Algoritmos Bioinformática /Bioinformática 2023/2024

Group Assignment 3

Protein Sequence Classification

1. Goal

In this assignment, we will perform an exploratory project to evaluate the possibilities of classifying protein sequences based solely on their sequences, in particular the dinucleotide composition of the sequence.

The idea is to use a machine learning approach to achieve this task. Packages like Pandas and Scikit-learn can be used for that purpose. However, first you will have to pre-process the sequences so they can be represented in a tabular format. Then, the respective protein family (class) label needs to be associated as an extra column on the table. The goal will be to predict this label.

Once sequence data is represented in such a tabular format, you can apply train/test or cross-validation procedures using *sklearn* to evaluate the performance of the classifier.

Goal: given the protein sequences from two protein families (globin and zinc finger), create a predictive model for a binary classification problem.

2. Data

You will receive two protein sequence families: zinc finger and globins.

"The globins are a superfamily of heme-containing globular proteins, involved in binding and/or transporting oxygen. These proteins all incorporate the globin fold, a series of eight alpha helical segments. Two prominent members include myoglobin and hemoglobin." **Wikipedia**.

"Zinc-finger proteins (ZNFs) are one of the most abundant groups of proteins and have a wide range of molecular functions. Given the wide variety of zinc-finger domains, ZNFs are able to interact with DNA, RNA, PAR (poly-ADP-ribose) and other proteins." https://www.nature.com/articles/cddiscovery201771

See files globin.fasta and zincfinger.fasta

3. Data pre-processing

For data pre-processing the goal is to transform the sequence data in tabular format. This will allow to all sequences be compared against each other directly. Then, different machine learning algorithms can be applied.

The **Feature Frequency Profiles (FFP)** approach will be used. Confer the Methods section of the following paper:
https://www.pnas.org/doi/epdf/10.1073/pnas.0813249106

The idea is that for an alphabet A, and for a k-mer of 2, A^2 combinations of the alphabet symbol will be counted. In the case of proteins this will be 20^2

```
amino_acids = ['A', 'R', 'N', 'D', 'C', 'Q', 'E', 'G', 'H', 'I', 'L',
'K', 'M', 'F', 'P', 'S', 'T', 'W', 'Y', 'V']
```

Then we will have AA, AR, AN,….., YV, VV in a total of 400 two letter combinations.

Scan the input sequence and count the absolute frequency of each pair of amino-acids <i,j> as $C_{i,j}$. Then, normalize the frequency by the total number of 2-mers in the sequence $F_{i,j} = C_{i,j}$ /|2-mers|

The vector with all the 400 $F_{i,j}$ will correspond to the FFP representation of one sequence. This will then be updated in the FFP table for all the sequences.

4. Protocol

The following steps are suggested to guide the work. Feel free to use other approaches.

1. Create a function to read the sequences from fasta files. Return the data as a dictionary where the id of the sequence is the key and the sequence is the value in the dictionary.

   Example in the globin file, for the first sequence:

   ```
   >sp|E2RTZ4|HMP_GIAIC Flavohemoprotein OS=Giardia intestinalis (strain
   ATCC 50803 / WB clone C6) OX=184922 GN=hmpA PE=1 SV=1
   ```

   Use E2RTZ4 as the identifier of the sequence.

2. Create a function that generates all the 2-mers of Amino-Acids (AAs).

3. Create a function that creates and fills a pandas dataframe, with the FFP values for all the sequences in both input files. The columns should have the 400 $F_{i,j}$ dinucleotide values while the rows should correspond to each sequences in the two datasets identified by the respective sequence id.

4. Add an extra column (class) to the table to include the label for the type of protein, *i.e*. zincfinger or globin. This can be encoded as a 0 or 1 value.

5. Create a classification pipeline with *sklearn*.

    a. Use 2 or 3 Machine Learning algorithms, *e.g.* SVM, Random Forests or NaiveBayes.
    b. Create a cross-validation object with 10 folds. Choose one of the cross-validation approaches, for instance Stratified k-fold or Leave-one-out.
    c. Evaluate the models with 3 or 4 measures, *e.g.* recall, precision or F1-score.
    d. Get the average and standard deviation across the 10 folds.
    e. Compile in a table the results for the different methods and highlight in bold the best performing one.

6. Write a small discussion on the results. Remember that there is no *a priori* expectation on the results, so these will be totally novel results requiring interpretation.

5. Report

For this assignment, you should submit:

- Report – a small report with 1 page maximum, describing your approach, the main results, the difficulties, the tools and packages used. You should include the table with the results and finish with a conclusion. Submit as a .**pdf** file.
- Code – submit the developed script. You should have a main file called *classify_proteins.py*. Note that this should be a unique script and all the results should be generated in the current directory. As an example, the script should be called:

```
python classify_proteins.py –a zincfinger.fasta –b
                globin.fasta –k 2
```

In the command above k is the length of the k-mer, in this case is 2. Parameters *a* and *b* are the names of the input fasta files.

Submit a **zip** file with **2 files**: report and code as described above.

Do not forget to mention in the report all the students who contributed for the assignment and any particular notes on their specific contributions.