Karl Menzel

CompBio

12/4/2015

Final Project

Rearrangements vs Point Mutations

For this project I used data on three species: human, chimpanzee, and mouse. I got my data from two major sources, the first being Cintiny.com and NCBI. The data that I got from Cintiny.com was the data about the synteny blocks. This gave me the number of rearrangements needed to get from humans to chimp and human to mouse. From that source I also got the synteny blocks with coordinates and which blocks line between the species compared. For this data I sorted out the synteny block that were on the X chromosome. I also added a data column for distance and sorted the rows with the shortest one first and the longest one last. I also changed the species codes to the names of the species so that it would be easier to look at. The data that I got from NCIB I download the X chromosomes for human, chimpanzee, and mouse. I did not alter the sequences that were read in from these chromosomes. I only took segments to use in the alignment for the program.

The programing for this project was sectioned into three major sections: first is reading and handling of the synteny and chromosomal data, the second is sliding window aligner, and the third is putting both together to be able to analyze the right parts. For the first part this was mostly dealing with annoying formatting in the output file. After getting it into the right format I changed two major things. The first was adding two columns for length, one for each species. To do this I too the difference between the two coordinate columns and then added that column right after the two distances. After adding the lengths of each of the segments I then sorted all of the sequences such that the lowest was first. Chromosomes were read in with a simple reading function. For the window alignment, I used the simple longest common substring as the scoring because I was using nucleotides so I could not use a

Pam or Blossom matrix because finding the right open reading frame was way out of the scope of this project. The most interesting part of the alignment was that I used a sliding window instead of a table for the dynamic method of alignment. It only kept track of one line of what would be the dynamic table and updated it as it went along. The third part of this project was to put both parts together to be able to get a result. What I did here was read through the synteny tables that were created in part one and get the segments of the appropriate chromosomes and feed them into the sliding window alignment. Because these are really large strings of DNA these operations take forever I wrote the results of each of the alignment to a file after they were done so that I did not need to be done with all of them before all of them were done.

At this point I was able to analyze two sets of the synteny blocks, one for human and chimpanzees and for humans and mice. The data for the chimp human comparison seems pretty reasonable 59.7% and 67.5% coverage of the shorter strand. This isn't that good considering the chimpanzee and human genome are supposed to be extremely similar but I was just using the longest common substring method of alignment, just a simple plus one when there is a match and no penalty for indels or mismatches. I do think this is good enough for the comparison that I am trying to make now because I am just trying to compare one thing to another so the kind of errors I will be making will be the same for both so it will most likely work out. I am assuming that the rates indels and mismatches are the same between all three species which I think is pretty valid. The data for the mouse human comparison does not make any sense. I figured out it was an error in my code. I am currently reading calculating what it should be but if you are reading this then I didn't have time to complete the run because I figured out my mistake way too close to when this was due. This program was very difficult because almost everything took a really long time to happen. It about an hour to read in one of the genomes and the comparisons took even longer. This made de bugging really hard because a lot of things had a very weird formats and setting up a dummy variables is really difficult to do.