NOT DE

Global Within-Array Normalization

- Multiply one channels with normalization factor
  $\Rightarrow$ Ch2 x mCh1/mCh2 (treats both channels differently)
- Linear regression fit of log2(Ch2) against log2(Ch1)
  $\Rightarrow$ adjust Ch1 with fitted values (treats both channels differently)
- Linear regression fit of log2(ratios) against avg log2(int)
  $\Rightarrow$ subtract fitted value from raw log ratios (treats both channels equally)
- Non-linear regression fit of log2(ratios) against avg log2(int)
  Most commonly used: Loess (locally weighted polynomial) regression joins local regressions with overlapping windows to smooth curve
  $\Rightarrow$ subtract fitted value on Loess regression from raw log ratios (treats both channels equally)

MA Plots


Unnormalized
M
A
-3
-2
-1
0
1
2
8
10
12
14
16


Loess Normalized
M
A
-3
-2
-1
0
1
2
3
6
8
10
12
14
16

Normalization If Large Fraction of Genes IS NOT DE

Spacial Within-Array Normalization

All of the above methods can be used to correct for spacial bias on the array. Examples:

- Block or Print Tip Loess
- 2D Loess Regression

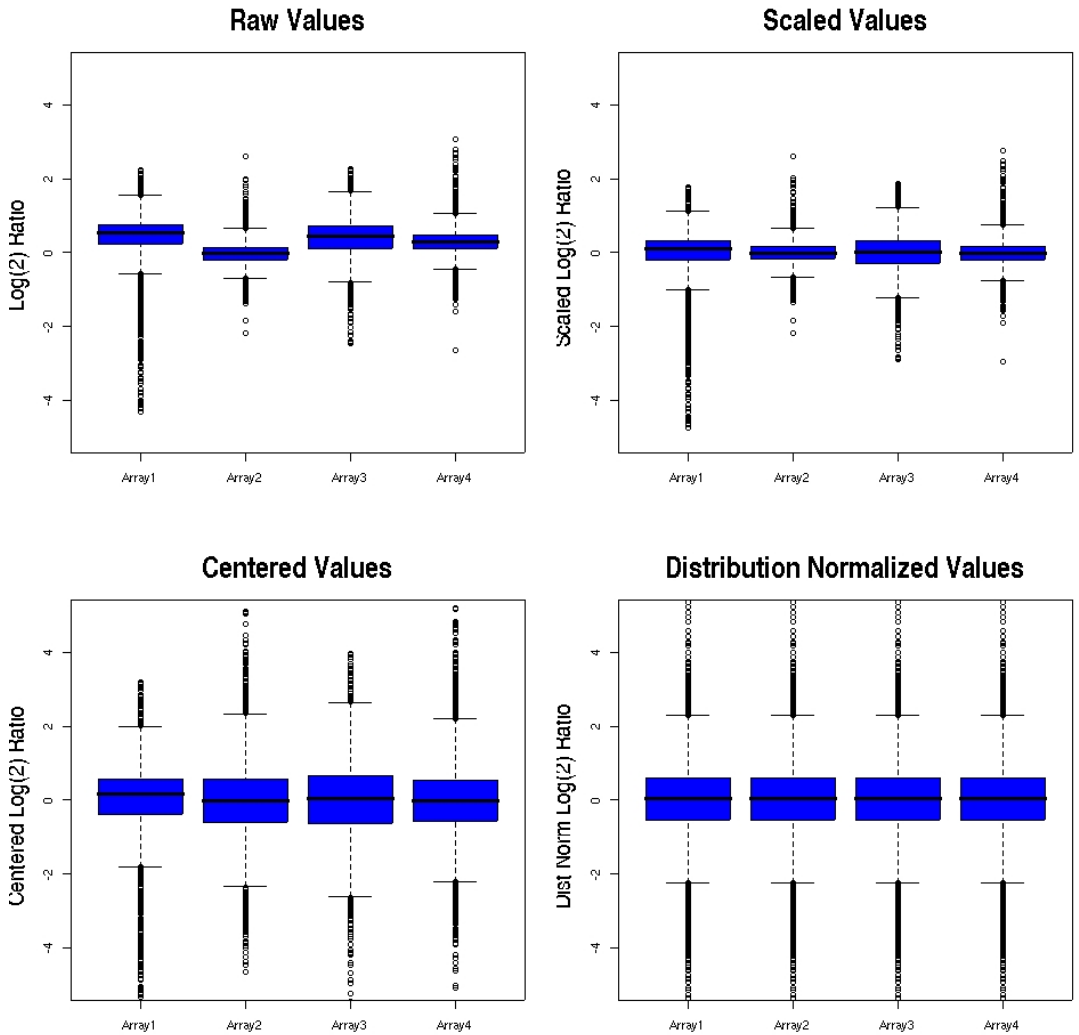
Normalization If Large Fraction of Genes IS NOT DE

Between-Array Normalization

To compare ratios between dual-color arrays or intensities between single-color arrays

- Scaling
  ⇒ log(rat) - mean log(rat) or log(int) - mean log(int)
  ⇒ Result: mean = 0
- Centering (z-value)
  ⇒ [rat - mean(rat)] / [STD] or [int - mean(int)] / [STD]
  ⇒ Result: mean = 0, STD = 1
- Distribution Normalization (apply to group of arrays!)
  ⇒ (1) Generate centered data, (2) sort each array by intensities, (3) calculate mean for sorted values across arrays, (4) replace sorted array intensities by corresponding mean values, (5) sort data back to original order
  ⇒ Result: mean = 0, STD = 1, identical distribution between arrays

Box Plots for Between-Array Normalization Steps



Analysis Methods for Affymetrix Gene Chips

Method	BG Adjust	Normalization	MM Correct	Probeset Summary
MAS5	regional adjustment	scaling by constant	subtract idealized MM	Tukey biweight average
gcRMA	by GC content	quantile normalization	/	robust fit of linear model
RMA	array background	quantile normalization	/	robust fit of linear model
VSN	/	variance stabilizing TF	/	robust fit of linear model
dChip	/	by invariant set	/	multiplicative model
dChip.mm	/	by invariant set	subtract mismatch	multiplicative model

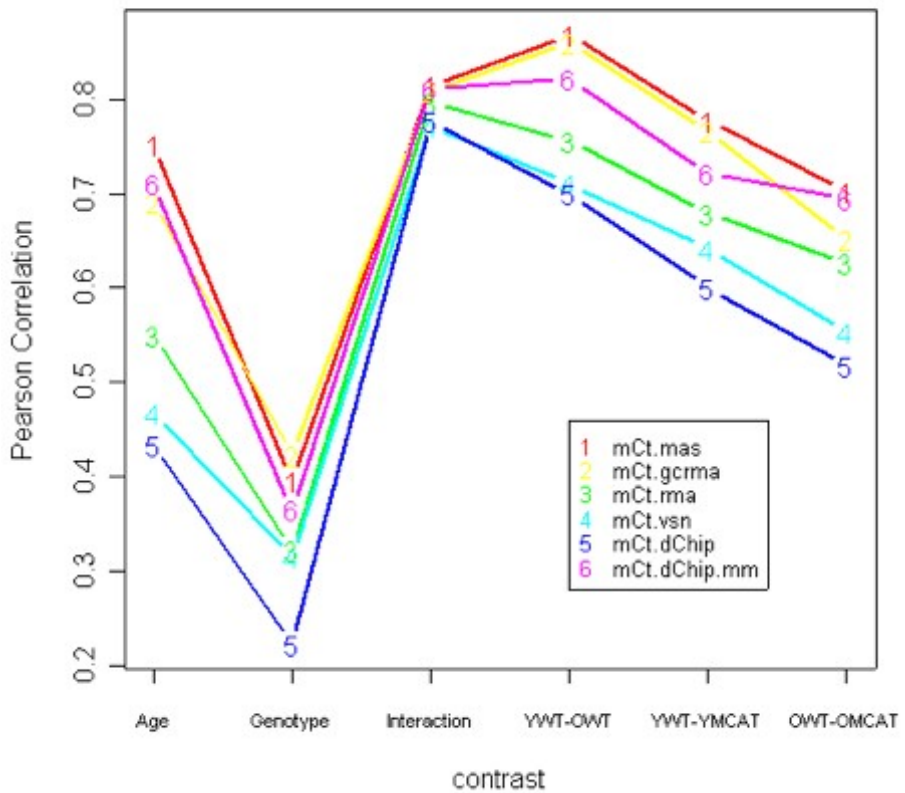
Qin et al. (2006), BMC Bioinfo, 7:23.

Reverences

MAS 5.0: Affymetrix Documentation: MAS5
PLIER: Affymetrix Documentation: PLIER, not included here
gcRMA: Wu et al. (2004), JASA, 99, 909-917.
RMA: Irizarry et al. (2003), Nuc Acids Res, 31, e15.
VSN: Huber et al. (2002), Bioinformatics, 18, Suppl I S96-104.
dChip & dChip.mm: Li & Wong (2001), PNAS, 98, 31-36.

Performance Comparison of Affy Methods

Qin et al. (2006), BMC Bioinfo, 7:23: 24 RNA samples hybridized to chips and 47 genes tested by qRT-PCR, plot shows PCC for 6 summary contrasts of 6 methods.



MAS5, gcRMA, and dChip (PM-MM) outperform the other methods. PLIER not included here.

Analysis of Differentially Expressed Genes

- Advantages of statistical test over fold change threshold for selecting DE genes
  - Incorporates variation between measurements
  - Estimate for error rate
  - Detection of minor changes
  - Ranking of DE genes
- Approaches
  - Parametric test: t-test
  - Non-parametric tests: Wilcoxon sign-rank/rank-sum tests
  - Bootstrap analysis (boot package)
  - Significance Analysis of Microarrays (SAM)
  - Linear Models of Microarrays (LIMMA)
  - Rank Product
  - ANOVA and MANOVA (R/maanova)
- Multiplicity of testing: p-value adjustments
  - Methods: fdr, bonferroni, etc.

Outline

Microarray Databases and Depositories

- NCBI GEO: http://www.ncbi.nlm.nih.gov/geo
- Microarray @ EBI: http://www.ebi.ac.uk/microarray
- SMD: http://genome-www5.stanford.edu
- Many Others

Outline

Why Using R and Bioconductor for Array Analysis?

- Most comprehensive environment for array analysis
- Large user and developer community
- Highly reproducible analysis
- Extensive visualization tools
- Large number of clustering and machine learning tools

Books & Documentation

- simpleR - Using R for Introductory Statistics (Gentleman et al., 2005)
- Bioinformatics and Computational Biology Solutions Using R and Bioconductor (John Verzani, 2004)
- UCR Manual (Thomas Girke)

Installation of Bioconductor Packages

R needs to be available on your system:

http://cran.at.r-project.org Link

RStudio is a nice interface to R

http://www.rstudio.com/ide/download Link

Installation of Bioconductor Packages

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite()
> biocLite(c("GOstats", "graph", "GO", "Category", "plier",
"affylmGUI", "limmaGUI", "simpleaffy", "ath1121501.db",
"ath1121501cdf", "ath1121501probe", "biomaRt",
"affycoretools"))
```

Data Download: Affymetrix Sample Set

- Right-click this link Link and save its content to your computer: Workshop.zip. These sample CEL files are from the GEO data set: GSE5621 Link
- After unpacking this file archive one should see six *.cel files in the Workshop directory.
- Change the directory of your R session to the Workshop directory containing the cel files.

Data Import

Environment settings and CEL file import

```
> ## Set working directory and check its content
> # getwd()
> dir(pattern="*.cel")
```

```
[1] "COLD_12H_SHOOT_REP1.cel"          "COLD_12H_SHOOT_REP2.cel"
[3] "COLD_6H_SHOOT_REP1.cel"           "COLD_6H_SHOOT_REP2.cel"
[5] "COLD_CONTROL_12H_SHOOT_REP1.cel" "COLD_CONTROL_12H_SHOOT_REP2.cel"
```

```
> ## Load required libraries
> library(affy); library(ath1121501probe)
> ## Print some probe sequences and their mapping positions for first Affy ID
> print.data.frame(ath1121501probe[1:3,1:4])
```

```
                  sequence   x   y Probe.Set.Name
1 TTGCTGCTATTCTATCTATTTGTGC 594 703      244901_at
2 GACTTTCAAAGTGACTCTCGACGGG 267 353      244901_at
3 GAGCCTCCAGGCTATTCAGGAAGAA 586 367      244901_at
```

```
> ## Import all cel files in current working directory into affybatch object
> mydata <- ReadAffy()
> # mydata # Prints summary of mydata content
```

Normalization

Normalization with RMA

```
> eset_rma <- rma(mydata) # Generates RMA expression values and stores them as Expre
Background correcting
Normalizing
Calculating Expression
> exprs(eset_rma)[1:4,1:2] # Prints first 4 rows in data frame structure.
          COLD_12H_SHOOT_REP1.cel COLD_12H_SHOOT_REP2.cel
244901_at                6.312229                6.095745
244902_at                5.665007                5.363950
244903_at                8.222791                7.624313
244904_at                7.194826                6.828998
> # exprs(eset_rma) <- 2^(exprs(eset_rma))[1:4,] # If needed unlog RMA expression va
> ## Generic approach for calculating mean values for any sample combination.
> mydf <- 2^exprs(eset_rma)
> myList <- tapply(colnames(mydf), c(1,1,2,2,3,3), list)
> names(myList) <- sapply(myList, paste, collapse="_")
> mymean <- sapply(myList, function(x) rowMeans(mydf[,x]))
> mymean[1:2,1:2]
          COLD_12H_SHOOT_REP1.cel_COLD_12H_SHOOT_REP2.cel
244901_at                                        73.92773
244902_at                                        45.96033
          COLD_6H_SHOOT_REP1.cel_COLD_6H_SHOOT_REP2.cel
244901_at                                      67.99992
244902_at                                      46.82990
```

Present, Marginal and Absent Calls

MAS5 to compute PMA values and Wilcoxon p-values

```
> ## Generate MAS 5.0 P/M/A calls and combine RMA intensities, P/M/A calls
> ## and Wilcoxon p-values in one data frame.
> eset_pma <- mas5calls(mydata)

Getting probe level data...
Computing p-values
Making P/M/A Calls

> my_frame <- data.frame(exprs(eset_rma), exprs(eset_pma),
+                        assayDataElement(eset_pma, "se.exprs"))
> my_frame <- my_frame[, sort(names(my_frame))] # Sort columns by cel file name
> my_frame[1:2,1:2]

          COLD_12H_SHOOT_REP1.cel COLD_12H_SHOOT_REP1.cel.1
244901_at                6.312229                        P
244902_at                5.665007                        P

> ## Export results to text file that can be imported into Excel
> write.table(my_frame, file="my_file.xls", sep="\t", col.names = NA)
```

Add Annotation Information

Constructs a data frame containing the gene IDs, gene symbols and descriptions for all probe sets on the chip.

```
> library("ath1121501.db") # Loads required annotation package.
> Annot <- data.frame(ACCNUM=sapply(contents(ath1121501ACCNUM), paste, collapse=", "
+                     SYMBOL=sapply(contents(ath1121501SYMBOL), paste, collapse=", "
+                     DESC=sapply(contents(ath1121501GENENAME), paste, collapse=", "
> Annot[3:4,]
```

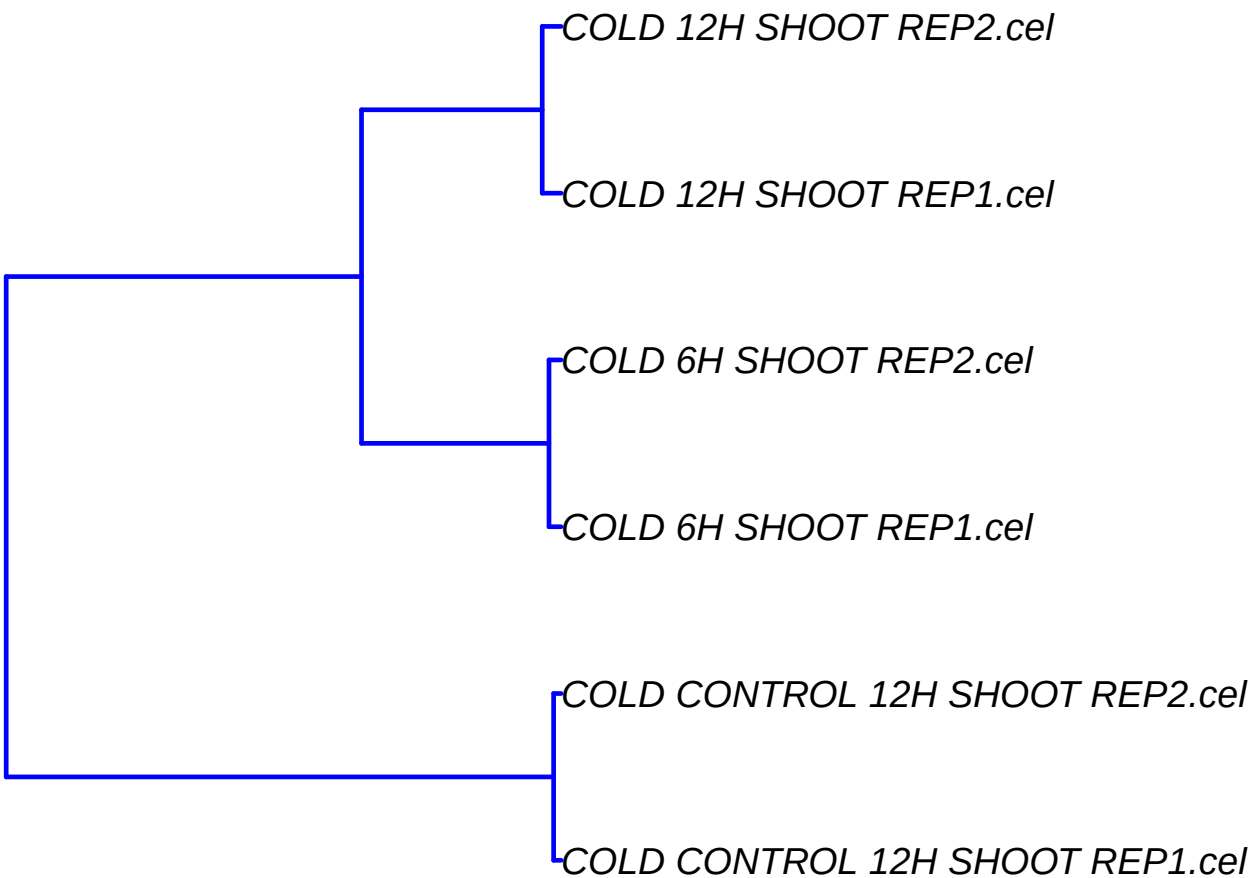
```
            ACCNUM SYMBOL                  DESC
244903_at ATMG00660 ORF149 hypothetical protein
244904_at ATMG00670 ORF275 hypothetical protein
```

```
> ## Merge annotations with expression data
> all <- merge(Annot, my_frame, by.x=0, by.y=0, all=T)
> ## Export data to text file that can be imported into Excel
> write.table(all, file="my_annot_file.xls", sep="\t", col.names = NA)
```

Clustering Tree for Samples

Generates sample correlation matrix and sample tree

```
> library(ape)
> d <- cor(2^exprs(eset_rma), method="pearson")
> hc <- hclust(dist(1-d))
> plot.phylo(as.phylo(hc), type="p", edge.col=4, edge.width=2, show.node.label=TRUE,
```



Quality Report

Generate a comprehensive QC report for the AffyBatch object 'mydata' in PDF format. See affyQCReport Link for details.

```
> library(affyQCReport)
> QCReport(mydata, file="ExampleQC.pdf") # Writes QC report to PDF
```

For more quality control steps, consult the manual section: "Visualization and quality controls" Link

Analysis of DEGs with Limma I

For the DEG analysis, download this targets file to your R working directory: Link

```
> library(limma) # Loads limma library.
> targets <- readTargets("affy_targets.txt") # Import sample information
> data <- ReadAffy(filenames=targets$FileName) # Control CEL files with targets file
> eset <- rma(data) # Normalization with RMA

Background correcting
Normalizing
Calculating Expression

> ## If eset contains absolute intensity values, then they should be transformed to
> ## values. RMA/GCRMA generate log2 values and MAS5 produces absolute values.
> # exprs(eset) <- log2(exprs(eset))
> pData(eset) # Lists the analyzed file names.

                                sample
COLD_CONTROL_12H_SHOOT_REP1.cel      1
COLD_CONTROL_12H_SHOOT_REP2.cel      2
COLD_6H_SHOOT_REP1.cel               3
COLD_6H_SHOOT_REP2.cel               4
COLD_12H_SHOOT_REP1.cel              5
COLD_12H_SHOOT_REP2.cel              6

> write.exprs(eset, file="affy_all.xls") # Exports all affy expression values to tab
```

Analysis of DEGs with Limma II

Set up design and contrast matrices.

```
> ## Creates appropriate design matrix. Alternatively, such a design matrix can
> ## be created in any spreadsheet program and then imported into R.
> design <- model.matrix(~ -1+factor(c(1,1,2,2,3,3)))
> colnames(design) <- c("group1", "group2", "group3") # Assigns column names.
> design[1:2,]
  group1 group2 group3
1      1      0      0
2      1      0      0
> ## Fit a linear model for each gene based on the given series of arrays.
> fit <- lmFit(eset, design)
> ## Creates appropriate contrast matrix to perform all pairwise comparisons. Alter
> contrast.matrix <- makeContrasts(group2-group1, group3-group2, group3-group1, leve
> contrast.matrix
        Contrasts
Levels   group2 - group1 group3 - group2 group3 - group1
  group1              -1               0              -1
  group2               1              -1               0
  group3               0               1               1
```

Analysis of DEGs with Limma III

First, compute estimated coefficients and standard errors for a given set of contrasts. Second, computes moderated t-statistics and log-odds of differential expression by empirical Bayes shrinkage of the standard errors towards a common value.

```
> fit2 <- contrasts.fit(fit, contrast.matrix)
> fit2 <- eBayes(fit2)
```

Generate list of top differentially expressed genes sorted by B-values for first one of the three comparison groups ('coef=1'). The summary table contains the following information: logFC is the log2-fold change, the AveExpr is the average expression value across all arrays and channels, the moderated t-statistic (t) is the logFC to its standard error, the P.Value is the associated p-value, the adj.P.Val is the p-value adjusted for multiple testing and the B-value (B) is the log-odds that a gene is differentially expressed.

```
> topTable(fit2, coef=1, adjust="fdr", sort.by="B", number=3)
```

```
                ID    logFC   AveExpr        t      P.Value    adj.P.Val
9166    254066_at 7.883622 10.130096 98.92804 8.321848e-14 1.898213e-09
21820 266720_s_at 6.106943  8.207481 86.74913 2.416768e-13 2.020105e-09
9175    254075_at 7.019791 10.289670 84.58880 2.965542e-13 2.020105e-09
              B
9166  21.38214
21820 20.68747
9175  20.54548
```

```
> write.table(topTable(fit2, coef=1, adjust="fdr", sort.by="B", number=Inf), file="
```

Limma Exercise

Task 1 The following for loop that outputs the topTable results for all contrasts in one data frame. Write a function that runs this loop and allows the user to choose any column in topTable and outputs the result to an Excel readable file.

```
> limmaDF <- data.frame(row.names=row.names(fit2[[1]]))
> for(i in colnames(contrast.matrix)) {
+         tmp <- topTable(fit2, coef=i, adjust="fdr", sort.by="none", number=Inf)
+         names(tmp) <- paste(names(tmp), i, sep="_")
+         limmaDF <- cbind(limmaDF, tmp[,c(2,5,6)])
+ }
```

GO Term Enrichment Analysis with GOstats

The following example uses the `GOstats` library to identify overrepresented GO terms in the top 100 DEGs of the first contrast of the limma analysis.

```
> library(ath1121501.db); library(GOstats); library(ath1121501cdf)
> affySample <- topTable(fit2, coef=1, adjust="fdr", sort.by="B", number=100)$ID
> geneSample <- na.omit(as.vector(unlist(mget(affySample, ath1121501ACCNUM, ifnotfou
> affyUniverse <- ls(ath1121501cdf)
> geneUniverse <- as.vector(unlist(mget(affyUniverse, ath1121501ACCNUM, ifnotfound=N
> params <- new("GOHyperGParams", geneIds=geneSample, universeGeneIds=geneUniverse,
+               annotation="ath1121501", ontology="MF", pvalueCutoff=0.5, conditiona
+               testDirection = "over")
> hgOver <- hyperGTest(params)
> summary(hgOver)[c(3,7),c(1,2,5:7)]
```

```
      GOMFID      Pvalue Count Size                                Term
3 GO:0019904 0.002749447     2   16    protein domain specific binding
7 GO:0005432 0.019547099     1    4 calcium:sodium antiporter activity
```

```
> htmlReport(hgOver, file = "MyhyperGresult.html")
```

Venn Diagram for DEG Sets

The following example stores the gene identifiers for all three DEG comparisons in a list that meet the following threshold criteria: at least 2-fold change and an adjusted p-value of less then 0.01.

```
> source("http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/My_R_Scripts/overLapper
> deglist <- sapply(colnames(contrast.matrix), function(x) {
+         tmp <- topTable(fit2, coef=x, adjust="fdr", sort.by="B", number=Inf)
+         affyids <- tmp[(tmp$logFC >= 1 | tmp$logFC <= -1) & tmp$adj.P.Val <= 0.01,
+         geneSample <- unique(as.character(na.omit(as.vector(unlist(mget(affyids, a
+         ifnotfound=NA))))))
+ })
> sapply(deglist, length)
```

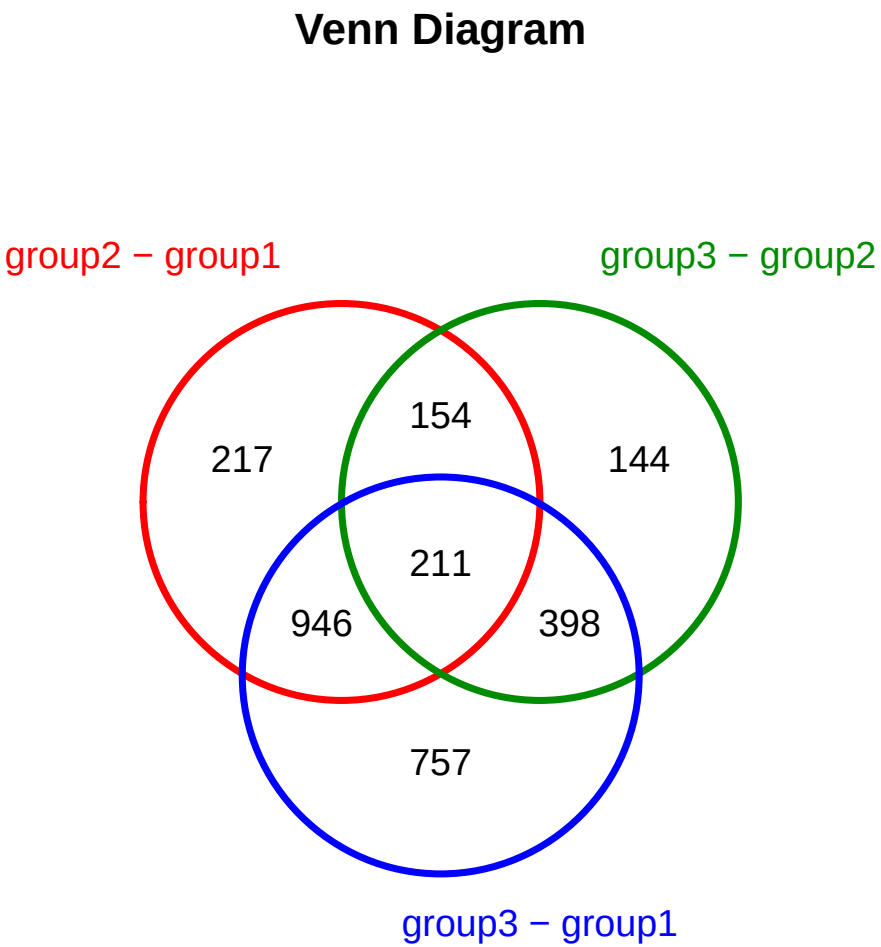
```
group2 - group1 group3 - group2 group3 - group1
           1528             907             2312
```

```
> OLlist3 <- overLapper(setlist=deglist, sep="_", type="vennsets")
> counts <- sapply(OLlist3$Venn_List, length)
> counts[1:4]
```

```
                group2 - group1                 group3 - group2
                            217                             144
group3 - group1 group2 - group1_group3 - group2
                            757                             154
```

Venn Diagram for DEG Sets

```
> vennPlot(counts=counts)
```



Clustering of DEGs with Euclidean Distances I

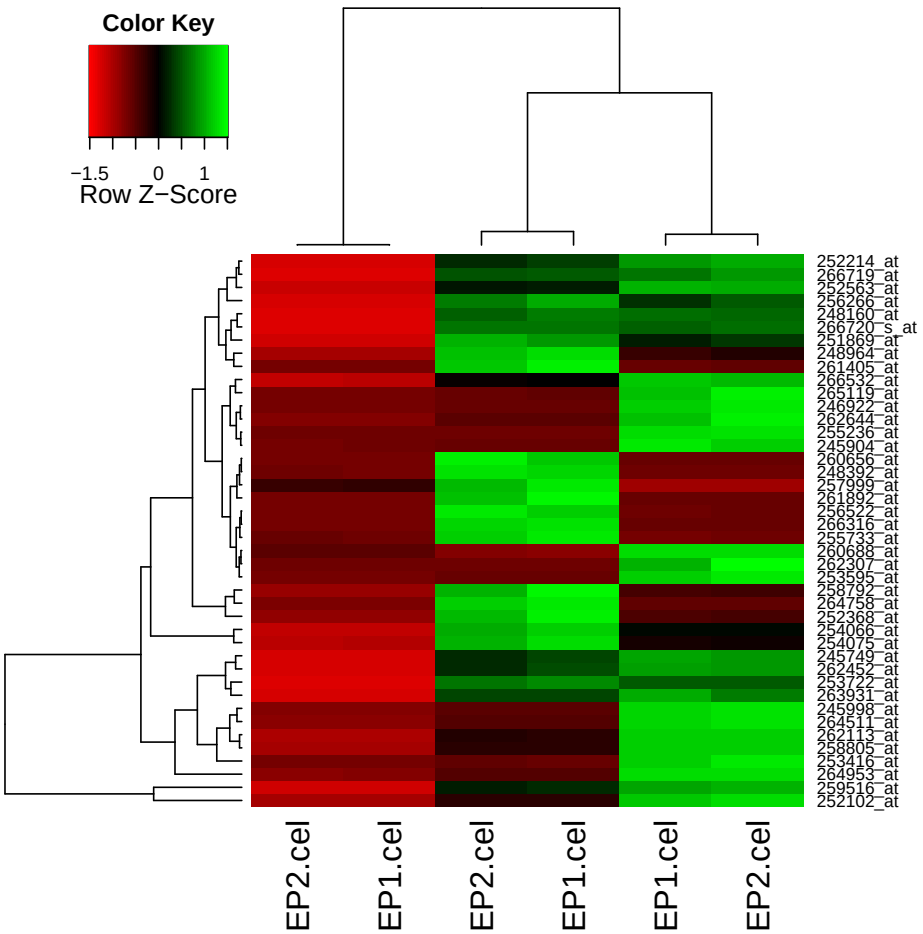
Obtain first 20 DEGs from each DEG comparison and perform hierarchical clustering on the expression matrix containing only those genes.

```
> library("gplots")
> deglist <- sapply(colnames(contrast.matrix), function(x) {
+         tmp <- topTable(fit2, coef=x, adjust="fdr", sort.by="B", number=Inf)
+         affyids <- tmp[(tmp$logFC >= 1 | tmp$logFC <= -1) &
+                        tmp$adj.P.Val <= 0.01, "ID"][1:20]
+ })
> y <- 2^exprs(eset)[unique(as.character(deglist)),]
```

Clustering of DEGs with Euclidean Distances II

The following performs hierarchical clustering with Euclidean distances and plots the result in form of a heat map.

```
> heatmap.2(y, col=redgreen(75), scale="row", trace="none", density.info="none")
```



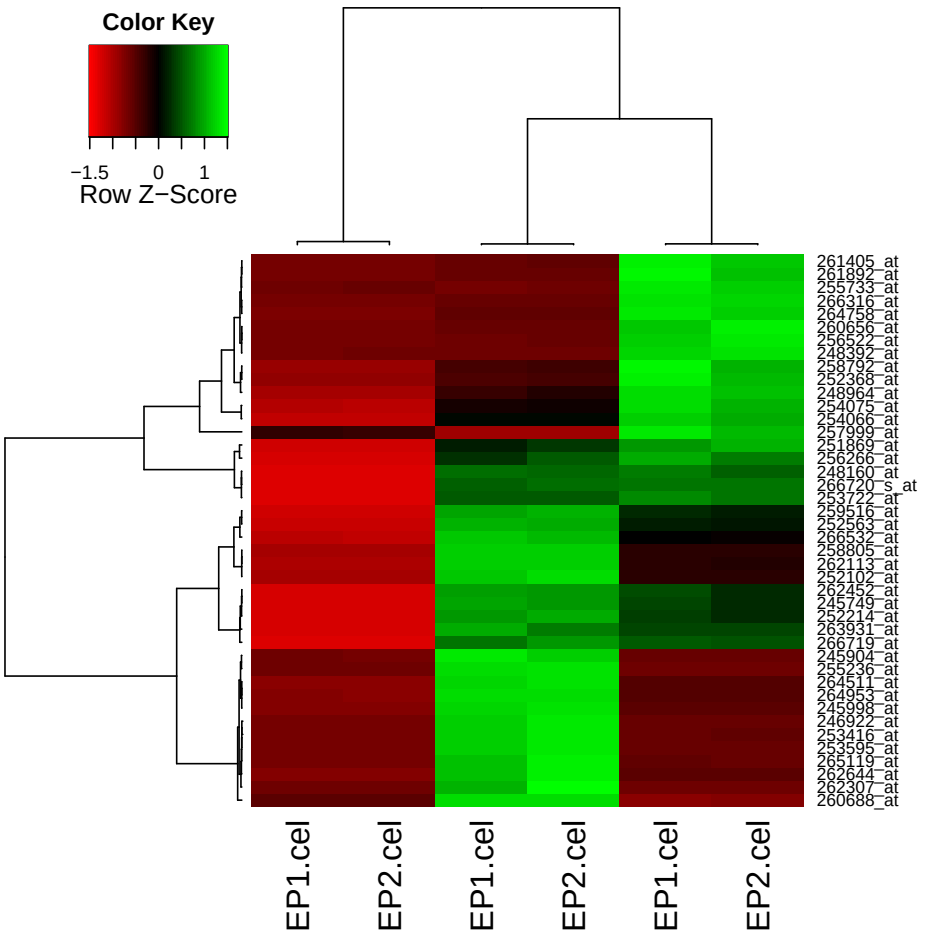
Clustering of DEGs with Correlation-based Distances I

Same as before but using correlation-based distance measures

```
> hr <- hclust(as.dist(1-cor(t(y), method="pearson")), method="complete")
> hc <- hclust(as.dist(1-cor(y, method="spearman")), method="complete")
```

Clustering of DEGs with Correlation-based Distances II

```
> heatmap.2(y, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), col=redgreen(75),
+           scale="row", density.info="none", trace="none")
```



Homework Tasks

A. Generate expression data with RMA, GCRMA and MAS 5.0. Create box plots for the raw data and the RMA normalized data.

B. Perform the DEG analysis with the limma package and determine the differentially expressed genes for each normalization data set using as cutoff an adjusted p-value of $\leq$0.05. Record the number of DEGs for each of the three normalization methods in a summary table.

C. Create for the DEG sets of the three sample comparisons a venn diagram (adjusted p-value cutoff $\leq$0.05).

D. Generate a list of genes (probe sets) that appear in all three filtered DEG sets (from B.).

$\Rightarrow$ Command summary: source("homework_script.R")

Session Information

```
> sessionInfo()

R version 2.15.2 (2012-10-26)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] gplots_2.11.0          MASS_7.3-22            KernSmooth_2.23-8
 [4] caTools_1.13           bitops_1.0-4.2         gdata_2.12.0
 [7] gtools_2.7.0           xtable_1.7-0           GO.db_2.8.0
[10] GOstats_2.24.0         graph_1.36.1           Category_2.24.0
[13] limma_3.14.3           ape_3.0-6              ath1121501.db_2.8.0
[16] org.At.tair.db_2.8.0   RSQLite_0.11.2         DBI_0.2-5
[19] ath1121501cdf_2.11.0   ath1121501probe_2.11.0 AnnotationDbi_1.20.3
[22] affy_1.36.0            Biobase_2.18.0         BiocGenerics_0.4.0

loaded via a namespace (and not attached):
 [1] affyio_1.26.0         annotate_1.36.0       AnnotationForge_1.0.3
 [4] BiocInstaller_1.8.3   gee_4.13-18           genefilter_1.40.0
 [7] GSEABase_1.20.1       IRanges_1.16.4        lattice_0.20-10
[10] nlme_3.1-105          parallel_2.15.2       preprocessCore_1.20.0
[13] RBGL_1.34.0           splines_2.15.2        stats4_2.15.2
[16] survival_2.36-14      tools_2.15.2          XML_3.95-0.1
[19] zlibbioc_1.4.0
```