CSI 5180. Topics in Artificial Intelligence
Machine Learning for Bioinformatics Applications
Fall 2019

Assignment 1

Deadline: October 10, 2019, 18:00

[PDF]

Learning objectives

- **Introduce** a basic type of data used in bioinformatics
- **Prepare** and **encode** biological data for a machine learning task
- **Learn** how to encode variable length sequences into vectors
- **Explore** the programming environment proposed by **Scikit-Learn**
- **Apply** an unsupervised learning algorithm to biological data
- **Determine** the optimal number of clusters for a given data set

Introduction

The international journal of science **Nature** defines the **microbiome** as follows:

> The microbiome comprises all of the **genetic material** within a microbiota (the entire **collection of microorganisms in a specific niche**, such as the human gut). This can also be referred to as the metagenome of the microbiota.[1]

Together with our life science collaborators, we are studying the microbiome of the human skin. Using next generation sequencing techniques, our collaborators were able to identify a list of organisms often found on the human skin. As a next step in this study, we want to compare the whole genomes of these organisms to identify subgroups[2].

It is important to note that the size of DNA sequences (strings) vary greatly, as illustrated by the table below, which makes a direct comparision challenging[3].

Species	**Size (nt)**
Potato spindle tuber viroid (PSTVd)	360
Human immunodeficiency virus (HIV)	9,700
Bacteriophage lambda (λ)	48,500
Mycoplasma genitalium (bacterium)	580,000
Escherichia coli (bacterium)	4,600,000
Drosophila melanogaster (fruit fly)	120,000,000
Homo sapiens (human)	3,000 000,000
Lilium longiflorum (easter lily)	90,000,000,000
Amoeba dubia (amoeba)	670,000,000,000

[1] https://www.nature.com/subjects/microbiome

[2] There are several well established bioinformatics methods to solve the problem. We opt for a machine learning approach.

[3] How do you compare strings with such large differences in length?

One of the widely used measure of distance for comparing genomic sequences (Levinstein distance) requires computing time increasing as the square of the size of the sequences being compared. This limits its application to sequence fragments (sub-strings). Furthermore, in its most basic form, it assumes that the sequences are approximately of the same length.

Herein, we take an approach that will go well with the machine learning approaches that we will be using throughout the course. Several publications, including one by Yang suggests that sequences can represented as l-tuple frequency vectors [Yang et al. 2008].

1 Encoding (5 points)

I should be able to execute your program on the command line as follows:

```
> python a1.py 6 human_skin_microbiome.csv
```

where **a1.py** is the name of your program, **6** is the size of the tuples (a user-defined parameter), and human_skin_microbiome.csv is the name of a coma-separated-value file. In that file, the first field is the scientific name of the organism and the second field is the URL of a compressed FASTA file [4] containing the entire genome of the organism. I should be able to run your program without having to download additional libraries other than these [5]:

- Standard libraries, such as **gzip**, **os**, **re**, **urllib**
- Standard data science libraries: **numpy**, **matplotlib**, **sklearn**

You program must download and process all the genomes. Each genome must be transformed into a frequency vector for all tuples of size l, where l is a command line argument. FASTA is one of the simplest file format. Such file contains one or more sequences. Each sequence begins with a single-line description, preceded by '>' — the information on those lines can be ignored. The data usually spreads over several lines, where each line is usually (but not always) 80-character long or less. This makes it easier to visualize the content of the files and was necessary on some older computer systems.

Genomes can be incomplete (i.e. having holes) or made of several chromosomes. Accordingly, each genome file will have several sequence entries. You must read an process all the entries of a given genome to produce the frequency vector.

Given an input sequence (string) S and a parameter l, X_i is the frequency of the tuple $[ACGT]^l(i)$ in S. Herein, words (tuples) containing symbols other than A, C, G, or T are ignored.

Clearly, two identical or similar sequences will have frequency vectors that are identical or similar. The further apart the sequences are, as mutations accumulate, the more dissimilar the frequency vectors will be. This representation has many advantages and limitations. It allows to compare sequences of different lengths. But also, it is tolerant to evolutionary events where the order of segments is rearranged or some segments are duplicated. However, a significant amount of information is discarded.

Given an string $S = GAAGAC$, over a four letter alphabet: A, C, G, T, its corresponding frequency (distribution) vector for words of size 2 is as follows:

```
AA = 1/5
AC = 1/5
AG = 1/5
AT = 0
CA = 0
CC = 0
CG = 0
CT = 0
GA = 2/5
GC = 0
GG = 0
```

[4] https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=BlastHelp

[5] Let me know if you think that additional libraries are needed.

```
GT = 0
TA = 0
TC = 0
TG = 0
TT = 0
```

2 Analysis (5 points)

Each input genome is now represented as a l-tuple frequency vector.

2.1 KMeans (3 points)

1. Apply **KMeans** to your input data for all possible values of k.
2. Create a figure, using **matplotlib**, showing the **inertia** of the clusters for all possible values of k.
3. Create a figure, using **matplotlib**, showing the **silhouette score** of the clusters for all possible values of k.
   (a) Based on that plot, what is the optimal number of clusters? Your program, should print the optimal number of clusters on the standard output.

2.2 Dendrogram

Finally, using **dendrogram** and **linkage** from **scipy.cluster.hierarchy** create a figure showing the result of a hierarchical cluster analysis based on single linkage. Make sure that the names of the species are displayed at the leaves of the tree.

Files

- human_skin_microbiome.csv
- a1.py (your program)
- report.pdf (short description of your program together with your answers for the analysis questions)

Reference

- Yang et al. (2008) Performance comparison between k-tuple distance and four model-based distances in phylogenetic tree reconstruction. *Nucleic Acids Res.* 36(5):e33, doi:10.1093/nar/gkn075.
- Gregory, T.R. (2008-03-23) Animal Genome Size Database — www.genomesize.com.
- Genome biology (2008-03-23) — www.ncbi.nlm.nih.gov/Genomes.

Last Modified: September 24, 2019