Lecture 4:
Slice, Dice, Combine, and Sort

Christopher Coletta

Goal of this Talk: Data manipulation

- Building up data into spreadsheet-like structure
- Reading from spreadsheets (.csv, .tsv)
- Representing that data on screen
- Sorting the data
- Writing to spreadsheets

Input data file: input_data.txt

```
agap5 = -5.86781126186,-1.13990744634,-2.2598482697,-4.97802930212
caps = 2.90522225548,8.19819454478
cyp1a2 = 1.04536416399,1.63284246304,2.10416875136
c13orf1 = -3.77857966439
adamts6 = 9.44871693381
rcn2 = -8.63273989972
ajap1 = -4.27351557343,-6.63948277453,-7.05342421886,-6.41602358833
prdm10 = -3.68538907371
c9orf33 = -0.678288893194,-0.975635989755,-0.712601094672
pcm1 = 5.33507040572
rragd = -3.48875558599
dtnb = -4.13157675273,-0.685625121506,-1.09860554796
loc400657 = 0.393028998087,0.440524379233
eif5al1 = 2.75030133744
dnd1 = 2.00166117373,-0.215063094773,0.45814010154
fastkd3 = 1.45892125868,1.68350324377,0.980917474842,1.55532886471
gopc = 0.263825633551,-2.95259832379
papd5 = -0.818059153229
otx2 = 6.19661410944,-0.532787139483
dusp7 = 3.34429619698,-0.712979754952,0.587178255415,4.03263248242
ctnnbip1 = -2.26229634384,-2.18622008043,0.0154020127482
```

Case Study: Gene list w/ z-scores

- Read raw data text file
  - 100,000 lines of genes and z-scores
  - Genes mentioned more than once
- Parse and generate simple statistics
  - Collect all repeats, and sort on up/down regulation
- Generate report
- Write statistics into a new file which can be opened in Excel

Topics Covered

- File IO
- Parsing strings
- Comprehensions (list/dict/set)
- Sorting using a lambda function
- String interpolation/formatting

File operations

- Open a file for reading

```
my_file = open( "/path/to/file" )
```

- Read in lines from file iteratively

```
for line in my_file:
    <do something with line>
```

- Important! Close the file when done!
  - `my_file.close()`
  - or use context management keyword `with`

using string.`split`()

- 1st argument to `split()`: the separator

  ```
  list_o_strs = some_string.split(“,”)
  ```

  – default separator is space
- 2nd optional arg, number of splits you want done
- Can also start from the back using `rsplit()`
- See also: reverse operation string.join():

  ```
  joined_str = ‘,’.join( some_list_or_tuple )
  ```

list comprehension

- Given `old_list=['45','67','92']`
- The following are equivalent:
  - For loop

```
new_list = []
for item in old_list:
    new_list.append( int( item ) )
```

  - List comprehension

```
new_list = [ int(item) for item in old_list ]
```

- Conditional list comprehension:

```
new_list = [ int(item) for item in old_list if <condition> ]
```

Sorting lists of tuples

- http://wiki.python.org/moin/HowTo/Sorting/
- Can specify which column to sort using `key` and `lambda` function with one argument
- Can specify special sorting using `lambda` function that takes 2 arguments, A and B
  - returns -1 if A < B
  - return 0 if A == B
  - returns 1 if A > B
  - `cmp( A, B )` is a handy function that does this

string.format()

- Use string with placeholders {}, and substitute values into placeholders
- Stringifies arguments automatically
- Can set alignment and padding of value within placeholder
- Can set the number of decimal places

`format()` Syntax Mini-language)

- http://docs.python.org/2/library/string.html#format-specification-mini-language