BIO/CMPSC 300 Introduction to Bioinformatics
Spring 2016
Kristen Webb and Janyl Jumadinova
http://cs.allegheny.edu/sites/jjumadinova/300

Lab Week 12 – Advanced Gene Prediction

Given: Friday April 8, 2016
Due: Friday April 15 by 2:30pm

Objectives:

- **Understand how eukaryotic genes introduce additional complexity into the problem of gene prediction and recognize the limitations of sequence-based methods**
- **Know some content-based methods of gene prediction and appreciate their strengths and limitations**
- **Be able to combine content-based and probabilistic methods of gene discovery to identify the most probable locations of introns and exons in a eukaryotic DNA sequence**
- **Know how to design an HMM to integrate sequence and content data for a more precise and accurate determination of exon-intron boundaries**

Reading Assignment:

Chapter 10 in Exploring Bioinformatics textbook.

Required Deliverables (submitted through your Bitbucket repository):

Please ensure that your bitbucket repository contains a directory named 'lab12' and that your report and your program (s) contain the names of all team members and the honor code pledge. Additionally, if you create a new team repository, you have to name it '300s2016-lab12_lastNames' and share it with both instructors.

1. An electronic submission of the report containing the answers to the GENSCAN Prediction Questions.

2. A properly formatted and commented Python program that implements other types of alignments and an output that your program produces.

General Guidelines for Labs

Work on the Alden Hall computers. If you want to work on a different machine, be sure to transfer your programs to the Alden machines and re-run them before submitting.

Keep all of your files! Don't delete your programs and reports after you hand them in---you might need them again later.

Back up your files regularly. Use a flash drive or Google Drive or whatever your favorite backup method is.

Review the Honor Code policy on the syllabus. Remember that you may discuss experiments and programs with others, but copying answers or programs is a violation of the Honor Code.

Part I: Background

As an influenza researcher, you have become interested in a small number of individuals you know where unvaccinated and repeatedly exposed to the 2009 H_1N_1 influenza virus but did not become ill. When immunological testing showed they were not actually immune to the virus (already possessing antibodies), you began to seek a genetic link that might explain their resistance to this disease. Using next-generation sequencing, you were able to identify common transcripts (mRNA sequences) from respiratory epithelial cells that are missing in your resistance patients. This leads you to sequence a particular genome region in one resistance patient, some 90,000 bp (90 kb) from the 1q25.3 region of chromosome 1. You would now like to analyze that genome fragment to identify genes within it that might be involved in susceptibility or resistance to influenza.

Part I: Gene Prediction with GENSCAN

The number of available tools for gene prediction is somewhat mind-boggling. Several popular gene prediction programs are comprehensive in nature, bringing together several kinds of analysis in one piece of software; these would be a good place to start the analysis of a genome sequence or segment. We will work with the gene prediction program GENSCAN.

GENSCAN combines Hidden Markov Model (HMM)-based models for coding-region and exon/intron boundary prediction with models that attempt to account for additional factors that affect exon/intron boundary choice as well as observed changes in exon/intron boundaries and gene density in low-GC versus high-GC regions of human DNA. GENSCAN claims to correctly identify 70-80% of known exons. This comprehensive program produces clear and compact graphical output, making it easy to compare to other programs' results.

1. Get the file 1q25.txt from the shared repository. This file contains 90 kb of FASTA-formatted DNA sequence from human chromosome 1.
2. Navigate to the web-based implementation of GENSCAN at the Pasteur Institute (http://mobyle.pasteur.fr/cgi-bin/portal.py#forms::genscan). Click the upload link and upload your sequence.
3. From the Organism menu, choose the training set appropriate for analyzing human DNA.
4. Click the advanced options button and choose to include Verbose output (provides additional information in the output files) and Create Postscript output (gives a graphical representation of the results).
5. Run the program. If this is your first time running GENSCAN you may be asked to verify your run.
6. When the results appear, you will see a window labeled "standard output" containing output. Click the full screen button for easier viewing.

   This output includes the specific location of the prediction introns and exons, information on reading frames and splice sites, and translations for the putative coding regions. There is also useful information about the reliability of the predictions.

   Returning to the Results page, there will also be a window labeled "PostScript output". This is a graphical representation of the results. Click the full screen button for easier viewing. (if viewing options are not available on Linux, a .pdf version of the graphical output has been uploaded to the shared repository). The key at the bottom of the page describes the graphical representation used to show potential coding regions across the 90 kb sequence.

Part I: GENSCAN Prediction Questions

1. List the genes that GENSCAN found within the sequence, along with their orientation (direction), number of exons, lengths, and the approximate length of the processed mRNA that will be translated into protein. Use a table with the following headings:

Gene # (1, 2, 3…)	Direction (to right/to left)	Number of Exons	Length (in nucleotides)	Processed Length (in nucleotides)

2. In the graphical representation of the results, why do the arrows point in different directions?

3. In the text output, what is the difference between an exon marked Init and an exon marked Intr? Why is this difference significant in prediction genes?

4. Look at the amino acid sequences of the predicted proteins. Look at how the predicted proteins begin. Does this information strengthen or weaken the case for any of the gene predictions?

5. Looking at the text output, what other features did GENSCAN identify? Do these provide additional support for any of the predicted genes?

Part II: CpG Island Prediction Algorithm

CpG island prediction is an example of content-based algorithm that searches sequences for CpG islands that may indicate a nearby promoter. An increase in the frequency of CG pairs has been observed between nucleotides -1,500 and +500 relative to a transcriptional start site; finding such a CpG island appropriately positioned upstream of a putative gene would strongly contribute to the hypothesis that it is an actual gene.

In a random DNA sequence, we expect CG nucleotides to occur once in every 16 nucleotides, where 1 of every 4 nucleotides should be a C and the next nucleotide will be a G ¼ of the time. To identify CpG islands, we need to search for the CG sequence pattern, and more importantly, determine its frequency. A frequency-matching algorithm can accomplish this task. Here, we use a sliding window to traverse the sequence, keeping the count of the CG pairs within each window and searching for higher than average CpG ratios. The following algorithm (in a pseudocode form) shows how this could be done. In this pseudocode, all CpG ratios are stored and displayed, but if a CpG ratio is greater than 1.5 (strong indicator), starts (***) print next to the value to highlight the ratio. Another alternative is to only print the windows where the ratio is greater than 1.5.

CpG Island Prediction Algorithm (p. 210 of the book)

Goal: To identify regions of CpG islands
Input: A FASTA formatted input file containing a sequence
Output: Window start positions, CpG ratios, and text indicating high ratios

Step 0: Initialization - read a sequence data
Open the input file containing the sequence
Get the input window size from the user, save into variable named window
Read and discard the first line (fasta format) from the input file
for each remaining line of data in the input file
    seq = seq + line

Step 1: Determine CpG ratios

```
lenSeq = length of the sequence
ratios = array of size lenSeq-window+1 (holds CpG ratio of each window)

for each i from 0 to lenSeq-window+1
        cCtr = gCtr = cgCtr = 0

        for each j from 0 to window-1
                if seq[j+i] == 'C'
                        cCtr++
                        if seq[i+j+1] == 'G'
                                cgCtr++
                 else if seq[j+i] == 'G'
                        gCtr++
        if cCtr*gCtr != 0
                ratios[i] = cgCtr/((cCtr*gCtr)/window)
        else
                ratios[i] = 0
```

Step 2: Print window start position and CpG ratios

```
for each i from 0 to length of ratios
        if ratios[i] > 1.5
                output i+1, ratios[i], `***'
        else
                output i+1, ratios[i]
```

1. Your task for Part II is to implement the CpG algorithm in Python given the pseudocode. A pseudocode is a description of an algorithm using structural conventions of programming languages, but not using a particular language specific syntax. You can modify the pseudocode as long as your output is unchanged. Your goal is to read in a sequence from a file and produce a tabular list if high-CpG regions with their scores into a text file. Design simple test sequences to test your program, and then run it on the long sequence (1q25.txt) used in Part I of the lab.

2. Navigate to the Sequence Manipulation Suite, click on CpG Islands link on the left, upload your test sequence and submit it. Study the results obtained from the Sequence Manipulation Suite and compare the output of your program with the output of the CpG island prediction tool. How similar are the predictions? Suggest an explanation for any discrepancies.

Optional Part:

1. The program described in Part II has the same problem as the CpG island prediction tool from the Sequence Manipulation Suite. Since it shows each window where the CpG ratio exceeds a threshold value, it produces a long list of overlapping CpG islands. Make the output of your program more user-friendly by merging overlapping CpG islands into single entries in the results table.

2. To make your program more effective, you might apply additional criteria. CpG islands associated with actual promoters are usually at least 500 bp in length and have an overall G+C content greater than 55% and a ratio of observed to expected CpG pairs exceeding 65%. Implement these additional criteria as part of your program.