Path 218 Discussion 2 (07/03/14): Some real world experiences with data files, strings, and lists

- **File-related tools you may use**
  - A statement **F1=open("in.txt",mode='rU')** opens file **in.txt** to read, associating variable name "**F1**"
  - A statement **for L1 in F1:** goes through the file line-by-line (**L1**) with the subsequent indented code block.
  - A statement **F2=open("out.txt",mode='w')** opens file **out.txt** to write, associating variable name "**F2**"
  - A statement **F2.write(StrF1)** then writes string **StrF2** at the end of the file referred to by **F2**
  - Once finished with a file, close it (with command **F1.close()** for file F1) to ensure python updates the file
- **String-related tools you may use**
  - The expression **S1.strip()** yields a derivative of **S1** with any "white space" removed from both sides
  - **S1[0]** is the first character of **S1**, **S1[1]** is the second character... **S1[-1]** is the last character
  - **S1[a:b]** yields a derivative list of **S1** starting with character a and ending with character b-1
  - **S1.replace('x','y')** yields a derivative of **S1** replacing every occurrence of **'x'** with **'y'**
  - **S1.find('y')** yields the position in S1 of the first occurrence of **'y'** in S1 (-1 if there is no occurrence)
  - **S1.rfind('y')** yields the position in S1 of the last occurrence of **'y'** in S1 (-1 if there is no occurrence)
  - **S1.count('y')** yields a count of occurrences of 'y' in S1 (0 if there is no occurrence)
  - **str(*number*)** yields a string corresponding to any given number (e.g., **str(*11*)** yields the string **'11'**)
- **List-related tools you may use**
  - A statement **A1=[]** creates an empty list **A1**
  - The expression **A1[0]** yields the first entry in list **A1**, **A1[1]** yields the second entry, etc
  - A statement **A1.append(X1)** modifies list A1 by adding element **X1**to the end of list **A1**
  - A statement **A1.extend(B1)** appends all elements of list **B1** to the end of list **A1**
  - **n1=A1.index(*x1*)** yields a value **n1** that is the index (position) of the first occurrence of ***x1*** in list **A1**.
  - The expression **A1=[0]*100** creates a list **A1** with 100 zeros
  - **range(X1)** yields a list of the first **X1** integers (starting with zero): so **range(5)** yields **[0,1,2,3,4]**
  - **for i in A1:** Goes through the ensuing block with i as each element of list A1 (e.g., **for i in range(10):**)

---

1. What's in a name? (A quick lesson in naming; type the following into the live interpreter/python-shell)

```
A = ['hey','I','just','met','you']
B = ['and','this','is','crazy']
C = A
A.extend(B)
```

What is A?, What is C?

```
C.extend(['but',"here's",'my','number'])
```

What is A?, What is C?

```
A = ['so','call','me','maybe']
```

What is A?, What is C?

Surprised? This is why we need to be careful if we want to get a copy of a list to work with independently: One trick **A=C[:]** A new list named "A" with all elements of C serves as a copy of C that can be modified.

---

2. A micro RNA parser: count miRNA abundances in an RNA sequence dataset from a (real) dog

- File "mature.fa" contains a list of known miRNAs from diverse species. It is in fastA format, with each miRNA represented by a naming line followed by a sequence line, such as

```
>cfa-miR-448 MIMAT0001535 Canis familiaris miR-448
UUGCAUAUGUAGGAUGUCCCAU
```

- File "myDogsRNA.txt" contains a list of small RNA sequence reads taken from a cell population derived from our trusty dog Fido (actually we don't have a dog named fido, but these *are* real sRNA reads from a canine lymphocyte sample sequenced at the Max Planck Institute in Berlin). The sequences in myDogsRNA.txt have no barcode at the beginning (they start immediately with the small RNA sequence) but they do have a linker at the end with a sequence that starts "TCGT". As with many such experiments, many of the reads are "junk".

Step A. Read through the mature.fa file, generating two lists : One will be a list of miRNA names and one a list of miRNA sequences.

Step B. Define a new list to store the number of hits for each relevant miRNA. The list will have one entry for each canine miRNAs and the starting value for each counter will be zero.

Step C. Read line-by-line through the myDogsRNA file, trimming linkers from each line, testing each trimmed line for matches to the known miRNA list, and incrementing counts for each miRNA as matches are found.

Step D. Write the resulting table of miRNA names, sequences, and incidences into a new file.

---

3. Some fun with E. coli

File ColiDH1.fa contains the E. coli DH1 genome sequence. We'll do some composition analysis of this genome.

a. Count A, C, G, and T bases in the E. coli DH1 genome

b. Count each dinucleotide (AA,AG,AC,AT,GA,GG, etc) in the E. coli DH1 genome