MCB 5472 Assignment #4: Introduction to command line BLAST February 12, 2014

Assignment feedback

- Everyone is doing very well!
- Most people lose marks because they have not read the question close enough (e.g., not handing in pseudocode for Assign. #2)
- Check that your output files match your input and that they contain what you think they should

Code hints:

- Filehandles should be in block capitols

```
open (INFILE, $ARGV[0]); # good
open (infile, $ARGV[0]); # less good
```

- Will work otherwise but not under "`use warnings`" and "`use strict`" pragmas, i.e., sloppy code

Code hints:

- Align you code blocks properly
- Purpose: so you can intuitively see your code logic

```
foreach $word (@array){
    if ($word =~ /fox/){
        print "found the fox";
    }
}
```

Code hints:

- When using pattern matching, you should always consider possible exceptions in your input file

```
$line =~ /^>/; # match fasta header
$line =~ />/; # matches any line with a ">"
character
$line =~ tr/ACGT/TGCA/; # compliments high
quality sequence
$line =~ tr/ACGTacgt/TGCAtgca/; # compliments
high and low quality sequence
$line =~ /^[ACGT]/; # line starts with a
nucleotide
$line !~ /^>/; # any line not a fasta header,
accommodates degenerate bases: N, V, B, H, D,
K, S, W, M, Y, R
```

Code hints:

- "`\s`" means "any white space character"
- Includes:

```
" " # space
"\t" # tab
"\r" # return
"\n" # new line
```

Code hints:

- Filehandles should be in block capitols

```
open (INFILE, $ARGV[0]); # good
open (infile, $ARGV[0]); # less good
```

- Will work otherwise but not under “`use warnings`” and “`use strict`” pragmas, i.e., sloppy code
- Keep your tabs aligned
- Matching /^>/ vs />/
- tr/ATCG/TAGC/ vs tr/ATCGatcg/TAGCtagc/
- “\s” matches: “ “, “\t”, “\r”, “\n”

Command line BLAST

- We will be using BLAST+
- Need to run on the Biotechnology Center server
  - Preinstalled on Biolinux so can be run locally
- Two parts to every BLAST
  1. Format the BLAST database
  2. Perform the BLAST itself

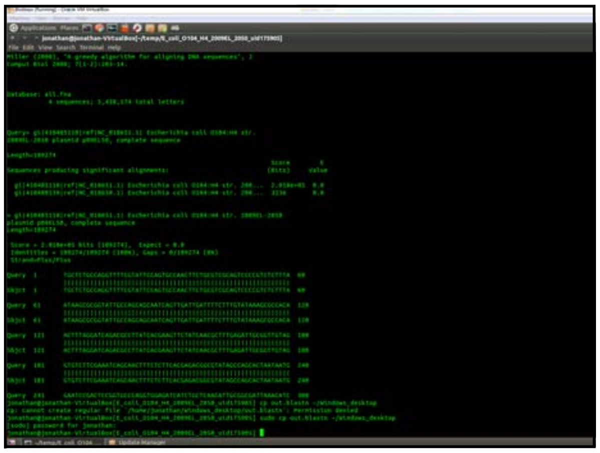
Formatting a BLAST database

- `[jlklassen@bbcsrv3 ~]$ makeblastdb -in [name of input file] -dbtype [either ‘nucl’ or ‘prot’]`
  - e.g., `[jlklassen@bbcsrv3 ~]$ makeblastdb -in all.fna -dbtype nucl`
- Produces:
  - nucleotide: `[name].nhr`, `[name].nin`, `[name].nsq`
  - protein: `[name].phr`, `[name].pin`, `[name].psq`

  where `[name]` is whatever was entered for the `makeblastdb -in` flag
- For help: `[jlklassen@bbcsrv3 ~]$ makeblastdb -help`

Running BLAST

- `[jlklassen@bbcsrv3 ~]$ blastn –query [query file name] –db [database name]`
  - e.g., `[jlklassen@bbcsrv3 ~]$ blastn –query NC_018651.fna –db all.fna`
- For other BLAST flavors: replace `blastn` with `blastp`, `blastx`, `tblastn` or `tblastx`
- For help: `[jlklassen@bbcsrv3 ~]$ blastn –help`
- Multiple fasta file can be used as query
  - multiple BLAST outputs in the same output file



```
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.



Database: all.fna
           4 sequences; 1,438,174 total letters



Query= gi|410485118|ref|NC_018651.1| Escherichia coli O104:H4 str.
2009EL-2050 plasmid p09EL50, complete sequence

Length=109274
                                                                      Score     E
Sequences producing significant alignments:                          (Bits)  Value

  gi|410485118|ref|NC_018651.1| Escherichia coli O104:H4 str. 200...  2.018e+05  0.0
  gi|410488139|ref|NC_018658.1| Escherichia coli O104:H4 str. 200...  3338       0.0


> gi|410485118|ref|NC_018651.1| Escherichia coli O104:H4 str. 2009EL-2050
plasmid p09EL50, complete sequence
Length=109274

 Score = 2.018e+05 bits (109274),  Expect = 0.0
 Identities = 109274/109274 (100%), Gaps = 0/109274 (0%)
 Strand=Plus/Plus

Query  1       [illegible]  60
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  1       [illegible]  60

Query  61      [illegible]  120
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  61      [illegible]  120

Query  121     [illegible]  180
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  121     [illegible]  180

Query  181     [illegible]  240
               ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  181     [illegible]  240

Query  241     [illegible]  300
jonathan@jonathan-VirtualBox[E_coli_O104_H4_2009EL_2050_uid175905] cp out.blastn ~/windows_desktop
cp: cannot create regular file `/home/jonathan/windows_desktop/out.blastn': Permission denied
jonathan@jonathan-VirtualBox[E_coli_O104_H4_2009EL_2050_uid175905] sudo cp out.blastn ~/windows_desktop
[sudo] password for jonathan:
jonathan@jonathan-VirtualBox[E_coli_O104_H4_2009EL_2050_uid175905]
```

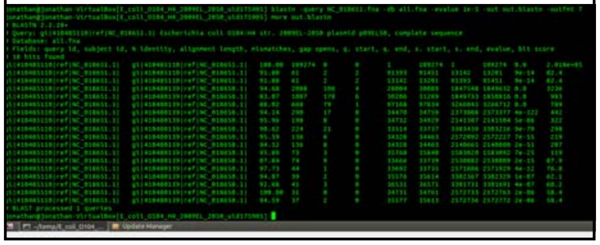
Other helpful BLAST options

- `–evalue [maximum evalue threshold]`
- `–out [output file name]`
- `–outfmt [0 for normal alignment format; 7 for easy to parse table format]`

Tabular output format

- Separated by tabs ("\t")

```
@blastline = split "\t", $line;
```



```
jonathan@jonathan-VirtualBox[E_coli_O104_H4_2009EL_2050_uid175905] blastn -query NC_018651.fna -db all.fna -evalue 1e-5 -out out.blastn -outfmt 7
jonathan@jonathan-VirtualBox[E_coli_O104_H4_2009EL_2050_uid175905] more out.blastn
# BLASTN 2.2.28+
# Query: gi|410485118|ref|NC_018651.1| Escherichia coli O104:H4 str. 2009EL-2050 plasmid pO91150, complete sequence
# Database: all.fna
# Fields: query id, subject id, % identity, alignment length, mismatches, gap opens, q. start, q. end, s. start, s. end, evalue, bit score
# 18 hits found
gi|410485118|ref|NC_018651.1|   gi|410485118|ref|NC_018651.1|   100.00  109274  0    0  1      109274  1        109274   0.0     2.018e+05
gi|410485118|ref|NC_018651.1|   gi|410485118|ref|NC_018651.1|   91.80   61      2    3  91393  91451   13142    13201    9e-14   82.4
gi|410485118|ref|NC_018651.1|   gi|410485118|ref|NC_018651.1|   91.80   61      2    3  13142  13201   91393    91451    9e-14   82.4
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   94.68   2088    106  4  28004  30089   1847548  1849632  0.0     3236
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   83.87   1087    170  6  30206  31289   1849733  1850816  0.0     983
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   88.92   668     79   1  97168  97834   3266061  3266712  0.0     789
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   94.14   290     17   0  34470  34759   2373088  2373377  4e-122  442
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   95.96   198     8    0  34732  34929   2141307  2141504  5e-86   322
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   90.62   224     21   0  33514  33737   3383439  3383216  9e-79   298
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   95.59   136     6    0  34328  34463   2572092  2572227  7e-55   219
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   94.12   136     8    0  34328  34463   2140061  2140000  2e-53   207
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   95.89   73      3    0  35768  35840   1583620  1583692  7e-25   119
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   87.84   74      9    0  33666  33739   2538882  2538809  2e-15   87.9
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   97.73   44      1    0  33692  33735   2571886  2571929  4e-12   76.8
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   94.87   39      2    0  35576  35614   3382307  3382329  1e-07   62.1
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   92.68   41      3    0  36531  36571   3381731  3381691  4e-07   60.2
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   100.00  31      0    0  34731  34761   2572733  2572763  2e-06   58.4
gi|410485118|ref|NC_018651.1|   gi|410480139|ref|NC_018658.1|   94.59   37      2    0  35577  35613   2572736  2572772  2e-06   58.4
# BLAST processed 1 queries
jonathan@jonathan-VirtualBox[E_coli_O104_H4_2009EL_2050_uid175905]
```

This week: Question #1

- Last week's complete genomes each had plasmids
- Are the plasmids from each organism with a complete genome homologous?
- Are the plasmids present in any of the draft genome sequences?

Practically:

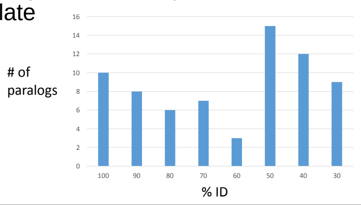
- Use BLASTn to compare plasmids with each other
- Use BLASTn to find homologous sequence to each plasmid type in the draft genomes
- Use your judgment to infer homologs – this is ultimately subjective and needs to be defended!
- YOU DO NOT NEED TO WRITE PERL SCRIPTS FOR THIS (unless you want to)

This week: Question #2

- Find paralogous genes and proteins in the complete *Escherichia coli* O104:H4 str. 2009EL-2050 genome and its plasmids
- Compare number of gene and protein paralogs
- Tabulate paralog age estimated from their percent BLAST similarly

Practically:

- Download the genes from NCBI
- BLAST all genes & proteins against each other using blastn and blastp (respectively)
- Round percent identity to the nearest 10% and tabulate



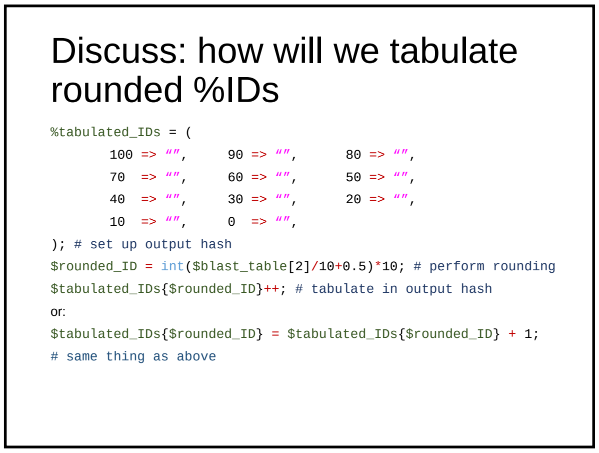
Rounding in perl is not trivial

- `int`: truncates decimals to integers

```
print int(1.4); # returns 1
print int(1.6); # returns 1
```

- So to round:
  1. Divide to convert to a decimal
  2. Add 0.5
  3. Apply `int`
  4. Multiply to revert divide

```
$number1 = 19;
print int($number1/10+0.5)*10; # returns 20
$number2 = 11;
print int($number2/10+0.5)*10; # returns 10
```

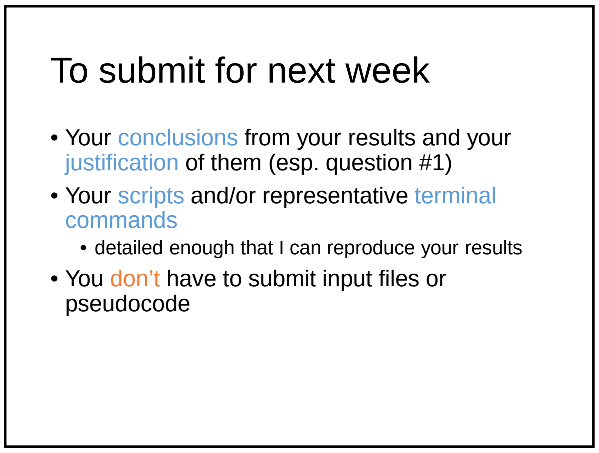


Discuss: how will we tabulate rounded %IDs

```
%tabulated_IDs = (
        100 => "",     90 => "",      80 => "",
        70  => "",     60 => "",      50 => "",
        40  => "",     30 => "",      20 => "",
        10  => "",     0  => "",
); # set up output hash
$rounded_ID = int($blast_table[2]/10+0.5)*10; # perform rounding
$tabulated_IDs{$rounded_ID}++; # tabulate in output hash
```

or:

```
$tabulated_IDs{$rounded_ID} = $tabulated_IDs{$rounded_ID} + 1;
# same thing as above
```



To submit for next week

- Your conclusions from your results and your justification of them (esp. question #1)
- Your scripts and/or representative terminal commands
  - detailed enough that I can reproduce your results
- You don't have to submit input files or pseudocode