# Genome 559

Web page:
http://evolution.gs.washington.edu/gs559/yr2009/web/index.html

## One-minute responses

- The phylogeny info was mostly review for me, though I definitely found it helpful.
- Given the mathematical limitations with searching for trees, will we ever be able to do anything interesting with trees in Python?
- I liked the explanation of phylogenies, especially list of tips/topologies–it was easy to understand why some approaches aren't feasible. the exercise on trees and scoring was helpful too.
- I was really interested in the phylogeny section of today. (x3)
- Today's class was interesting. I like how the relevance of programming to analyzing sequence data was mentioned.
- I really enjoyed the phylogeny portion of the talk today. The programming portion was very challenging, but the review was very helpful.

- It will be interesting to see how we use the Python tools we have to address bioinformatics issues.

## One-minute responses

- I felt somewhat more comfortable with looping last time; today's problems made me more confused. I think I just needed more time to work through the problems and will do so on my own.

- It would be helpful to have more time for sample problems. (x2)

- I'm still a bit (or **more** than a bit) confused on `for`... but, I think I'm slowly getting there....

- Good class content today and I appreciated taking the time to go back over `for` loops. A little hard in the programming section, but made sense over time.

- Problems were challenging. (x3)

- I appreciate the ample time to work on problems. Also liked the use of FASTA data and biological reference of practice problems.

- Very confusing class–mostly the Python part!

## One-minute responses

- I don't know what "+=" or "!=" means. *"x += 2" means "x = x + 2". "x != 2" means "x is not equal to 2" and is used in if statement tests.*

- It would be nice to have had the for loop summary in the lecture notes. (x3) *I printed last year's notes, not this year's! Oops.*

- Could we still get the summary slides of code that we've gone over each day? *Yes.*

- It would be helpful in the in-class problems to have the code input and output–just so it's clear what's expected. *I'll try to do this.*

- Sometimes it was hard to hear due to printer, people talking etc. *Please tell me when you can't hear me!*

# More on loops

- There are many ways to write a loop
- Some of them don't work very well!
- We'll look at a simple case in detail

# Print the first ten lines from a file

- We'll look at several ways to do this
- Points to consider:
  - What if the file is huge?
  - What if it has less than 10 lines?

# Ready the file for reading

I assume the file name is read from the command line

```
#open file for reading
import sys
filename = sys.argv[1]
filehandle = open(filename,"r")
```

I'll assume each program starts with this boilerplate.

# For loop

```
#read entire file
linelist = filehandle.readlines()

#print first 10 lines
count = 0
for myline in linelist :
  if count < 10 :
    print myline
  count += 1
filehandle.close()
```

# For loop

```
#read entire file
linelist = filehandle.readlines()

#print first 10 lines
count = 0
for myline in linelist :
  if count < 10 :
    print myline
  count += 1
filehandle.close()
```

This is not ideal:

- It reads the whole file unnecessarily
- It loops many more times than necessary
- It does handle a short file correctly

# Different for loop

```
#read entire file
linelist = filehandle.readlines()

#print first 10 lines
for index in range(0,10) :
  print linelist[index]
filehandle.close()
```

# For loop

```
#read entire file
linelist = filehandle.readlines()

#print first 10 lines
for index in range(0,10) :
  print linelist[index]
filehandle.close()
```

This is different but still flawed

- It still reads the whole file unnecessarily
- It loops only 10 times
- It ends with an error if there are less than 10 lines

# For loop

```
#read and print first 10 lines
for counter in range(0,10) :
  myline = filehandle.readline()
  print myline
filehandle.close()
```

# For loop

```
#read and print first 10 lines
for counter in range(0,10) :
  myline = filehandle.readline()
  print myline
filehandle.close()
```

This is much better:

- It reads only 10 lines
- It loops only 10 times
- Readline() returns the empty string if there are no more lines, so this program works correctly for very short files

# While loop

```
#read and print first 10 lines
count = 0
while count < 10 :
  myline = filehandle.readline()
  print myline
  count += 1
filehandle.close()
```

## While loop

```
#read and print first 10 lines
count = 0
while count < 10 :
  myline = filehandle.readline()
  print myline
  count += 1
filehandle.close()
```

This is good too:

- It reads only 10 lines
- It loops only 10 times
- Readline() returns the empty string if there are no more lines, so this program works correctly for very short files
- It's harder to write than the `for` loop

## Too many possibilities?!

- You don't need to be comfortable with all of them
- If you have a collection of something, consider `for` first
- If you don't have a collection, consider `for` with a `range`
- If neither `for` makes sense, try `while`

## The result of readlines() is a list of strings

- The first entry is the first line
- The second entry is the second line
- What does this do?
- `mylines = filehandle.readlines()`
- `target = mylines.find(">")`

## The result of readlines() is a list of strings

- The first entry is the first line
- The second entry is the second line
- What does this do?
- `mylines = filehandle.readlines()`
- `target = mylines.find(">")`
- `AttributeError: 'list' object has no attribute 'find'`

## The result of readline is a string

- The first entry is the first character
- The second entry is the second character
- What does this do?
- `myline = filehandle.readline()`
- `target = myline.find(">")`

# Practice problem 1

- Read the first 5 lines from a file
- Print characters 7-12 of each line
- Test this with the file short.txt from last lecture
- Make sure to handle lines with less than 12 characters safely

## loop.py

```
import sys
filename = sys.argv[1]
filehandle = open(filename,"r")

# read first 5 lines
for index in range(0,5) :
  line = filehandle.readline()
  # we don't count the end-of-line character
  numchars = len(line) - 1
  if numchars > 12 :
    print line[6:12]
  else :
    print line[6:numchars]
```

## loop.py results

python loop.py small.txt
THEPA
LFNILC
NVVIWE
VFSLNM
LINVFS

# Iterating over two containers with zip

```
seq1 = "ATGGCGA"
seq2 = "AAGGCGT"
count = 0
for (b1, b2) in zip(seq1, seq2) :
  if b1 != b2 :
    count += 1
print count
```

# Summary of commands (this lecture and last)

| Example | Meaning |
|---|---|
| for <item> in <container>: | iterate over container |
| while <condition>: | iterate until condition is met |
| x += 5 | x = x + 5 |
| x != 5 | x not equal to 5 |
| range(0,10) | create a list of 0, 1, 2, ... 9 |
| zip(<container1>,<container2>) | create a list of tuples |