Your Homework: The task

1. Download this microarray gene expression data set for 4 cell lines:
   http://www.broadinstitute.org/mpr/publications/projects/SOM_Methods_and_Applications/data_set_HL60_U937_NB4_Jurkat.txt
2. Read the description of the data set:
   http://www.broadinstitute.org/mpr/publications/projects/SOM_Methods_and_Applications/Datasets_description.txt
3. Using Python and pandas:
   a. Import the data file.
   b. Convert the data file into a DataFrame.
   c. Write functions to answer questions A - G (following page).

Your Homework: Questions

A. How many distinct genes are represented in the data set?
B. Which two time points are the most highly correlated for each cell type?
C. Which two cell types are the most similar?
D. It is often useful to know which genes change very little across samples for the sake of normalization or calibration. Based on this data set, what are ten good candidates for genes to use to calibrate machinery or analyses across all these samples?
E. Do any genes show two-fold higher expression at 24 hours versus 0 hours for all four cell types? If so, which ones?
F. Which genes are differentially regulated (at least two-fold higher or lower) in HL60 cells as compared to U937 cells at 0 hours?
G. Take the list of Gene Accession codes from (F), and run them through the DAVID ontology analyzer. (at http://david.abcc.ncifcrf.gov/summary.jsp . These are GenBank Accession codes.) Are there any enriched ontology terms?

Your Homework: Restrictions

- You should turn in the answers to questions A - G, and also the code used to generate the answers– print outs or emailed to me&Jeremy.
- You may work in groups, but each student should turn in their own assignment.
- Python is not just for scripting! Your final code should be structured as an object that has methods that separately import the file, determine each desired answer, etc. (See example on following page.)
- Points (i.e., mad props) given for style, clarity, and reusability in code.

Your Homework: Example

```
from pandas import DataFrame
from pandas.io import parsers

class ExpressionAnalyzer(object):
    '''
    Responsible for importing and analyzing gene expression data.
    Expects data with named samples as columns and genes as rows.
    '''
    def import_file(self, filename):
        ''' Import data from file and return a DataFrame object. '''
        ...
...

if __name__ == '__main__':
    ea = ExpressionAnalyzer()
    ea.import_file('path_to_file')
    ...

========
cmm171-164:~ karmel$ python expression_analyzer.py
```