Medical Bioinformatics

Project

SS15

Stefan Jacob & Zarin Shakibaei

Outline

1. Introduction
2. Data
3. Methods
4. Results
5. Conclusion
6. Introduction
7. Data

Two different data archives have been downloaded.

The first archive comes from the Genome Characterization Center (GCC) and contains txt files with gene samples and their corresponding expression level (log2 lowess normalized). The second archive which is provided by Biospecimen Core Resource(BCR) contains carefully cataloged tissue and sampleinformation with important medical information about the patient. Each text file from the BCR contains other information e.g. the gender of the patient or the treatment, used drug and some information about the tumor (size, site, location, etc.)

Among the biotab txt files, eight contained useful information which have the following names:

nationwidechildrens.org\_biospecimen\_cqcf\_luad.txt

nationwidechildrens.org\_clinical\_cqcf\_luad.txt

nationwidechildrens.org\_biospecimen\_tumor\_sample\_luad

nationwidechildrens.org\_clinical\_cqcf\_luad.txt

nationwidechildrens.org\_clinical\_follow\_up\_v1.0\_luad.txt

nationwidechildrens.org\_clinical\_omf\_v4.0\_luad.txt

nationwidechildrens.org\_clinical\_patient\_luad.txt

nationwidechildrens.org\_clinical\_radiation\_luad.txt.

As mentioned previously, the downloaded GCC-Data folder, contains 35 txt files, each referring to one patient. The first column contains all Genes from Microarry chip, the second column are expression levels corresponding to each gene.

1. Methods
   1. Read in all data

The goal here was to read all files in the GCC folder and join them so that at the end we get one single file, which contains one Gene column and 35 Gene Expression Level of all 35 patients.

*// function to list all files in a directory***def** getListOfFiles(dir: String): List[File] = {  
 **val** d = **new** File(dir)  
 **if** (d.exists && d.isDirectory) {  
 d.listFiles.filter(\_.isFile).toList  
 } **else** {  
 *List*[File]()  
 }  
}

*// extract the filename of absolute path to file***val** files = getListOfFiles(*dataGCCFilePath*)  
**val** filenameArray = files.toString.split(**","**)  
**val** sizeOfFilenameArray = filenameArray.size

*//val items = getDataSetFile(env,filenameArray(1)).as('firstFileCol1, 'firstFileCol2)***val** firstFile = *getDataSetFile*(env,filenameArray(1)).as(**'firstFileCol1**, **'firstFileCol2**)  
**for** (i <- 2 to filenameArray.size-2) {  
 *// Read a file but only includes the 1st, 2nd column - returns DataSet[MyLineitem]* **val** CurrentFile = *getDataSetFile*(env,filenameArray(i)).as(**'col1**, **'col2**)  
  
 **val** items =  
 firstFile.join(CurrentFile)  
 .where(**'firstFileCol1** === **'col1**)  
 .select()  
  
}  
items.writeAsCsv(**"file://path"**, **"\n"**, **"\t"**)

This method has been fail due to a .writeAsCsv() method problem.

Somehow this method doesn’t recognize items variable out of the fore

loop…..

* 1. Machine Learning

*“perform a number of transformations to that data, and then output the transformed data, either to be used as the input (features) of a predictor function, such as a learning model, or just output the transformed data themselves, to be used in some other task.”*

[*Supervised learning*](https://en.wikipedia.org/wiki/Supervised_learning)*: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that*[*maps*](https://en.wikipedia.org/wiki/Map_(mathematics))*inputs to outputs.*

*Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:*[*[3]*](https://en.wikipedia.org/wiki/Machine_learning#cite_note-bishop-3)*:3*

*In*[*classification*](https://en.wikipedia.org/wiki/Statistical_classification)*, inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one (or*[*multi-label classification*](https://en.wikipedia.org/wiki/Multi-label_classification)*) or more of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".*

Regarding