Unix Tutorial for Biologists

Scripting
Nathan Dunn
CASSPR (ndunn@cas.uoregon.edu)
Overview

• Emails -> Survey
• Scripting Languages: bash, python, perl
• Simple Tasks: for loop, read in / read out, conditional
• Libraries
• Installing Software
• Mercurial Source Control
Remember Bash

- Good at chaining tools
- Tools faster than scripts
- Bad at complicated
Perl

- First big bio scripting language
- Lots of libraries
- Not bad to write, horrible to read
Python

• Object-oriented scripting language
• Easy, readable, lots of doc
• Whitespace!
Ruby

• Similar to python
Groovy

• A Java scripting language
• In Java VM, lots of libraries
• Integrating native code more difficult
Simple Tasks

- loop
  - for each element in list <statements>
- conditional
  - if <expression> then <statements> else <statements>
- read in / read out from file
Bash

- **loop**
  
  ```bash
  for i in <list>
    do
      <statement>
    done
  ```

- **conditional**
  
  ```bash
  if [ "$1" == "sometext" ]; then
      <statement>
  else
      <statement>
  fi
  ```

- **read in**
  
  ```bash
  cat filename1.txt | grep pattern
  ```

- **write out**
  
  ```bash
  cat filename1.txt > filename2.txt
  ```
Perl

- **loop**
  ```perl
  foreach(@list){
    <statement on $_[ ]>
  }
  ```

- **conditional**
  ```perl
  if(var1==var2){
    <statement>
  }
  ```

- **read in**
  ```perl
  open(MYINPUTFILE, "<input_file.txt");
  while(<MYINPUTFILE>){
    <statement on $_[ ]>
  }
  ```

- **write out**
  ```perl
  open(MYOUTFILE, ">output_file.txt"):
  print MYOUTFILE @list;
  close(MYOUTFILE) ;
  ```

Tuesday, May 29, 12
Python

• **loop**
  ```python
  for i in <list>:
      <statement>
  ```

• **conditional**
  ```python
  if var1==var2:
      <statement>
  ```

• **read in**
  ```python
  pwd = os.path.abspath(os.path.dirname(__file__))
  input_file = open(pwd+'/inputfile.txt','r')
  # read input_file
  input_file.close()
  ```

• **write out**
  ```python
  output_file = open(pwd+'/outputfile.txt','w')
  output_file.write(stringsofstuff)
  output_file.close()
  ```
GPL Example Bash


gunzip GPL4014_family.soft.gz

grep -v ^! GPL4014_family.soft | grep -v \# | grep ENSDARG00000 | cut -f2 | sort -u | grep "^B\|^NM" > processed_GPL.txt ;
GPL Example Perl

• Does gunzip automatically

```perl
open(out, ">:utf8", "GPL4014_family.soft");
print out $data ;
```
GPL Example Perl

while(<MYINPUTFILE>){
    my($line) = $ _ ;  # $ _ is the next value
    chomp($line);  # strips the trailing newline
    if($line =~ /^[A-Z].*.*ENSDARG00000/){
        # split . . . only need second variable
        ($garbage,$accession) = split(\"\t\",\"$line\") ;
        if($accession =~ /^[B|^NM]/){  # push it to an array
            push(@accessions,$accession.\"\n") ;
        }
    }
}

urllib.urlretrieve(url,"GPL4014_family.soft.gz")

pwd = os.path.abspath(os.path.dirname(__file__))

gzip_file = gzip.open(pwd+'/GPL4014_family.soft.gz', 'rb')

file_content = gzip_file.read()

decompressed_file = open(pwd+'/GPL4014_family.soft','w')

decompressed_file.write(file_content)
GPL Example Python

```python
input_file = open(pwd+'/GPL4014_family.soft','r')
output_file = open(pwd+'/processed_GPL.txt','w')
output_list = []
for line in input_file:
    if (False==line.startswith('!') or False==line.startswith('#')) and line.find("ENSDARG00000") > 0:
        parsed_line = line.split("\t")
        if len(parsed_line) >5:
            accession = parsed_line[1]
            if (accession.startswith("B") or accession.startswith("NM")) and accession not in output_list:
                output_list.append(accession)
output_list.sort()
for item in output_list:
    output_file.write(item+"\n")
```
Barcode Reader Python

• For each barcode X dump all fastq sequences to a file named X.fastq

@HWI-ST0747:210:D05GJACXX:3:2308:19482:126261 1:Y:0:
CAACT
TCCGCGCCCCCAGCGCCCGGAACGCGTACTCCTGTA
+
=1:(DB:D@)0))2::?3CD:)0::;@A<5–9?D@(;

CAACT.fastq

@HWI-ST0747:210:D05GJACXX:3:2308:19482:126261 1:Y:0:
TCCGCGCCCCCAGCGCCCGGAACGCGTACTCCTGTA
+
B:D@)0))2::?3CD:)0::;@A<5–9?D@(;

Tuesday, May 29, 12
from Fastq import *

class Fastq:
    def __init__(self):
        self.comment = ""
        self.sequence = ""
        self.barcode = ""
        self.comment2 = ""
        self.score = ""

    def isEmpty(self):
        return self.comment == "" and self.sequence == "" and self.comment2 == "" and self.score == ""
Barcode Reader Python

- Use a “dictionary” or “map”
- `<barcode>,<list of sequences>`
- if not found add, otherwise append

```python
barcode_file = {}

def handleBarcode(obj,mydict):
    retrieved_obj = barcode_file.get(fastq_object.barcode)
    if retrieved_obj == None:
        barcode_file[fastq_object.barcode] = [fastq_object]
    else:
        retrieved_obj.append(fastq_object)
    barcode_file[fastq_object.barcode] = retrieved_obj
```
for line in input_file:
    if count==0:
        if False==fastq_object.isEmpty():
            handleBarcode(fastq_object, barcode_file)
        fastq_object = Fastq()
        fastq_object.comment = line
        count = 1
    elif count==1:
        fastq_object.sequence = line[5:]
        fastq_object.barcode = line[:5]
        count = 2
    elif count==2:
        fastq_object.comment2 = line
        count = 3
    elif count==3:
        fastq_object.score = line
        count = 0
Barcode Reader

BioPython

• Library already written

```python
from Bio import SeqIO

original_reads = SeqIO.parse("illumina.fastq", "fastq")
for read in original_reads:
    primer = str(read.seq[:primer_length])
    trimmed_read = read[primer_length:]
    handleBarcode(trimmed_read, primer, barcode_file)

for i in barcode_file.iterkeys():
    output_file = open(pwd+'/output/'+str(i)+'.fastq','w')
    for sequence in barcode_file.get(i):
        output_file.write(sequence.format("fastq"))
```

Tuesday, May 29, 12
Install BioPython

- Install:
  - http://biopython.org/DIST/docs/tutorial/Tutorial.html
  - http://biopython.org/wiki/Download
  - python setup.py build
  - sudo python setup.py install
Libraries

- Python: BioPython, egg
- Perl: BioPerl, CPAN
- Ruby: BioRuby, gems
- Groovy: BioJava/ BioGroovy, jars / maven
Installing Software

• sudo <command>

  • make root for one command

• mac: “brew”

• ubuntu: “sudo apt-get”

• redhat: “sudo yum”
Mercurial Source Control

• install mercurial (probably already installed)
• hg init  # in directory
• hg add <files>
• hg diff
• hg commit
Performance

• Things to remember:
  • Code is a factory
  • Trace / Time
  • Google!
  • Ask!
Next Week

• ACISS?
• More scripting languages?
• More libraries?
• More problems?
• In-class problems!