Assignment 1 Review

2/2/18
Easy ways to get points!

- **Name output files** as required in submission folder
  
  ~/assignment1/submission/random_seq_1M.txt
  ~/assignment1/submission/make_seq.py
  ~/assignment1/submission/random_seq_1k.txt
  ~/assignment1/submission/mod_make_seq.py

- **Include usage statement!**

- Please note that we are using **python3** in this class
  
  python3 nuc_count.py hs_ref_GRCh38.p7_chr20.fa
  python3 nuc_count.py hs_ref_GRCh38.p7_chr20.fa

- **Comment** your code. (Good job on this!)
TC: 0.080
TG: 0.030
TT: 0.049

Question 3
Run the modified nuc_count.py for both human chr20 and your generated ‘random_seq_1M.txt’ from part 4. Compare the two lists of frequencies. What are the differences? Can you provide a biological explanation for these differences?

What to turn in
- Two modified scripts nuc_count.py and make_seq.py.
- A completed README.txt
- The sequence file random_seq_1M.txt

These four files should be in your assignment1/submission folder.

Note: to copy your work files to your submission folder, type

```
$ cp <file_name> ~/assignment1/submission/
```

where `<file_name>` is the name of the file you want to copy.
Counting with **sliding window**

• Why can’t we use `count()`?

```python
>>> nucleotides = "AAANNNNCACAACA"
>>> nucleotides.count("AA")
2
>>> nucleotides.count("CA")
3
```

count() does not allow for overlapping patterns

• Mind the `range()`

```python
>>> range(len(nucleotides))
range(0, 14)
>>> list(range(len(nucleotides)))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

- By default, left lower range starts with 0
- Higher range is exclusive
How to ignore nonATCG characters?

- **Mononucleotide frequency**
  - Total # nucleotides == sum # of A, C, T, Gs
  - Total # nucleotides != length of the sequence
    - N is included

- **Dinucleotide frequency**
  - Total # of dinucleotides == sum # of AA, AC, ..., GT, GGs
  - Total # of dinucleotides != (length of sequence) − 1
    - NN, AN, NA are included

- Remove N from the sequence? E.g. ["ACTGCAGTNNNNNNNNTTACGCCTGA"]
  - Remove & break the sequence into sub-sequences
    - ["ACTGCAGT", "TTACGCCTGA"]
  - Remove & append to make a new sequence
    - ["ACTGCAGTTTACGCCTGA"]
If-else statement

• Make_seq.py: random number generator

• Set range in if-else statement
  • General if-else syntax:

```python
if freq < a_freq:
    nucleotides += "A"
* elif freq < a_freq + t_freq:
    nucleotides += "T"
```

* "elif" is short for "else if"
Test runs for your script

• Is my code doing what I have expected?
  • Running on real dataset can be a pain
  • D.I.Y your own toy dataset -> test your code FASTER! 😊

• For nuc_count.py:
  • Create a short sequence file in FASTA format
  • $ vi toy.fa
  • Predicted results

> toy sequence
AAACNNNNNCT

• For make_seq.py:
  • $ python3 make_seq.py 100 0.4 0.4 0.1 0.1
  • See if sequence length, nucleotide proportions are correct