

Genome Sciences 373

Genome Informatics

Quiz Section #2

April 5, 2016

Today

- Local alignment review
- Debugging strategies
- Python: data types review, input/output, if/else, for loops

Reminder: Office hours Mondays
4:30-5:30, Foegen S-040

Additional background question: stats and probability background?

- Can you define a p-value right now?
- Can you define null and alternative hypotheses right now?
- Are you familiar with the multiple hypothesis testing problem?

Local alignment DP

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

- Align sequence x and y .
- F is the DP matrix; s is the substitution matrix; d is the linear gap penalty.

$$F(0,0) = 0$$

Score of a match (or mismatch)

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

Adding a gap to one of the sequences

(corresponds to start of alignment)

Local alignment example

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

$$F(0,0) = 0$$

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

(corresponds to start of alignment)

		G	A	G	T	A
	0	?				
A						
G						
T						
T						
A						

Linear gap
penalty d
 $= -4$

Local alignment example

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

$$F(0,0) = 0$$

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

(corresponds to start of alignment)

		G	A	G	T	A
	0	0	0	0	0	0
A	0	?				
G	0					
T	0					
T	0					
A	0					

Linear gap
penalty d
 $= -4$

Local alignment example

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

(corresponds to start of alignment)

		G	A	G	T	A
	0	0	0	0	0	0
A	0	0				
G	0	?				
T	0					
T	0					
A	0					

Linear gap
penalty d
 $= -4$

Local alignment example

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

$$F(0,0) = 0$$

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

(corresponds to start of alignment)

		G	A	G	T	A
	0	0	0	0	0	0
A	0	0				
G	0	10				
T	0	6				
T	0	2				
A	0	?				

Linear gap
penalty d
 $= -4$

Local alignment example

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) + d \\ F(i, j-1) + d \\ 0 \end{cases}$$

(corresponds to start of alignment)

		G	A	G	T	A
	0	0	0	0	0	0
A	0	0	10	6	2	10
G	0	10	6	20	16	12
T	0	6	6	16	30	26
T	0	2	2	12		
A	0	0	12	8	22	36

Maximum score

36

Local alignment example

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

GAGT-A
AGTTA

		G	A	G	T	A
	0	0	0	0	0	0
A	0	0	10	6	2	10
G	0	10	6	20	16	12
T	0	6	6	16	30	26
T	0	2	2	12	26	25
A	0	0	12	8	22	36

When local and global alignments differ

We want to align two sequences:

AGTTA

AGAGTATTA

Optimal Local Alignment: -4

AGAG-TATTA
AGTTA

Optimal global alignment

4 gaps: -16

AGAGTATTA
AG-----TTA

Suboptimal global
alignment with 6 gaps: -24

AGAG-TATTA
--AGTTA---

So many deletions/insertions
might be implausible

Reminder: the alignment parameters are a model of mutational processes

Substitution matrix:

	A	C	G	T
A	10	-5	0	-5
C	-5	10	-5	0
G	0	-5	10	-5
T	-5	0	-5	10

AGAG-TATTA
AGTTA

- What kinds of mutations?
- How do we know whether to trust the mutations described by an alignment?

Debugging your code

Photo # NH 96566-KN (Color) First Computer "Bug", 1947

9/2

9/9

0800 Andam started

1000 " stopped - andam ✓

1300 (032) MP-MC { 1.2700 9.037 847 025
 2.130476415 } 9.037 846 995 correct


(033) PRO 2 2.130476415
 correct 2.130476415

Relays 6-2 in 033 failed special speed test
 in relay " 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.

1630 Andam started.

1700 closed down.

Relay 2145
 Relay 3376

Big picture tips to keep in mind

- Start small! Work in pieces!
- Regular print statements to check your progress
- Read error messages...
- Use toy examples to check your work
- Be patient!
- When the above fails, Google/Stack Overflow *may* be helpful...

Python error types

- `ParseError` = Syntax errors - look for formatting problems
- `TypeError`, `ValueError` = check your data types
- `NameError` = check your variable names
- `IndexError` = check your list indices

Debugging workflows

- The classic: text editor + terminal
- Run chunks in a Jupyter notebook
- IDEs: PyCharm, Spyder

Here's a broken program

- Copy and paste it, save as startCodon.py
 - What's it doing?

```
x = 'atggataccagg'  
print "x is", x
```

```
start_codon = 'atg'  
if start_codon = x[0:3]:  
    print 'Yes!'  
else:  
    print 'No!'
```

Here's a broken program

```
x = 'atggataccagg'  
print "x is", x
```

```
start_codon = 'atg'  
if start_codon == x[0:3]:  
    print 'Yes! '  
else:  
    print 'No! '
```

There's another problem!

```
x = 'atggataccagg'
print x
print "first 3 characters in x are",
x[0:3]

start_codon = 'atg '
print "start_codon is ", start_codon

if start_codon == x[0:3]:
    print 'Yes!'
else:
    print 'No!'
```

There's another problem!

```
x = 'atggataccagg'
print x
print "first 3 characters in x are",
x[0:3]

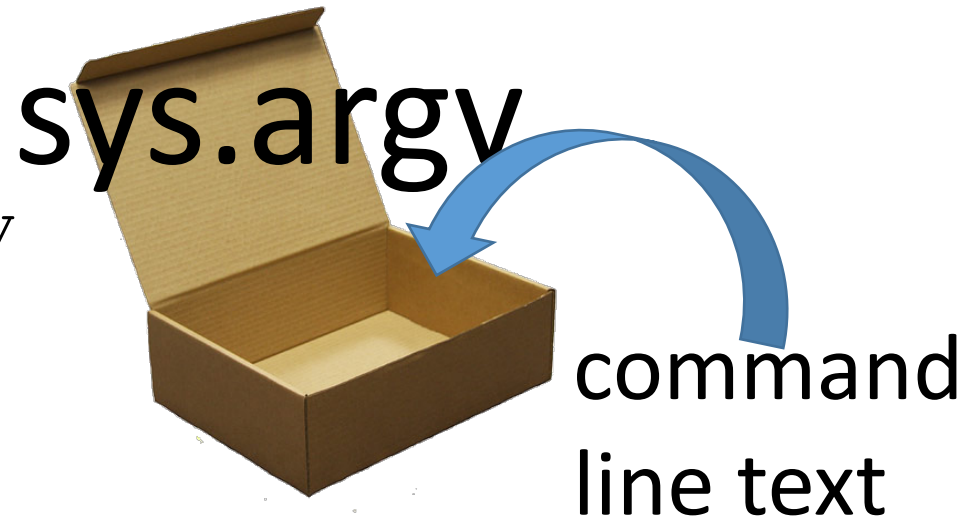
start_codon = 'atg'
print "start_codon is ", start_codon

if start_codon == x[0:3]:
    print 'Yes! '
else:
    print 'No! '
```

Let's learn more Python!

Taking input from the terminal command line

```
# This is inputs.py
import sys
print sys.argv
print sys.argv[1]
```



```
python inputs.py apple banana
```

```
['inputs.py', 'apple', 'banana']
'apple'
```

`sys.argv` only contains strings, so to get numbers, you need to **change the variable's type**

`python inputs.py 1.1 2`

```
import sys
print sys.argv
print sys.argv[1]
print type(sys.argv[1])
print float(sys.argv[1])
print int(sys.argv[2])
print int(sys.argv[1])
```

`sys.argv` only contains strings, so to get numbers, you need to **change the variable's type**

`python inputs.py 1.1 2`

```
import sys
print sys.argv
print sys.argv[1]
print type(sys.argv[1])
print float(sys.argv[1])
print int(sys.argv[2])
print int(float(sys.argv[1]))
```


One more example

python minus.py 3.3 1.1

```
import sys
print sys.argv
print sys.argv[1] - sys.argv[2]
```

```
['minus.py', '3.3', '1.1']
```

```
TypeError: unsupported operand
type(s) for -: 'str' and 'str'
```

One more example

```
python minus.py 3.3 1.1
```

```
import sys
print sys.argv
print float(sys.argv[1]) -
      float(sys.argv[2])
```

```
['minus.py', '3.3', '1.1']
2.2
```

Review: If/else statements

```
x = 4
```

```
if x == 5:  
    print 'x is 5!'  
elif x == 6:  
    print 'x is 6!'  
else:  
    print 'x is neither 5 nor  
6!'
```

Review: If/else statements

x = 5 Note: Indents are important!!

y = 7

```
if x == 5:
    print 'x is 5!'
    if y == 7:
        print 'x is 5 and y is 7!'
    else:
        print 'x is 5 and y is not 7!'
else:
    print 'x is not 5!'
```

Review: If/else statements

```
x = 5
```

```
y = 7
```

Note: Indents are important!!

```
if x == 5:
```

```
    print 'x is 5!'
```

```
    if y == 7:
```

```
        print 'x is 5 and y is 7!'
```

```
    else:
```

```
        print 'x is 5 and y is not 7!'
```

```
else: A block inside a block (nested)
```

```
    print 'x is not 5!'
```

A block

Example

```
x = [ 1, 3, 2]
```

From a list x containing 3 numbers, print the number with the smallest value

```
smallest_value = x[0]
```

```
print smallest_value
```

Example

`x = [1, 3, 2]`

From a list `x` containing 3 numbers, print the number with the smallest value

```
smallest_value = x[0]
if x[1] < smallest_value:
    smallest_value = x[1]
if x[2] < smallest_value:
    smallest_value = x[2]
print smallest_value
```

For loops let you repeat the same commands for each element in a list

```
x = [1, 2, 3]
for i in x:
    print i
print 'done!'
```

1
2
3
done!

i takes on the value of each element in the list for each iteration of the code inside the for loop block

For loops also work for strings!

```
x = 'actg'  
for i in x:  
    print i  
print 'done!'
```

```
a  
c  
t  
g  
done!
```

Compute the sum of the numbers in x!

```
x = [1, 2, 4, 5]
```

```
sum = 0
```

```
for v in x:
```

```
    sum = sum + v
```

```
print 'The sum is:', sum
```

```
The sum is: 12
```

How about the product?

```
x = [1, 2, 4, 5]
```

```
print 'The product is:', product
```

```
The product is: 40
```

How about the product?

```
x = [1, 2, 4, 5]
```

```
product = 1
for v in x:
    product = product * v
print 'The product is:', product
```

The product is: 40

Powerful strategy: Combining for loops and if/else statements

```
x = [12, 3, 4.4, 6]
```

Output how many numbers in the list x with values greater than 5

```
count = 0
for v in x:
    if v > 5:
        count = count + 1
print count
```

Watch the indents!!

Example: finding the sum using command line arguments

- Write a program to find the sum all numbers given as input
 - `python sum.py 1.0 11 15 12.692`

```
import sys
```

```
SUM = 0
```

```
print SUM
```

Example: finding the sum using command line arguments

- Write a program to find the sum all numbers given as input
 - `python sum.py 1.0 11 15 12.692`

```
import sys
SUM = 0
for v in sys.argv[1:]:
    SUM += float(v)
#same as SUM = SUM + float(v)
print SUM
```

Programming keys to remember

- Variables: objects that hold data
 - numbers, strings, lists, etc.
- Operators: take objects, do something, and give you something else
 - and, or, +, =
- If/else statements: control the flow of the program based on Boolean objects derived from the data
- For loops repeat the same code for each value in a list

