Using BioPython to analyze sequence alignments

Kelsey E. Witt, School of Integrative Biology (kewitt2@illinois.edu)

Introduction: BioPython

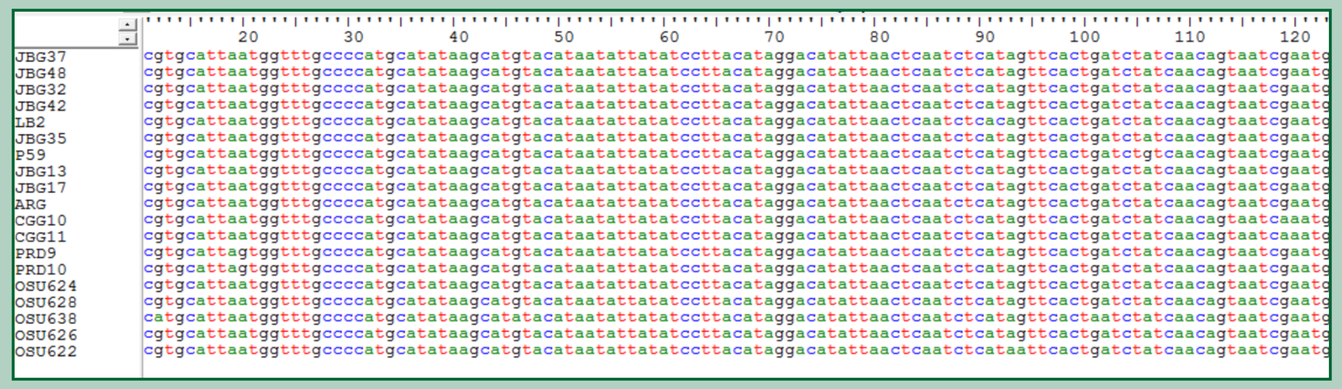
- BioPython is a freely available Python package designed to assist with bioinformatics research
- Mainly, the package parses sequence alignments to make them easy to manipulate using Python
- Other uses include file format conversion, sequence comparison, as well as running of third -party programs (including MUSCLE, blast, etc.)

Aims and Process

- Goal: Learn to use BioPython to manipulate sequence data
- Worked through provided BioPython tutorial
- Wrote a program to compare sequences in an alignment to a reference file and output the results into a .csv file

Materials

- Alignment files of dog mitochondrial genome (mitogenome) sequences
- Python version 2.7—accessed with Pycharm
- BioPython version 1.66



	20	30	40	50	60	70	80	90	100	110	120
JBG37	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
JBG48	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
JBG32	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
JBG42	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
LB2	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcacagttcactgatctatcaacagtaatcgaatg									
JBG35	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
P59	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctgtcaacagtaatcgaatg									
JBG13	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
JBG17	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
ARG	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
CGG10	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcaaatg									
CGG11	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcaaatg									
PRD9	cgtgcattagtggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
PRD10	cgtgcattagtggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
OSU624	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
OSU628	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
OSU638	catgcattaatggtttgccccatgcatataagcatatacataatattatatcctt	acataggacatattaactcaatctcatagttcactaatctatcaacagtaatcgaatg									
OSU626	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcatagttcactgatctatcaacagtaatcgaatg									
OSU622	cgtgcattaatggtttgccccatgcatataagcatgtacataatattatatcctt	acataggacatattaactcaatctcataattcactgatctatcaacagtaatcgaatg									

	100	105	118	123	131	181	185	216	258	267	271	273	279	335	347	397	400	404
JBG37																		
JBG48																		
JBG32																		
JBG42																		
LB2																		
JBG35																		
P59		g																
JBG13												t	t					
JBG17												t	t					
ARG					c													
CGG10			a															
CGG11			a															
PRD9																-		
PRD10																		
OSU624																	r	
OSU628																		
OSU638	a								a	c	a							
OSU626		y																r
OSU622						a	a	t					r	m				

Figure 1: A sample sequence alignment file

Figure 2: The output of the python script. Each column is a base position that differs from the reference, and the base pairs in the table indicate what the different base pair is

Source Code

```
from Bio import SeqIO
from Bio.Alphabet import IUPAC
dog_ref = SeqIO.read("ref_aln.fas", "fasta", IUPAC.ambiguous_dna) #initializes dog reference
from Bio import AlignIO
alignment = AlignIO.read("dogs_full.fas", "fasta") #creates alignment as a file
SNPs = {} #creates dictionary called SNPs
dog_names = [] #creates list called dog_names
for record in alignment: #go line by line through the alignment
  SNPs[record.id] = {}#Creates a new dictionary for each sample
  dog_names.append(record.id) #add dog name to list of dog names
  for position in range(0, len(dog_ref)): #As you read through each nucleotide
    if dog_ref.seq[position] != record.seq[position] and record.seq[position]!= "n": #if this position differs from the reference
      SNPs[record.id][str(position)] = record.seq[position] #make an entry in the dictionary with the key as the sample name
print SNPs #prints dictionary
all_dog_keys = set() #creates a set for the keys
for subdct in SNPs.values(): #for all nested dictionaries within SNPs
  all_dog_keys |= set(subdct) #combine all keys
all_dog_keys = reduce(set.union, map(set, SNPs.values())) #makes a single set containing all keys
dog_keys = list(all_dog_keys) #converts to list
dog_keys.sort()
num_dogs = len(dog_names)-1

print dog_names
print dog_keys

import csv
output_file='./SNPs.csv'
with open('SNPs.csv', 'wb') as csvfile:
  w = csv.writer(csvfile, delimiter=',')
  w.writerow(['']+ dog_keys) #creates header with base pairs represented
  for dog in range(0, len(dog_names)):
    file_row = [] #creates a list that can be used by the file
    file_row.append(dog_names[dog]) #first column is sample name
    for SNP in range(0,len(dog_keys)): #iterate through SNPs
      file_row.append(SNPs[dog_names[dog]].get(dog_keys[SNP], "")) #add on either the different base pair or nothing if there's no key in
that dictionary
    w.writerow(file_row) #write the whole row to the csv file
```

Acknowledgments

- Thanks to the creators of BioPython, who have made this freely available
- This work was part of a Focal Point grant funded by the Graduate College at the University of Illinois at Urbana-Champaign