Enhancer Prediction - Project

Research Workshops

April 17, 2023

1 Project description

As part of the project, in groups of 3 people, you must train a classifier for detecting enhancer sequences based on DNA sequence data and the enhancer database.

1.1 Tasks

1. First, you will need enhancer DNA sequences that will serve as positive data to train your classifier. You can perform all data preparation operations using appropriate bioinformatics libraries/programs or write your scripts. As part of the assignment, you must show the entire pipeline - code that must be reproducible so that I can easily run and obtain your results.

2. From the enhancer atlas 2.0 database (http://www.enhanceratlas.org/), download the file with the coordinates of enhancers for the GM12878 cell line. The file is in BED format, where the first three columns are the genomic coordinates of a given region, and the following columns contain various annotations - in the case of Enhancer Atlas, it is the consensus score that a given region is an enhancer (https://academic.oup.com/nar/article/48/D1/D58/5628925). You can find the negative set on Moodle. These are the coordinates of random sequences of lengths corresponding to positive sequences. Extract the DNA sequences using both these files and the FASTA genome sequence file. Now we need the appropriate DNA sequences. Find the file with the genome ("human primary assembly") (Genome sequence, primary assembly (GRCh37) from https://www.gencodegenes.org/human/release_43lift37.html).

3. NOTE: If you are not using the bedtools library, please note that the coordinates in BED files are indexed from 1 (not 0), with the minimum value being 1.

4. Some DNA sequences (especially those from the negative set) may have "N" symbols, indicating that the nucleotide there is unknown (due to the imperfection of the sequencing experiment) - remove them.

5. The test set will consist of chromosomes: chr1, chr14, and chr21, while the training set will consist of the remaining autosomal chromosomes: chr2-13, chr15-20, and chr22. All others should be removed (e.g., chrY, chrX, chrM, or anything not in the format chrNumber_Chromosome).

6. Count the frequencies of k-mers similarly to the method practised during classes. In example 4-mers, as a result there should be 136 columns for unique 4-mers. In a given column, the frequency of a given k-mer should be counted: the number of occurrences divided by the sequence length. NOTE! We don't know which DNA strand the enhancer is on, so we must also consider reverse complementary k-mers, i.e., for ATCG, the reverse complement is CGAT. We count the frequency of the k-mer and its reverse complementary version as one feature. If we encounter ATCG or CGAT in the sequence, we consider that ATCG occurred twice (or that CGAT occurred twice, depending on which label we assign to the pair). The value of "k" in k-mers depends on you - a test which values give you the best results.

7. The choice of classifier, k-mer length, and other parameters appropriate for the classifiers is up to you. The only constraint is that the classifier should be binary,

8. Provide appropriate metrics for the results (i.e. ROC, AUC)

What can be useful:

1. Biopython (https://biopython.org/), where functions for extracting DNA sequences, analysis, etc., are already implemented,
2. Bedtools (https://bedtools.readthedocs.io/en/latest/), a very fast software (core written in C) for filtering, finding intersections, searching records in file formats typical for bioinformatics, such as BED (https://en.wikipedia.org/wiki/BED_(file_format)) or FASTA (https://pl.wikipedia.org/wiki/FASTA_format),
3. There is also a python wrapper for bedtools, pybedtools (https://daler.github.io/pybedtools/).

Good luck!