CSCI 652/490 Algorithmic Bioinformatics I Project 2 100 pts

Large-scale pairwise alignment between two genomic sequences is affected by a set of parameters, including substitution matrix, gap open, gap extension, seed pattern, etc., and a few score thresholds. In this project, you will study the impact of gap open penalty and alignment score threshold on alignment accuracy.

You will use 10 datasets of simulated sequences of human, chimpanzee, dog, and mouse. Datasets are located at /home/turing/mhou/csci652fall2018/data/simulation.

Each dataset contains four genomics sequences and three true alignment files: human vs. chimpanzee, human vs. dog, and human vs. mouse. You will use a pipeline of pairwise alignment tools to produce computed pairwise alignment, and then compare such alignment with true alignment file to determine accuracy of the computed alignment. Alignment accuracy is defined by sensitivity and specificity. Evaluation details are explained below.

The pair of input sequences are *X* and *Y*, and each sequence has *m* and *n* bases respectively:

X: $x_1,x_2,\ldots,x_m$

Y: $y_1,y_2,\ldots,y_n$

Suppose x_i is aligned to y_j in true alignment, and x_i is aligned to $y_{j'}$ in computed alignment. If $|j - j'| \leq c$, where c is a constant (default value is 5), we consider x_i and $y_{j'}$ correctly aligned.

Let *F* be the file of computed alignment, and *T* be the file of true alignment.

An aligned position in a pairwise alignment refers to a match or mismatch.

We then have:

Sensitivity = correctly-aligned-positions-in-*F*/all-aligned-positions-in-*T*

Specificity = correctly-aligned-positions-in-*F*/all-aligned-positions-in-*F*

You need to write a program to conduct such evaluation. You can use any preferred programming language.

A shell script (pipeline.sh) is prepared for you to run the pipeline of alignment tools to produce pairwise alignment between two sequences and save the alignment result in maf format. You need to try different parameters and modify this shell script to do experiments. See details below.

1. Project preparation:

1.1 Create a sub-directory `proj2` for this project, and create 10 sub-directories under proj2. Link data files to each of datasets.

```
mkdir proj2
cd proj2
for dir in 01 02 03 04 05 06 07 08 09 10
do
  mkdir dataset$dir
  ln –s /home/turing/mhou/csci652fall2018/data/simulation/dataset$dir/* dataset$dir
done
```

1.2 Copy the shell scripts to your sub-directory that contain all datasets:

```
cp /home/turing/mhou/csci652fall2018/progs/links.sh .

cp /home/turing/mhou/csci652fall2018/progs/pipeline.sh .
```

1.3 Make sure shell scripts are executable:

```
chmod 700 *.sh
```

1.4 Run the shell script links.sh:

```
./links.sh
```

Running this shell script will add links of all necessary pairwise alignment programs to each of the dataset sub-directory. You only need to run this script once.

2. Description on alignment and sequence file format:

2.1 maf format: you have worked with maf format in the first project. The computed alignment in this project is maf format. However, you will need the position information for each aligned sequence in an alignment block for this project.

2.2 (Simplified) fasta format: in an alignment block, each species has a line of header and a line of alignment text. The line of header is simply in form of ">speciesName". All alignment texts have the same length in the same alignment block.

A sequence file of fasta format has the same structure as the alignment file except that the alignment text is replaced by sequence, without any '-'s. If there are multiple sequences in the same fa file, the sequence lengths may be different. For your convenience, each fasta format sequence file only contains one sequence in this project.

3. The work you need to do:

3.1 Design and implement an evaluation program that compares a maf format pairwise alignment with a fasta format (true) pairwise alignment. Both alignment files are based on the same pair of sequences.

3.2 Try different parameters to examine impact of parameters on alignment quality.

a. In order to change parameters of computing pairwise alignment, you need to modify the line of "O=" and/or "L=" in `pipeline.sh`.
b. Change only one parameter for each experiment.
c. To run the pipeline of tools to create pairwise alignment, go to a sub-directory of data set, e.g., dataset01:

```
cd dataset01
```

Make sure you have copied or linked `pipeline.sh` to this directory, then execute:

```
./pipeline.sh
```

Go back to the parent directory by command:

```
cd ..
```

3.3 There are 10 datasets. You can compute average and standard deviation for each evaluation.

3.4 Write a report about your project.

4. Suggestions of major steps in working on this project:

4.1 Link all datasets and programs as instructed above

4.2 Design and implement your evaluation program using HUMAN and just ONE other sequence in ONE dataset to test and debug. Then test (and debug if necessary) on all sequences in the same dataset.

4.3 Then run experiments on all datasets and compute average and standard deviation. You may want to revise the shell script pipeline.sh or write a new shell script to automate this process.

4.4 Change parameter L (alignment score cut-off) and O (gap open penalty) for blastz to study the impact of such parameters on alignment results. Some suggested values to try: L=3000, L=5000, L=2000, O=400, O=300, O=500.

4.5 Change the value of *c* (specified in page 1 regarding sensitivity and specificity) to 0. Re-compute above experiments. Observe difference. You do not have to run all tests, but it is useful to run tests between HUMAN and all other species.

4.6 Summarize your results and write your report.

5. Submission: Submit your source code to Blackboard. Submit written report in class on due day.

6. Brief description of alignment tools used in this project:

6.1 blastz: pairwise aligner of two genomic sequences.

6.2 lav2maf: convert output of blastz to maf formt.

6.3 single_cov2: remove alignment of duplication or repeat from a maf format pairwise alignment file to make sure any position in one sequence is aligned to at most one position in the other sequence.

6.4 maf_project: organize alignment blocks based on coordinates of the reference sequence.

7. Shell script reference:

You may find this website very useful: https://www.shellscript.sh/index.html