Introduction to Spark

1 Description

The goal of this part of the assignment is to learn to do some distributed computing using Spark and PySpark.

There are 2 different datasets that you will use:

1. Rx dataset: Medication prescriptions in the United Kingdom from July 2016 to September 2017
2. Bioinformatics dataset: Tardigrade and bacteria genome sequences

There are 3 tasks:

- Rx dataset
  1 Compute the total "net ingredient cost" of prescription items dispensed for each PERIOD
  2 Compute the 5 practices that issued the prescriptions with the highest total net ingredient cost
- Bioinformatics dataset
  3 Compute and label each sequence from a provided sample as most likely being Tardigrade or bacterial using Edit Distance.

2 Datasets

2.1 Rx Dataset

The data set itself is a set of simple text files. Each prescription/prescribing practice is a different line in a file. The attributes present on each line of the files are, in order:

Attribute	Description
SHA	Area team identifier
PCT	Clinical commissioning group identifier
PRACTICE	Practice identifier
BNF_CODE	British National Formulary (BNF) code
BNF_NAME	BNF name
ITEMS	Number of prescription items dispensed
NIC	Net ingredient cost (pounds and pence)
ACT_COST	Actual cost (pounds and pence)
QUANTITY	Quantity - whole numbers
PERIOD	YYYYMM

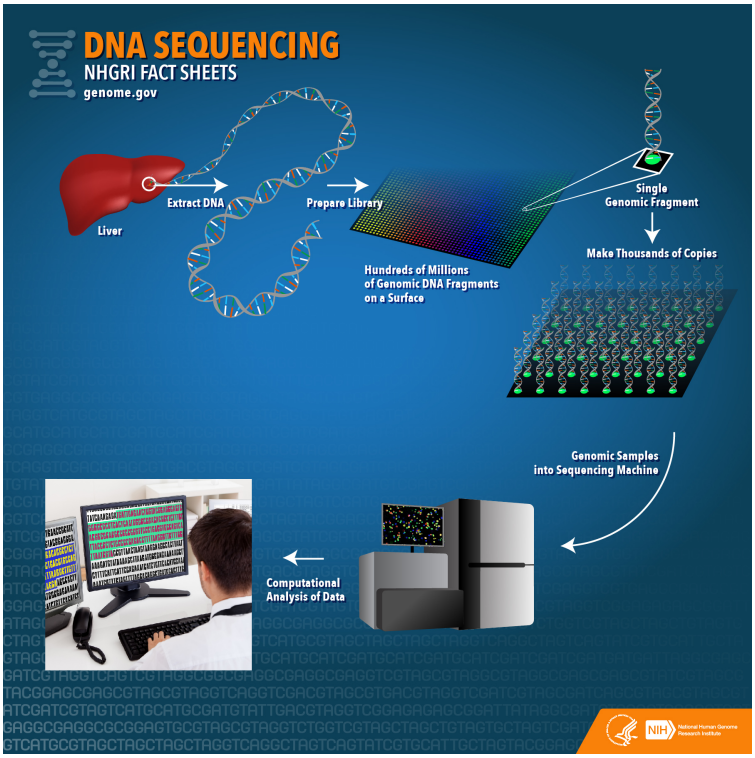
The data files are in comma separated values (CSV) format. It is stored as fifteen different files.
This is about 21 GB of data in all. Important note: be aware that the URLs may not copy-and-paste from this PDF correctly, as you may lose underscore characters. This problem may happen with the other commands below.
For testing and development, you can run your Spark jobs on just one of the fifteen files.

A super-small subset of the first file (only about 1000 lines) is available for download (see Canvas). This file may be used on your computer using Docker and the Spark container. If you want, you can also use this file for testing and debugging by loading it into HDFS (just like you did in lab) and then running your Spark program over it.

2.2 DNA Sequence Dataset

Background

Genomes are sequenced to help us better understand the genetic make-up of an individual or organism. The basic process involves taking a DNA sample (e.g. saliva), breaking it up into small pieces, choosing random fragments, determining the genetic composition of the fragments, and then ordering the fragments into a coded sequence. The machines used to sequence genomes are expensive, and are often provided as a service to an organization or as a remote lab. The following diagram from the National Human Genome Research Institute illustrates the process.



Consequently, there is the risk of contamination at many steps of the process when a genome is sequenced. When contaminated, genetic material from other organisms are introduced into the sample, and are included in the assemblies that are output from the sequencing device. Including this genetic content in the genome of the target organism can be misleading and lead to errors in research or diagnosis.

In this assignment, we will examine the assemblies from a contaminated tardigrade genome. We will compare the coded assemblies (or "contigs") with codes from a clean tardigrade set of assemblies as well as with codes from a number of bacteria.

The goal is to determine, for each coded segment of the contaminated sequence, whether or not it is likely to be bacteria or actual tardigrade DNA.

Tardigrades

What is a tardigrade and why are we looking at this problem?

Tardigrades, also known as "water bears" are micro-animals that live in the water. They are caterpillar-like, with 4 pairs of legs and segmented bodies. They are ubiquitous and resilient. They have found just about everywhere in the world. (`https://en.wikipedia.org/wiki/Tardigrade`)

In 2015, Boothby et al. published a paper claiming that the tardigrade's ability to survive extreme conditions is due to horizontal gene transfer (HGT)(transfer of genetic material between species) from many different species, including bacteria, fungi, and plants [1].

Koutsovoulos et al. investigated Boothby's claim and rebutted it. Basically claiming that the evidence seen was DNA sample contamination, not actual HGT[2].

All of these papers (including the appendix for the rebuttal) are included in the assignment.

The data

- The contaminated tardigrade assemblies are in `s3://risamyersbucket/tardigrade/LMYF01.1.oneline.fa`.
  You will be comparing these contigs with contigs in the following other files:
- The "clean" tardigrade reference assemblies are in the file `s3://risamyersbucket/tardigrade/nHd.2.3.abv500.oneline.fa`.
- Bacterial contigs are in the file `s3://risamyersbucket/tardigrade/exp1.oneline.fa`.

Each file contains a set of lines, one line per contig. Valid lines start with the '>' symbol, followed by the organism name. Next is a vertical bar ('|') followed by a unique identifier for the contig within the organism. There may then be additional text describing the contig. Finally, there will be a '<' symbol. After this symbol, the remaining text on the line contains the DNA code. As you may know, this text consists of the characters A, C, T, and G.

Valid contig lines start with a '>' and contain only the specified letters in the DNA code. You should only include valid lines in your analysis.

3 The Tasks

There are three programming tasks associated with this part of the assignment. They involve analysis on the datasets.

3.1 Task 1: 35 Points

Write a PySpark program that checks all of the files and computes the total "net ingredient cost" of prescription items dispensed for each PERIOD in the data set (total pounds and pence from the NIC field).

As you do this, be aware that this data (like all real data) can be quite noisy and dirty. The first line in the file might describe the schema, and so it doesn't have any valid data, just a bunch of text. You may find lines that do not have enough entries on them, or where an entry is of the wrong type (for example, the NIC or ACT_COST cannot be converted into a decimal number. Basically, you need to write robust code. If you find any error on a line, simply discard the line. Your code should still output the correct result.

For your results, print out each period, in sorted order, followed by the total net ingredient cost for that period.

3.2 Task 2: 30 Points

Write a PySpark program that computes the 5 practices that issued the prescriptions with the highest total net ingredient cost in the data set. This is actually quite simple to do, and very similar to Task 1.

3.3 Task 3: 35 Points

Your task is to classify each sequence in the contaminated tardigrade file as being most likely bacteria or tardigrade. To measure the similarity of two sequences, you will compute the Edit Distance (https://en.wikipedia.org/wiki/Edit_distance). Concretely, you need to write a function calculating the minimum amount of steps that is required to transform a sequence into the other, given two sequences. Some examples:

Sequence A	ACCTTGC	CTGCCAA	ACTGCTG
Sequence B	CCCTTAC	CTGCCAA	CTGCTGA
Edit Distance	2	0	7

Note that since sequences in the data are all truncated to a same length. Your function does not have to be too complicated.

Then, for each sample sequence:

- Compute the Edit Distance against all the clean and bacterial sequences
- Find the group of sequences that have the shortest distance
- Label the sample as the majority of the group, or "Not sure" in case of a tie

Some examples:

		Contaminated Sample			
		ACTGA	ACCTT	GCCAA	GTGCA
Clean	CCAGG	**3**	4	4	5
	GCAAA	**3**	4	**1**	**3**
Bacteria	ACGCT	**3**	**2**	4	**3**
	TATCG	4	5	5	4
Label		Clean	Bacterial	Clean	Not Sure

Once you labeled the contaminated sample set, answer the following question:

1. If you are working on subset dataset, please print all the classifications. For full dataset, please give the classification[Clean, Bacterial, or Not Sure] for the following contigs instead:
   (a) LMYF01000203.1
   (b) LMYF01002593.1
   (c) LMYF01004256.1
   (d) LMYF01007394.1
   (e) LMYF01009100.1
2. How many sequences in the contaminated file are believed to be bacterial sequences?

4 Important Considerations

4.1 Small data vs. big data

We have provided a small subset of the data that you can use locally to develop and test your code. However, to get full credit, you must run your code on the full datasets.

4.2 Machines to Use

One thing to be aware of is that you can choose virtually any configuration for your EMR cluster—you can choose different numbers of machines, and different configurations of those machines. And each is going to cost you differently! Pricing information is available at:

`http://aws.amazon.com/elasticmapreduce/pricing/`

Since this is real money, it makes sense to develop your code and run your jobs on a small fraction of the data. Once things are working, you'll then use the entire data set. We are going to ask you to run your Spark code over the "real" data using two c3.2xlarge machines as workers.

This provides 8 cores per machine (16 cores total) so it is quite a bit of horsepower.

Be very careful, and shut down your cluster as soon as you are done working. You can always create a new one easily when you begin your work again.

Academic Honesty

The following level of collaboration is allowed on this assignment:

You may discuss the assignment with your classmates at a high level. Any issues getting Spark running is totally fine. What is not allowed is direct examination of anyone else's code (on a computer, email, whiteboard, etc.) or allowing anyone else to see your code. You may also use (and in fact are encouraged to use) the Spark reference manual

`https://spark.apache.org/docs/2.3.1/`

You may use the search engine of your choice to look up the syntax for Spark commands, but may not use it to find answers.

You MAY post and discuss results with your classmates.

5 Turnin

Create a single document that has results for all three tasks and a screenshot indicating that you used 16 cores on AWS. Then zip up all of your code and the document (use .gz or .zip only, please!), or else attach each piece of code as well as your document to your submission individually.

6 Grading

If you get the right answer and your code is correct, **and you make use of the 16 cores available in your cluster** to get parallelism using the Spark framework, you get all of the points. If you don't get the right answer or your code is not correct or you don't make use of the cores, you won't get all of the points; partial credit may be given at the discretion of the grader.

Bibliography

References

[1] T. C. Boothby, J. R. Tenlen, F. W. Smith, J. R. Wang, K. A. Patanella, E. O. Nishimura, S. C. Tintori, Q. Li, C. D. Jones, M. Yandell, Evidence for extensive horizontal gene transfer from the draft genome of a tardigrade, Proceedings of the National Academy of Sciences 112 (52) (2015) 15976–15981.

[2] G. Koutsovoulos, S. Kumar, D. R. Laetsch, L. Stevens, J. Daub, C. Conlon, H. Maroon, F. Thomas, A. A. Aboobaker, M. Blaxter, No evidence for extensive horizontal gene transfer in the genome of the tardigrade hypsibius dujardini, Proceedings of the National Academy of Sciences (2016) 201600338.