

Genomics (Ecol 553) Computational Lab
Week 10: Oct 24, 2008.

Course webpage: <http://genomics.arizona.edu/553/computation/>

Topics

Perl: logical tests, arrays, hashes

Note

This homework is complete. There are only 3 questions.

Homework due at 11:59PM Monday, Nov 3

All work described below should be placed in a new directory on ice.hpc.arizona.edu, named: ~/homework/homework4. When you are ready to submit your work, you will cd to ~/homework, then run the command `turnin homework4`. (In case your PATH is broken, the `turnin` command is: `/home/u1/twheeler/553/bin/turnin`)

1) Write a program, called `blast_counts.pl`, which does the following:

- Reads in the blast result file `/scr1/twheeler/big.blast`;
- Counts the number of hits for each query sequence;
- For each query sequence, prints the query gi number and the number of hits for that gi ;
- Sort the output in descending number of hits. An example output (from a much smaller file) might be:

```
15594476: 10
15594575: 7
15594886: 7
15594662: 4
15594321: 2
```

2) Write a program, called `blast_high_pctid.pl`, which does the following:

- Takes two arguments
 - name of a file containing a list of gi numbers (I'll call it `gi_file`)
(an example can be found at `/scr1/twheeler/gi_nums`)
 - a percent identity threshold (I'll call it `pct_thresh`)
- Reads the list of gis from `gi_file`
- Reads in the blast result file `/scr1/twheeler/big.blast`
- For each query in `big.blast` that corresponds to an id in `gi_file`, prints:
 - The gi number for that query, and
 - The list of all hit (subject) gis with percent identity $>$ `pct_thresh`
 - (do not print self-hits)

For example, if you run:

```
blast_high_pctid.pl /scr1/twheeler/gi_nums 92.25
```

you might get something that looks like this :

```
15594476: 111114954, 51598395
15594662: 111115142
15594886: <--- there are no hits > 92.25%
15594575: 51598491, 111115055
.
.
.
```

Extra credit: sort these entries in order of decreasing number of hits exceeding `pct_thresh`. For example, the entries above would be sorted as:

```
15594476: 111114954, 51598395
15594575: 51598491, 111115055
15594662: 111115142
15594886:
```

3) Write a program, called `intersect_lists.pl`, which does the following:

- Takes two arguments for two filenames
- Reads the contents of those two files (they contain numbers)
- Prints out the list of numbers that are found in both files, sorted numerically.

For example, if one file contained:

```
1451
9123
8712
```

and another file contained

```
4318
9123
1451
9933
```

then your output should be

```
1451
9123
```

You can use `/scr1/twheeler/numbers1.txt` and `numbers2.txt` as your example inputs. Feel free to generate your own test files.

Please use hashes in your solution.

Extra credit: Do this two different ways. One way should use hashes, and one way should use ... something else. Name the script that implements the alternate method `intersect_lists2.pl`.