# Project for Introduction to R Course

Leslie Cope, Elana Fertig, Luigi Marchionni

May, 2012

The extended exercise described here is designed as a typical data analysis project. We are happy to make specific suggestions for any of the tasks but try to plan it out yourself first. The description is intentionally vague on methods and functions since scientific questions don't come with precise instructions for answering them.

## The Data

In 1999, Todd Golub and collaborators at the Whitehead Institute published this paper in **Science**.

**Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring T. R. Golub, et al. Science 286, 531 (1999); DOI: 10.1126/science.286.5439.531**

This study has become one of the milestones in the history of gene expression microarrays, instrumental in establishing that microarray arrays are reliable enough to be used for such complex tasks as tumor class discovery, and prediction. The accompanying data files contain a slightly modified version of the original study data. We have reduced the number of genes to speed up computation, and removed redundant probes so that each gene is only measured once. We have also messed with the original sample names so that the project can include the kinds of data grooming tasks that research datasets often require before they can be analyzed.

## The project

The main goal of this project is to provide an opportunity to practice the course material, by carrying out a complete data analysis, from reading data in and cleaning it up, through the analysis and reporting of results. We encourage anyone who wishes to, to try additional or different analyses of the data, and will be happy to answer questions or make suggestions in class, or by email, as you work on these exercises.

- Read the gene expression and clinical data files into R using one of the data import functions available in R. Make sure you understand the format, and size of each. And make sure that the same samples appear in both.
- Do some exploratory analysis of the clinical data, to be sure you understand it. This might include calculating the 5 number summary and making histograms of the individual quantitative variables, and tables of the qualitative variables, to get a sense of the range and distribution of values for each. You might look at pairs of variables as well, using scatter

plots, side by side box plots, and 2 X 2 tables, to identify correlations. To visualize the entire gene expression dataset, look at the help files for the `heatmap()` function and try it out to visualize the expression data.

- When you compare gene expression and clinical data files, you will notice that there are some anomalies in the coding of sample names. Write a script to automatically match the names.
- Among the gene specific results reported in the Science paper, were findings that Cyclin D3 (CCND3) was more expressed in ALL whereas CD33 was more highly expressed in AML.
  - Identify these genes in the expression data and try to verify the finding by performing t-tests on them. This will require getting the AML-ALL designations from the clinical data file.
  - Use whatever plotting function(s) you wish to visualize the expression differences for these genes. Modify the plotting arguments to customize the plot so that it is easy to read, including the addition of informative axis labels and titles, colors, adjustments to the size of plotting elements, axis labels for easier viewing, etc... Try to automatically insert the appropriately formatted results from the t-test into the title. Try out the `legend()` function.
- Use loops or apply to run t-tests on every gene. Better yet, do it both ways, this project is all about practicing. Build a data frame for the output, include the probe and gene names, and the t-statistic, and p-value, rounding each appropriately. Organize it into some reasonable order, and write the top genes out to a file.
- **Challenge** Write a loop to make plots for all of the top genes. Use the plotting format you developed above for CCND3 and CD33, and automatically paste the gene name and t-test results into a title for each page. Run the whole thing inside of `pdf()` and `dev.off()` commands to make a multipage pdf. Or better yet, use `pdf()` and `dev.off()` inside the loop to make separate pdfs for each gene, labeling each file by gene.

## Requirements

To receive full credit for the project:

- The student, while not required to follow the outline provided above exactly, should make a real effort to practice what was covered in class in a complete project that includes reading in data, doing something intentional to it and reporting the results of that analysis in an organized way.
- Results should be embedded in a well-written document that clearly describes the intention of the analysis, and presents results and interpretations in a logical and readable way. Plots should have meaningful labels, and should incorporate such elements as color, multiple plotting symbols, and legends as necessary for clarity. Summary statistics should be rounded to a relevant number of significant digits, etc...
- Code should employ self-documenting variable names, and include comments as needed so that an experienced R user could follow it easily. The code **must** meet standards of reproducibility.

This means that another investigator, with the same files in an appropriately named directory, could reproduce your final results simply by running `source()` on your clean R script file. This will be easier if you designate a discrete work space for the project, and keep everything there, always work from a script you write rather than working interactively at the R-prompt, organize your code into logical sections, and annotate your code with comments so you can keep track of what each chunk is supposed to do.