

#Miscellaneous Scripts To Post--Clean 10-22-23.txt

#####

#1) FastaStepThroughToFiles_Perl
#FROM: fastaby2_SepFiles.pl

#!/usr/local/bin/perl

#this perl script reads in a fasta protein file,
#finds ID lines,
#and steps through by 2
#prints to separate text files by group

use strict;
use warnings;

#read in file
my \$filename = 'Data/small_file.txt';
open (my \$fh, '<', \$filename)
 or die "Could not open file '\$filename' \$!";

#initialize counters***
my \$n = 0; #line number
my \$g = 1; #group number
my \$cg = \$g-1; #nonadvanced group number (for printing lines belonging
to group)
my \$id = 0; #i.d. line number

#checks if line starts with? '>', and if multiple of 2, prints if so
#uses variable for file names

```
while (<$fh>) {  
    chomp $_;  
    #print "$_\n";  
    if (/^>/) {  
        if ($id%2 == 0) {  
            #make new file for new group, print line to  
it  
            my $file_out = "group_$g.txt";  
            open (my $fo, '>>', $file_out) or die "Can't  
open output file: $!";  
            print $fo "Group number $g\n";  
            print $fo "$_\n";  
            $g++;  
            $cg++;  
            $id++;  
        } else {  
            #use nonadvanced group number ($cg)  
            my $file_out = "group_$cg.txt";
```

```

open output file: $!";
open (my $fo,'>>', $file_out) or die "Can't
#print current line to current group file
print $fo "$_\n";
$id++;
}
} else {
#use nonadvanced group number ($cg)
my $file_out = "group_$cg.txt";
open (my $fo,'>>', $file_out) or die "Can't open
output file: $!";
#print current line to current group file
print $fo "$_\n";
}

#print $fo "Group var = $g\n"; #give value of group # var
#print $fo "ID var = $id\n"; #give value of id # var

$n++; #increment line number
}

close($fh);
#close($fo);

```

```
#####
```

```

#2) FastQ Sym Link Maker for NextFlow GATK Pipeline
#FROM: python3_symlink_subdir_notebook.py

```

```

# %load /Users/michaeldhar/Documents/School/NYU-Poly/
DatepalmFilesBackup-7-17-16/Notes-Output/GATK Project/Python_Symlink/
python3_symlink_subdir.py
#!/usr/bin/env python3

```

```

#This script creates symlinks of FastQ forward and reverse files with
file names usable by NextFlow glob read function.

```

```

import os
import re
from re import search

```

```

#declare FULL path variable of samples
path = '/Users/michaeldhar/Documents/School/NYU-Poly/
DatepalmFilesBackup-7-17-16/Notes-Output/GATK Project/Python_Symlink/
samples'
#declare symlink directory path
symdir_path = os.path.join(path,"sym_links")
print(symdir_path)

```

```

#create directory for symlinks if doesn't already exist
if not os.path.isdir(symdir_path):
    print(symdir_path + " does not exist")
    #make dir
    os.mkdir(symdir_path)
else:
    print("Can't create directory " + path + ". Directory already
exists.")

#step through files
for root,directories,files in os.walk(path):
    for name in files:
        #extract fastq file base name
        #check if file is already NFsym link
        if not search("_NFsym_",name):
            #get base name, only proceed if .fastq:
            try:
                base = re.search('(.(
+?).fastq',name).group(1)
            except AttributeError:
                found = "FastQ extension not found"
            else:
                #declare original FASTQ path
                fastpath = os.path.join(root, name)
                #find if forward or reverse read and
                set symlink name and path; use search() to find any instance of R1/R2/
                read1/read2
                if search("read1|R1",name):
                    read = "read1"
                if search("read2|R2",name):
                    read = "read2"
                newname = base + "_NFsym_" + read +
                ".fastq"
                #check if symlink or file path
                exist, make symlink if not
                #declare newsym path var
                newsym =
                os.path.join(symdir_path,newname)
                if not os.path.islink(newsym):
                    if not
                    os.path.exists(newsym):
                        #print base and
                        original path and symlink path to screen
                #print(base,fastpath,newsym)
                #print base,
                original file name, sym file name, original path, and symlink path to
                file
                #file_links

```

```
= open("linklist.txt", "a") #open file in append mode
file_links
= open(os.path.join(symdir_path,"linklist.txt"), "a") #open file in
append mode

file_links.write(base + "\t" + name + "\t" + newname + "\t" + fastpath
+ "\t" + newsym + "\n")

file_links.close()
#make symlink

os.symlink(fastpath,newsym)
```