Answers for challenges from Lecture 21

Because this set of challenges requires a mix of `Unix` and `Python/R` commands, I'll provide them in a pdf, rather than script format. If not specified assume the code is for `Unix`. Questions from the *Practice with basic bioinformatics* handout provided in Lecture 21 are below in *italics* and code is presented below the questions. Remember that paths to files, Muscle, hmmbuild, and hmmsearch will be different on your computer. Also, the Muscle binary usually has some version information following the word "muscle" that needs to be appended when using it.

All files required for the challenges are available on Sakai in `Resources/Week11/Lecture21files.tar.gz`. Because some of these challenges and tasks build upon each other some file names will be carried through multiple answers.

Based on the Muscle help page, what is the basic syntax for generating a sequence alignment? What is the default format for an input sequence file?

```
./muscle -in <a fasta file with sequences to be aligned> -out <a file name for the aligned sequences, in fasta format, to be saved>
```

Based on the hmmbuild and hmmsearch help pages, what is the basic syntax for generating a profile HMM and searching a database with a profile HMM?

```
./hmmbuild <a file name to store a profile HMM in> <the file containing a fasta formatted sequence alignment>
```

```
./hmmsearch <a file containing a profile HMM> <a fasta file containing sequences to be searched using the HMM>
```

If we want to get hmmsearch output in a tabular formatted file we add the `--tblout` option:

```
./hmmsearch --tblout <file name to store output> <a file containing a profile HMM> <a fasta file containing sequences to be searched using the HMM>
```

What is the E-value or bit score that indicates a "true" match?

There is not a good answer for this. The E-value is context (database size, alignment length) dependent. Closer to zero is a better match, but sometimes we are actually looking for "worse" matches. If we are hoping to find distantly related organisms or proteins we may not expect a very good match to our query sequence or HMM.

How could we reduce tabular output to the 1st, 4th, and 8th column of a tabular text file returned by `hmmsearch`?

If the tabular output was called `hmm.hits`, either of the following would work:

```
cat hmm.hits | grep -v "#" | sed -E 's/ +/ /g' | cut -d " " -f 1,4,8
```

```
cat hmm.hits | grep -v "#" | awk '{print $1,$4,$8}'
```

How could we find information contained in one table that is associated with sequence identifiers that were identified as sequences of interest by `hmmsearch`?

This question might be a bit unclear, but imagine we used hmmsearch to find some likely protein sequence matches to our profile HMM for sigma factors and we also had previously annotated (defined a putative

function) all the proteins in a bacterium's genome. We could cross-check our matches from hmmsearch with the existing annotations. One can do this with `R` or `Python` and it provides practice for working with these kinds of data in a scripting language.

In Python:

```
import pandas

# load the annotations, which has a sequence ID in the first column and a function in the 2nd
annot=pandas.read_csv("Roseobacter.annot",sep="\t",header=None)

# define a sequence ID we want to find, this would come from the first columon of hmmsearch results
searchstring='tr|B7RK13|B7RK13_9RHOB'

# find the row of the annotations that matches the search string
annot.loc[annot.iloc[:,0]==searchstring,:]
```

In R:

```
#This is actually harder in R because there are some special characters that cause problems
#load the annotations information
# read in the annotation file as simple text
annot=scan("Roseobacter.annot",sep = "\n",what=character())

# put the simple text into a dataframe
tempList=strsplit(annot,split="\t")
tempVec=unlist(tempList)
annot=data.frame(seqID=tempVec[seq(1,length(z)-1,2)],annotation=tempVec[seq(2,length(z),2)])

# define a sequence ID we want to find, this would come from the first columon of hmmsearch results
searchstring='tr|B7RK13|B7RK13_9RHOB'

# find the row of the annotations that matches the search string
annot[annot[,1]==searchstring,]
```

Challenges

1) *Align the sequences in sigma70.ref using muscle. How does the alignment file differ from the original file?*

```
./muscle -in sigma70.refs -out sigma70.align
```

The alignment file should have some '-' characters indicating gaps.

2) *Build a profile HMM using the sigma70 sequence alignment and hmmbuild.*

```
./hmmbuild sigma70.hmm sigma70.align
```

3) *Search the Roseobacter proteome (`Roseobacter.fasta`) for sigma factors using hmmsearch. How many matches do you get?*

```
./hmmsearch --tblout sigma70.hits sigma70.hmm Roseobacter.fasta
```

```
cat sigma70.hits | grep -v "#" | wc -l
```

I got 7 matches in the Roseobacter proteome.

4) *Make a histogram of the bit scores for all sigma factor hits from the Roseobacter proteome.*

```
cat sigma70.hits | grep -v "#" | awk '{print $1,$3,$6}' > sigma70.table
```

In Python:

```
import pandas
from plotnine import *

data=pandas.read_csv("sigma70.table",sep=" ",header=None,names=["seqID","hmm","score"])
ggplot(data,aes("score"))+geom_histogram()
```

In R:

```
library(ggplot2)

data=read.table("sigma70.table",sep=" ",header=FALSE,stringsAsFactors=FALSE)
colnames(data)=c("seqID","hmm","score")
ggplot(data,aes(score))+geom_histogram()
```

5) *Use the annotations of genes from the Roseobacter proteome (`Roseobacter.annot`) to check whether your sigma factor hits seem like good matches.*

In Python:

```
import pandas

# load the annotations, which has a sequence ID in the first column and a function in the 2nd
annot=pandas.read_csv("Roseobacter.annot",sep="\t",header=None)

# define a sequence ID we want to find, this would come from the first columon of hmmsearch results
hits=pandas.read_csv("sigma70.table",sep=" ",header=None,names=["seqID","hmm","score"])

# find the row of the annotations that matches the search string
for i in range(0,len(hits)):
  annot.loc[annot.iloc[:,0]==hits.seqID[i],:]
```

In R:

```
#This is actually harder in R because there are some special characters that cause problems
#load the annotations information
# read in the annotation file as simple text
annot=scan("Roseobacter.annot",sep = "\n",what=character())

# put the simple text into a dataframe
tempList=strsplit(annot,split="\t")
tempVec=unlist(tempList)
annot=data.frame(seqID=tempVec[seq(1,length(z)-1,2)],annotation=tempVec[seq(2,length(z),2)])

# define a sequence ID we want to find, this would come from the first columon of hmmsearch results
hits=read.table("sigma70.table",sep=" ",header=FALSE,stringsAsFactors=FALSE)
colnames(hits)=c("seqID","hmm","score")

# find the row of the annotations that matches the search string
for(i in 1:nrow(hits)){
```

```
  print(annot[annot$seqID==hits$seqID[i],])
}
```

It looks like are matches from `hmmsearch` are pretty good. The annotations all point to sigma factors.

6) *How many proteins are encoded in each of the eight proteomes provided?*

```
for file in *.fasta
do
  echo $file
  cat $file | grep ">" | wc -l
done
```