```
### TUTORIAL #1 -- LOADING DATA AND PRE-PROCESSING IT ##########################
# The goal of the first tutorial is to learn how to important Affymetrix data
# into R and to pre-process it appropriately. We are going to be working with
# dataset from the largest repository of microarray data, which is the Gene
# Expression Omnibus (GEO) repository hosted by NCBI.
#
# We are going to look at a study of the response of Burkitt's lymphoma cells
# to perturbation of the mRNA abundances of two oncogenes: Foxm1 and Myb. This
# study was selected because it is a small cancer dataset that allows for both
# univariate and multivariate statistical analyses. The dataset is small in that
# it contains only 3 experimental conditions, each with three technical
# replicates, for a total of 9 arrays. Further, it uses the relatively old U95Av2
# Affymetrix array, which assays about 12,000 transcripts. Thus there will be
# approximately 100,000 data-points in this dataset.
#
# All GEO datasets are given an accession number, and the accession for the
# dataset we will be using is GSE17172. The main GEO page for any given dataset
# has a lot of information about the study. For this dataset it is at:
# http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE17172
#
# Normally you would download the supplementary file containing the raw data.
# This file will be named: GSE39277_RAW.tar
# You can decompress this file using an archive-manager (e.g. winzip or 7zip),
# and subsequently decompress the .CEL files it contains. However in this case
# for the purposes of easing the tutorial, we have decompressed and uploaded all
# the data to the CBW wiki:
# bioinformatics.ca/workshop_wiki/index.php/Bioinformatics_for_Cancer_Genomics_2013_Workshop_Wiki
#
# There are 9 files to download. I recommend storing these files in a new directory
# on your computer, and keeping them separate from the scripts you will develop to
# analyze this dataset. Strict separation of data and code is a fundamental principle
# of good software-engineering that will make your life much, much easier over time!

### Question #1 -- Loading Data ################################################
# Use ReadAffy to load the data into an R object.

# You should have the affy library installed on your computer as part of your
# BioConductor installation. You can very this by typing:
library(affy);

# If this does not succeed, you can install the package by typing:
source("http://bioconductor.org/biocLite.R");
biocLite();
# This will install a series of BioConductor "core" packages

# Now, you will want to learn how to install the data. The function to do this is:
?ReadAffy;

# You will also want to consult the affy package primer on how to use this function.
# On machines with functional PDF processing you can view the primer by typing:
vignette('affy');

# Sometimes the vignette function fails on machines with improperly-configured PDF
# readers. The vignette document is still available in your R program directory,
# under the library/affy/ subfolder.
```

```
### Question #2 -- Pre-Processing Data ########################################
# A) Use the expresso() function to pre-process the data with the RMA method.
#
# Next, you will want to pre-process the data. There are a very large number of
# possible pre-processing methodologies. The affy package provides access to
# these through the function:
?expresso

# Use this function to pre-process the data in two different ways: according to
# the RMA model:
#      background correction            RMA
#      normalization method             quantile
#      pm correction method             pmonly
#      summary method                   median polish


# B) Re-preprocess the data using an alternative CDF
# When you initially loaded the CEL files from it into R, it is likely that your R
# session automatically went to download the default Affymetrix CDF file. This file
# was probably taken from the BioConductor web-site directly

# Fortunately ReadAffy() can easily accept alternative CDF files (and thus
# alternative Probe mappings) using the cdfname argument.  Your first challenge
# is to re-run the analyses from the first tutorial using an alternative
# mapping.  To get the mapping you will go to:
# http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/CDF_download.asp
# You will need to download the latest Entrez Gene ID CDFs for this microarray
# platform (U95Av2)and install them into R (using standard package-installation
# techniques -- ask for help if you get stuck here).

# Once you have the package installed, re-load the data using ?ReadAffy and then
# repreprocess with the alternative CDF (again using RMA).
```

```
### Question #3 -- Assess data quality ########################################
# A) Make histograms or density curves of the raw and pre-processed data
# B) Make a heatmap of the correlation matrix from the entire experiment

# Once your data has been pre-processed you will want to determine if it is
# of sufficient quality to actually do any meaningful analysis. There are a
# large number of EDA (exploratory data analysis) plots that can be constructed
# to help in this, and several are already implemented in R/BioConductor, mostly
# in the affy package.

# Many of the QA/QC plots you might wish to make are directly available from
# within the affy package itself.  To make histograms of the raw data you can
# simply use hist on an object of class Affybatch.  For example
raw.data <- ReadAffy();
hist(raw.data);

# To make density curves of the normalized data:
eset.rma <- expresso(raw.data, ...);
?plotDensity

# And to make a heatmap of the inter-sample correlations you will need to:
# a) use ?cor to generate a n x n correlation matrix (n = the sample-number)
# b) use ?heatmap to create a mapping of samples
# Consider changing the default scale parameter in heatmap -- can you see why?
```

```
### Question #4 -- Univariate statistical analysis ############################
# A) assess differential expression for each gene in the Foxm1 vs. NT comparison
# B) assess differential expression for each gene in the Myb vs. NT comparison
#
# Once we are convinced of data-quality, the next step is to do a detailed
# statistical analysis to assess which genes are being changed in this experiment.
# This study involves knock-down of two genes via siRNA, with a non-specific siRNA
# (non-targeting, NT) used as a control. There are some interesting multivariate
# statistical approaches viable here, but we can start with some straight-forward
# univariate analyses.
#
# The first step to doing this is to annotate the Affymetrix data with the
# sample classifications. This can be done using the *phenodata* slot. To
# achieve this you will want to create a tab-delimited text file that identifies
# the filename of each sample (first column) and which experimental group it
# belongs to (second column). You can read this phenodata directly into R
# by using the phenoData parameter in ?ReadAffy

# Next, you will want to use standard univariate statistical techniques. It is
# controversial whether a parametric approach (i.e. a student's t-test):
?t.test
# or a non-parametric test (i.e. Wilcoxon rank-sum test; u-test):
?wilcox.test
# is more appropriate. With very low n studies, in general a t-test will be superior

# You can use a for-loop to run a t-test or a u-test on each ProbeSet. Save these
# values as a vector for later use. Repeat this procedure for both comparisons.

# C) adjust for multiple-testing
# Any experiment of this magnitude will generate a large number of false positives by
# chance alone. If we select a 5% significance threshold and use a microarray platform
# containing 10,000 genes, then we will have 0.05 x 10000 = 500 false-positives by
# chance alone.

# To control for this large effect, we can use a multiple-testing adjustment. There
# are several types, but by far the most common for genomics studies is the false-
# discovery rate adjustment (FDR). FDR adjustment slightly changes the interpretation
# of a p-value: instead of giving the chance of a false-positive (i.e. of type I
# error) on a single test, an FDR-adjusted p-value gives the fraction of all
# statistically-significant tests that will be false-positives (i.e. it gives the
# type I error rate amongst statistically-significant hits).

# It is easy to adjust for multiple-testing using the function:
?p.adjust

# Create a plot comparing the naive and the multiple-testing-adjusted p-values.
# Create a histogram showing the distributions of naive & adjusted p-values.

# D) repeat these analyses for both default and alternative CDFs
```

```
### Question #5 -- Compare the two experimental conditions ###################
# Another experimental question one might ask is: how similar are the genes
# affected by knock-down of Myb to those affected by knock-down of Foxm1. There
# are two basic classes of techniques to look at this.

# A) Make plot(s) to compare and contrast all p-values between the two conditions
# The most direct way to compare these two experiments is simply to plot the
# p-values from one study against the p-values from the other (i.e. a scatterplot).
# The R function ?plot can do this.
#
# However a plot like this can miss the most interesting genes, which will be
# clustered around the origin (i.e. close to p = 0). One solution to this problem
# is to plot the data in log10-space rather than in normal-space.

# Compare the two plots: which is clearer? Do they give different information?

# B) Make plot(s) to compare statistically-significant hits between the conditions
# The other basic approach is to compare only those genes that are statistically
# significant. If a gene does not have a reasonably small p-value in at least
# one of the two experiments, then it is unlikely to be of significant interest
# in follow-up studies. It is therefore biologically reasonable to focus only on
# statistically-significant results.
#
# The most common plot for doing that comparison is a Venn diagram. There are a
# few different Venn diagram packages in R, and we will use the VennDiagram package
# here. If necessary install the package and then load it using:
library(VennDiagram);

# You can then directly compare two vectors using:
vector1 <- 1:150;
vector2 <- 121:170;
venn.diagram(
       list(
              A = vector1,
              B = vector2
              ),
       filename = "Venn_test.tiff"
       );

# Make Venn diagrams that compare genes that are statistically significant at a
# reasonable range of p-value thresholds.
```

```
### Question #6 -- Compare the experiments using effect size ##################
# A) Calculate fold-changes for every gene in both experiments.
# So far your analyses have only looked at statistical significance (p-values,
# adjusted or not) as a measure of how interesting a gene is. However a gene
# could conceivably be upregulated by Myb and down-regulated by Foxm1. Genes
# showing this type of divergent behaviour are being treated the same as genes
# showing convergent behaviour. We therefore need to consider effect-size.
#
# In statistics, looking at raw p-values is never sufficient. Instead one always
# needs to assess how large the effect being studied is -- is it sufficiently
# large to be of interest? With sufficient replication even very small effects
# can become statistically significant, but this does not necessarily make them
# biologically or clinically important.
#
# The most common measure of effect-size is the fold-change. Because RMA data is
# in log-space, fold-changes are simply calculated as:
#      mean(treated-samples) - mean(control-samples)
# If we were in normal-space this would instead be:
#      mean(treated-samples) / mean(control-samples)
#
# Calculate a fold-change for every gene in each experiment. Look at the distribution
# of effect-sizes using a histogram (?hist): can you see the advantage of log-
# transforming the data?
#
# B) Create volcano plots
# The most common way to visualize the relationship between effect-size and
# statistical significance is called a volcano plot. This is a scatter-plot with
# the -log10 of the p-value on the y-axis (i.e. a measure of statistical
# significance, with larger values being more significant) and the effect-size
# (in our case fold-change) on the x-axis. Create this plot for each experiment.
#
# C) Compare effect-sizes between experiments
# In question #5 you saw how to compare either all genes or only statistically-
# significant ones between the experiments. You can now do the same using fold-
# change cutoffs, or using combined cutoffs. For example, try this comparison:
#      genes with effect-sizes of at least 25% and unadjusted p-values below 0.05
```

```
### Question #7 -- Compare the experiments using effect size ##################
# So far all of our analysis has been done using the RMA pre-processing
# technique. What happens if you repeat these studies using MAS5? How much do
# the specific results change?

# A) Repeat pre-processing with MAS5 using the ?expresso function again

# B) Compare the results of the two methods overall
# To compare the results of the two methods you will want to get access to the
# "expression matrix".  This can be accessed using the function:
?exprs

# You will notice that the MAS5 data is in normal-space and the RMA data is in
# log2-space, so you will want to bring them to a common-space for plotting
?log2

# You will also want to see how correlated the results of the two analyses are.
# Calculate the correlation between the MAS5 and RMA preprocessed results

# C) How many genes change ordering between the two pre-processing techniques?
# How many of the ProbeSets on the array change ordering between the two pre-
# processing methods.  That is, ask if the ranks of samples are the same between
# the two expression matrices for each row, and count the number of rows that are
# altered. This can be easily visualized with a histogram.

# This turns out to being equivalent to asking if the Spearman correlation is
# exactly 1.0, so this question can be reduced to calculating that correlation
# between every row of the two matrices.  The key function is:
?cor

# D) Repeat your univariate statistical analysis with MAS5-processed data
# Do you get similar or different gene-lists using MAS5-processed data instead
# of RMA processed data. Repeat the gene-list comparison techniques (p-values,
# effect-sizes, and Venn diagrams) you learned above to compare the two
# methods.
#
# Does it matter significantly how the data is pre-processed?
```