Cancer Stratification Based on Genomics and Transcriptomics Data

Jake Leigh Watson

University of Glasgow
Canon Medical Research Europe

This dissertation is submitted for the degree of
Master of Science

Student Number: 2597296

August 2021

Table of contents

1 Introduction **1**

2 Methods **3**

2.1 Data Acquisition and Parsing 3

2.2 Data Cleaning, Filtering and Feature Construction 3

2.3 Machine Learning 4

2.3.1 Imbalanced Data 4

2.3.2 Cross Validation 5

2.3.3 Feature Engineering / Selection 5

References **7**

Chapter 1

Introduction

1. Start by discussing the rise of next generation sequencing, in relation to genomics and transcriptomics and how this increased data complexity in biology. Discuss the concurrent rise in bioinformatics and it's dependency on machine learning - potentially giving a few examples where machine learning has been used with cancer omics data.

2. Identify the 'curse of dimensionality' and how the high dimensional nature of omic data produces problems arising from the large amount of measurements that can be made (e.g. 20,000 genes in one genomic experiment) compared to the (usually) relatively small amount of biological replicates available

3. Briefly discuss the properties of cancer and then go onto discuss the role of isoform switches, eQTLs and protein coding variants in cancer - and how each could affect gene expression. In each case identify the databases containing the data used in the project (e.g. PancanQTL for eQTLs).

4. To finish introduction state how this project will look at reducing dimensionality, using the publicly available databases (mentioned above), to hopefully enable a machine learning approach to cancer stratification with reduced data complexity and dimensionality. Mention how in this case the features will be genes, with values analysed being FPKM gene expression.

   Also potentially include a hypothesis, although not fully certain if required in a computational based project?? ... If required, the null hypothesis could be along the lines of - If no significant change in gene expression is seen between the FPKM values for tumour and normal samples, in relation to genes associated with isoform switches, eQTLs and protein coding variants, then that gene does not contribute to the development of cancer - with the models classification accuracy determining the level of impact isoform switches, eQTLs and protein coding variants have on cancer development.

Chapter 2

Methods

2.1 Data Acquisition and Parsing

Initially detail how isoform switch data from the iso-kTSP database (Sebestyén *et al.*, 2015) is only available for 9 cancer types => restricting our subsequent focus onto these 9 cancer types, with one cancer type rejected on the basis of limited sample numbers.

Detail how each set of data was downloaded and acquired:

- FPKM and MAF data - TCGABiolinks R-based script. With protein coding varaints determined from MAF file by filtering for genes where the VEP was 'high' or 'moderate' indicating variants that are likely to disrupt and alter protein behaviour.
- Isoform switch data - iso-kTSP database direct download with python script parsing.
- eQTL data - PancanQTL direct download with python script parsing.

2.2 Data Cleaning, Filtering and Feature Construction

- Clean the FPKM files to only contain protein coding genes and remove (or rename) any duplicated genes. FPKM files then transposed to contain samples as rows and genes as columns - with additional target 'Cancer' column denoting if the sample/patient had (value = 1) or did not have (value = 0) cancer.
- Indicate that both the isoform switch and eQTL data was acquired using hg19, although the FPKM and MAF data was acquired using hg38. Additionally both isoform switches and eQTL data lacked Ensembl gene IDs => converted provided gene symbols and Entrez IDs, from hg19 to hg38, using biomaRt R-based script which returned the relevant Ensembl IDs for each gene - with the Ensembl IDs subsequently used throughout to filter the FPKM dataframes in relation to the associated isoform and eQTL genes.

- Potentially discuss the production of the difference plots used to assess the viability of filtering in relation to isoform switches and certain percentages of eQTL and protein coding variant genes. i.e. is there sufficient evidence to suggest these genes FPKMs are significantly up or down regulated between tumour and normal samples. Also discuss how change in FPKM between tumour and normal samples did not significantly change when only looking at matched samples - justifying our approach using non-matched samples and potentially signifying the reliability of how the non-matched approach and models used could be extrapolated to look at other/future samples (maybe mention parts of this in results and discussion though).
- Discuss any instances where the genes associated with isoform switches, protein coding variants and cis and trans eQTLs were combined to construct a viable set of features / genes for modelling e.g. in the case of PRAD where only one isoform switch gene is provided => need to combine with eQTLs...
- Detail the final filtered FPKM data frames that were passed through the model

2.3 Machine Learning

Will test a few machine learning classifiers, potentially including:

- Random Forests
- Support Vector Machine
- Deep Neural Networks
- XGBoost Classifier (May not perform as well as Random Forest due to 'noise' in FPKM data, but does contain a scale weight parameter for imbalanced data).

In each case train test split will be utilised to split the data into training and validation data ($\sim$ 30 (validation) : 70 (training) split).

Performance metrics for each model could include root mean squared error, confusion matrix (displayed in heatmap style) and classification report detailing the accuracy, recall, precision and f1 score. However, the area under the receiver operating characteristic (AUROC) score could be better performance metric for classification based models.

There are no categorical variables to handle and there should be no missing data; however, check for missing data before running through model.

2.3.1 Imbalanced Data

Note that each cancer type has an imbalance, with far more tumour samples than normal samples (generally > 10:1 ratio). => Will need to account for imbalance potentially by setting the 'class/sample weight' parameter in model or by 'resampling' imbalanced data ... not sure if there are different or preferred methods?

2.3.2 Cross Validation

Use to measure model quality on different subsets of training data - important due to the general small size of the data and training data sets => by using cross validation, can essentially 'increase' the training data set, reduce noise and get a more accurate account of model quality. Cross validation easier to perform using sklearn.pipeline .. don't think the FPKM data has to passed through scaling preprocessing since FPKM already normalised?

2.3.3 Feature Engineering / Selection

Want to reduce down the number of features / genes so that only the genes that have a strong influence on the target cancer variable are utilised by the model => increase models predictive capabilities, reduce model time and increase the general interpretability of the data/model.

Potential approaches to feature engineering:

- Initially, could simply remove 'quasi constant' genes which essentially have the same FPKM value across the vast majority (99%) of the data set. This could be done using sklearn VarianceThreshold with the threshold value set at 0.01.
- Mutual information and ANOVA f-test statistics seem a popular approach feature selection for models with numerical inputs / features (FPKM values in our case) and classification target variable ((binary tumour:1 or normal:0 in our case)
- Random Forests also come with a feature_importances_ method which can be used to rank the importance of the input features / genes.
- Maybe best approach would be to use feature selection algorithms as utilised in the CPEM paper. CPEM looked at extra tree-based feature selection - which worked well with random forest classifier. Also looked at LASSO and LSVC selection methods - which both worked well on the neural network, with LSVC displaying the higher classification accuracy. Note that LSVC can be implemented as a preprocessing step inside sklearn.pipeline.

References

E. Sebestyén, M. Zawisza, and E. Eyras, Nucleic Acids Research **43**, 1345 (2015).