Genome mining using machine learning techniques

Peter Wlodarczak, Prof. Jeffrey Soar, Dr. Mustafa Ally

University of Southern Queensland, West Street
Toowoomba Qld 4350, Australia
wlodarczak@gmail.com, Jeffrey.Soar@usq.edu.au,
Mustafa.Ally@usq.edu.au

Abstract. A major milestone in modern biology was the complete sequencing of the human genome. But it produced a whole set of new challenges in exploring the functions and interactions of different parts of the genome. One application is predicting disorders based on mining the genotype and understanding how the interactions between genetic loci lead to certain human diseases.

However typically disease phenotypes are genetically complex. They are characterized by large, high-dimensional data sets. Also usually the sample size is small.

Recently machine learning and predictive modelling approaches have been successfully applied to understand the genotype-phenotype relations and link them to human diseases. They are well suited to overcome the problems of the large data sets produced by the human genome and its high-dimensionality. Machine learning techniques have been applied in virtually all data mining domains and have proven to be effective in BioData mining as well.

This paper describes some of the techniques that have been adopted in recent studies in human genome analysis.

Keywords: Genome wide prediction, machine learning, cross validation, predictive medicine

1 Introduction

A central challenge in systems biology and medical genetics is to understand how interactions among genetic loci contribute to complex phenotypic traits and human diseases [1]. A major goal of medical genetics is to determine a set of genetic markers which, combined with some common risk factors, can be used to predict an individual's susceptibility to develop certain diseases. Genetic markers can be used to study the relation between inherited diseases and its genetic cause. Genetic markers are genes or DNA sequences used to identify the presence of specific genes or gene defects.

Typically the number of markers *p* is large and the sample size *n* is small: “large *p* small *n* problem”.

Disease phenotypes are usually genetically complex. A natural first step to tackling these formidable tasks is to construct an annotation of the genome, which is to (1) identify all functional elements in the genome, (2) group them into element classes such as coding genes, non-coding genes and regulatory modules, and (3) characterize the classes by some concrete features such as sequence patterns [3].

Common approaches in genome analysis are statistical methods such as whole-genome regression models and association testing. These methods regress phenotypes on thousands of markers concurrently. They have been improved using for instance shrinkage or regularization in Single Nucleotide Polymorphism (SNP) regression models, nevertheless they are prone to serious over-fitting problems due to the ratio between the number of markers and the available phenotypes. Use of sequencing technologies places further challenges because several million of variants per individual may need to be taken into account in predictive models [2].

In recent studies Machine Learning (ML) techniques have been applied since they are capable to deal with the high dimensionality problem in an efficient way. By its very nature, genomics produces large, highdimensional datasets that are well suited to analysis by machine learning approaches [3]. ML and predictive modelling has proven to be an effective way for mining genotype-phenotype relationships. Learning techniques are efficient in solving complex biological problems due to characteristics such as robustness, fault tolerances, adaptive learning and massively parallel analysis capabilities, and for a biological system it may be employed as tool for data-driven discovery [4].

In recent studies ML techniques in genome analysis have been used for risk prediction and treatment of cancer [5,6,7], multiple sclerosis [8], Alzheimer’s disease [9,10], diabetes [11] and Legionnaires’ disease [12] to name a few.

This paper gives an overview of machine learning techniques used in genome wide prediction (GWP).

1.1 Machine learning for genome analysis

Machine learning (ML) is a branch of Artificial Intelligence (AI). The basic idea is to construct a mathematical model based on historic data and apply it to new, unseen data. Human learning can be defined as making better decisions in the future based on past experiences. Since computers have no experiences they learn from data. A ML model learns from historic data to make predictions on new data, for instance predict the susceptibility to certain diseases. ML techniques are particularly useful when the amount of data is too large to be handled manually or when expert knowledge is incomplete.

ML techniques are divided into supervised, unsupervised and semi-supervised methods. For supervised learning algorithms, a given data set is typically divided into two parts: training and testing data sets with known class labels [13]. Supervised ML are used for classification, for instance to classify regions in the genome into regulatory, transcribed and functional sequence regions.

Unsupervised ML methods are used for clustering, when the class label is not known. They are adopted to understand the underlying structure of data. Genomic loci are naturally clustered together according to their similarities, the distribution of features. Clustering can be exclusive, if an instance falls into exactly one group, overlapping, if some instances belong into two or more groups, or it may be probabilistic. Semi-supervised techniques are employed when small amounts of labelled data and large amounts of unlabelled data exists. Usually in genome analysis supervised methods are employed since the class labels are known, for instance protein coding regions and regulatory regions.

Ultimately we want to find a decision function *f*, that classifies genome loci into labels $X=\{x_1, x_2, \ldots x_n\}$, such that $f: X \rightarrow \{E, NE\}$, predicts if a loci is for instance a enhancer *E*, or not *NE*. This is a binary classification problem since we have two class labels. *f* is called classifier. If there are more than two class labels, it is a multi-class classification problem. For instance if we want to identify regions such as silencers and insulators in addition to enhancers and promoters. If we do not have a set of pre-defined, discrete values but continuous values, we have a regression problem and *f* is called a regressor or estimator. Here we focus on classification, since we want to associate DNA sequences with specific element classes. However, it should be noted that many classifiers output the probability *Pr*, that a region x_i with corresponding label y_i belongs to class *j*:

$$\Pr(x_i | y_i = j) \tag{1}$$

To find a suitable *f* we need to:

1. Decide on an appropriate model for *f*, possible models are artificial Neural Networks (aNN), naïve Bayes classifier, boosting or Support Vector Machines (SVM)
2. Find a set of training data, for instance a set of regulatory regions that contain enhancers and promoters
3. An estimate for the classification accuracy such as a loss function

For task 1, to decide on the appropriate model, several models are trained and the one that predicts the label of a region most accurately is chosen. Experience shows that no single machine learning scheme is appropriate to all data mining problems [14]. To decide which scheme is the most appropriate we need a means of evaluating the trained model. Since performance on the training set is no good indicator of performance on unseen data, task 3, evaluating the model, is tricky, especially when the set of training data from task 2 is small. We need to be able to predict the performance of the model on future data and compare it to the estimated performance of the other trained models. Cross-validation is one of the most popular evaluation methods in limited-data situations. It will be described later in this paper.

1.1.1 Artificial neural networks.

Artificial neural networks (ANN) can act as universal approximators of complex functions because of their capability of learning linear or nonlinear relationships between predictor variables and responses, including also all sorts of interactions between explanatory variables [2]. There are many types of ANN, but in GWP usually multilayer ANN are used. Multilayer ANN consist of an input layer, one or more middle layers, called hidden layers and an output layer. The input layer is given for instance SNP genotype codes, pedigree and nuisance variables as input.

A multilayer ANN consists of preceptrons, the neurons, which are interconnected through weighted connections, the axons. The basic idea of a perceptron is to find a linear function *f* such that:

$$f(x) = w^T x + b \tag{2}$$

where f (x) > 0 for one class and f (x) < 0 for the other class, and $w = (w_1, w_2, \ldots, w_m)$ is the vector of coefficients (weights) of the function, and *b* is the bias. During training the weights and bias are adjusted until prediction accuracy is converging.

ANN are prone to over-fitting. Over-fitting occurs when the model describes the noise or random error instead of the underlying data. An over-fitted model would have good training accuracy but poor testing accuracy [3]. Two techniques that are widely used for overcoming over-fitting in ANN models are Bayesian regularization and cross-validated early stopping [2].

In recent studies ANNs have been used to study gene-gene interactions for biomarkers [18], to model gene-environment interactions [19] and to find splice sites in human [20].

1.1.2 Naïve Bayesian classifiers

Naïve Bayesian classifiers are a family of simple, probabilistic classifiers based on the Bayes theorem. The term naïve refers to the fact that there is a strong independence assumption between the features. The naïve Bayesian classifiers builds a probabilistic model of the features and predicts the classification of new, unseen examples. Naïve Bayes can use kernel density estimators, which improve performance if the normality assumption is grossly incorrect; it can also handle numeric attributes using supervised discretization [14].

The Bayesian classifier has been applied in analyzing effectors in the genome to detect the causative agent of Legionnaires' disease with an accuracy of more than 90% [12]. It has been adopted for analyzing of single nucleotide polymorphism for detecting Alzheimer's disease. Single Nucleotide Polymorphisms (SNPs) are a specific class of genomic variation responsible for about 90% of human variability [6]. A high classification accuracy has been achieved for the detection of Alzheimer's disease [9].

1.1.3 Boosting

Boosting is an ensemble learning method. It is often advantageous to take the training data and derive several different training sets from it, learn a model from each, and combine them to produce an ensemble of learned models [14]. By combining several weak learning schemes it is often possible to create a very strong one. If several schemes have been trained, it can be advantageous not to choose the best performing one but to combine them all.

Boosting combines models that complement each other. Boosting has several characteristics. The models are of similar type such as decision trees. Boosting is iterative, each new model is build based on the performance of the previous model. New models are trained in a way that it performs well for instances that were incorrectly handled by previous models. Also models are weighted by their confidence and are not treated equally.

Boosting has been applied to GWP in chicken, swine and dairy cattle with similar or better predictive ability than Bayes A or G-BLUP [2].

1.1.4 Support Vector Machines

The basic idea behind Support Vector Machines is to find a function that can be expressed in terms of a few support vectors and can be applied to non-linear problems. It uses linear models to implement non-linear regressions by mapping the input space into a higher dimensional feature space using kernel functions [2]. Support Vector Machines (SVM) create a feature space or vector space defined by a similarity matrix (kernel) and create a hyperplane, an affine decision surface, separating the examples, for instance enhancers and promoters, and maximizes the distance from it from the closest training samples. SVMs operate by finding a hyper surface in the space of gene expression profiles, that will split the groups so that there is largest distance between the hyper surface and the nearest of the points in the groups [17].

If the training data is linearly separable, then a pair (w, b) exists such that
$w^T x_i + b \geq 1$, for all $x_i \in P$
$w^T x_i + b \leq -1$, for all $x_i \in N$

with the decision rule given by:

$$\int_{w,b}(x) = \mathrm{sgn}(w^T x + b) \tag{3}$$

where w is termed the weight vector and b the bias (or $-b$ is termed the threshold) [15].

SVM have been primarily used for classification, but they can also be used for regression. SVM have been successfully applied for instance in predicting cancer-causing missense variants and achieved a 93% overall accuracy [6] and for analyzing gene-gene interactions by investigating SNPs for Type 2 diabetes mellitus (T2D) and achieved an accuracy of more than 70% [11].

1.1.5 Cross-validation

Often in genome sequencing the sample size is small and a major challenge is to estimate accurately the prediction performance of a ML model. It is in fact not very hard to find genetic features that can almost perfectly fit to a small training set but fail to generalize to unseen data, a phenomenon known a s model overfitting [1]. To overcome this problem the model has to be tested against an independent data set not used in training. Cross-validation is an effective method for predicting the performance in a small data situation. *n*-fold cross-validation divides the data set randomly into *n* folds. *n* -1 folds are used for training, one fold, the holdout set is used for testing. This process is repeated n times such that each of the *n* folds has been used once as holdout set for testing.

To avoid a class to be overrepresented in one data set, random sampling should be done in a way such that each class is represented properly in the training and testing set. This procedure is called stratification, and we might speak of stratified holdout [14]. However stratification is only a primitive safeguard against uneven representation and often leads to over-optimistic results. A more effective method to mitigate this bias is to repeat all iterations with different random samples. For each iteration the error rate is calculated and averaged at the end of all iterations to yield the overall error rate.

Usually *n* is between 5 and 10. Extensive tests on numerous different datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up [14].

Cross-validation has been used based on the analysis of SNP. Cross-validation in combination with the Bayes classifier achieved a high accuracy for the detection of Alzheimer's disease [9] and combined with SVM for T2D detection [11].

2 Discussion

Genome sequencing remains a challenging task due to the complexity, high-dimensionality and large size of the human genome. The number of sequences available is increasing exponentially [16]. There are challenges inherent to the data, the genome, and challenges originating from the analysis methods.

Complex diseases such as cancer are multi-factorial in nature and interactions among genetic loci, epistatic interactions, are believed to be a major contributing factor. There exist also increasingly complex interactions between genetic variants and environmental factors that may contribute to the disease risk on an individualized basis [1]. This suggests that a one variant at a time approach might not be expedient and a more holistic approach should be pursued.

It is often difficult to define the exact span of a genomic region. Biologically it could be fuzzy to define exactly where a functional element starts and ends (as in the case of an enhancer), and even if the span could be formally defined (as in the case of an RNA transcript), it is usually not known prior to machine learning [3].

The class labels of neighboring genomic regions are not independent. For example, if a base is within an intron, the next base should be either within an intron or a splice site [3].

Several aspects of ML influence the performance of ML techniques and need to be considered when evaluating the classifier. Different ML methods handle redundancy dramatically differently. For instance the naïve Bayes classifier assumes the input features to be independent from each other for each class, otherwise the redundant features could have an undesirable stronger influence on the predictions from the non-dependent ones.

In genome analysis the amount of samples, the training data, is often limited, “large *p* small *n* problem”. There might not be enough training samples to create a model that accurately predict class labels. The situation is aggravated if irrelevant features to the problem are included. Also there is a high risk of over-fitting. Semi-supervised ML techniques are a practical way to alleviate the problem.

Genomic features often combine different data types, for instance the frequency of a feature might be represented by a numeric value and the raw sequence of it as text string. One systematic approach to handling mixed data types is to turn each type of data into a numerical similarity matrix between the input regions before integrating them [3].

Currently there are still likely undiscovered genomic element classes given the rapid discovery of new classes (such as many non-coding RNAs (ncRNAs)) in recent years [3]. In ML terms this means that purely supervised techniques are not suitable. Combining several machine learning methods, a technique called ensemble learning, has been proposed. For instance boosting or random forests are ensemble learning techniques that have been used in genome mining [3,5].

To train ML models in a way that yields suitable predictors, each class has to be properly represented in the training and in the test data set. This process is called stratification. Since genome sample size is often small, having a large enough stratified holdout for testing can be difficult. Furthermore negative examples are crucial for ML, however often there is limited availability. They are irrelevant for the problem but they are crucial for training the model. Availability of more negative samples would be highly desirable and finding effective techniques for producing them could be an area of further research.

Feature selection is a key for developing effective predictive models for GWA and implementing scalable algorithms for genetic feature selection is crucial for successful ML modelling. Further research in the area of computational genetic feature selection is essential for building scalable ML-based predictive models.

Due to the complexity of the problem, more holistic approaches “which take into account the complexity of the genotype-phenotype relationships characterized by multiple gene-gene and gene-environment interactions [1]” should be pursued in future research. ML techniques are well suited for complex problems and large data sets and might shed more light into the genotype-phenotype relation as well as other influencing factors that lead to complex diseases.

3 Conclusions

ML techniques have been adopted in virtually all data analysis domains. ML techniques can generate highly complex models. This is particularly advantageous when the data to be analyzed is also complex such as for instance eukaryotic genes.

In recent years a major paradigm shift in disease treatment towards personalized medicine strategies and a network pharmacology, a paradigm which provides a more global understanding of drug action in their context of biological networks and pathways, has been surfacing. Genomic profiling might lead to tailored personalized treatments and help predict a patient's susceptibility for certain diseases and induce early treatment. ML techniques have shown to be well suited to enable these new paradigms. In this paper some of the more popular techniques were described, but many other techniques have been used in genome analysis. For instance ML techniques based on hidden Markov models have become a one of the most popular methods for computational gene finding.

ML techniques are very suited to handle multiple, heterogeneous features. Depending on the mathematical form of the model, the different features can be integrated in ways from linear combinations to highly nonlinear ones [3].

ML techniques can be used to probe drugs for their efficacy against cancer in silico. In silico methods to accurately predict the effectiveness of drugs based on the molecular making of tumors (i.e. genome, transcriptome) would be a major milestone towards personalized therapies for cancer patients based on molecular biomarkers [7]. Machine learning-based predictive modeling approaches are well-powered to make the most of the exciting functional and genetic screens toward revealing hidden genetic variants and their interactions behind cancer and other complex phenotypes [1].

4 References

1. Okser, S., T. Pahikkala, and T. Aittokallio, Genetic variants and their interactions in disease risk prediction - machine learning and network perspectives. BioData Mining, 2013. **6**(1): p. 5.
2. González-Recio, O., G.J.M. Rosa, and D. Gianola, Machine learning methods and predictive ability metrics for genome-wide prediction of complex traits. Livestock Science, 2014. **166**(0): p. 217-231.
3. Yip, K., C. Cheng, and M. Gerstein, Machine learning and genome annotation: a match meant to be? Genome Biology, 2013. **14**(5): p. 205.
4. Patel, M., et al., An Introduction to Back Propagation Learning and its Application in Classification of Genome Data Sequence, in Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012, B.V. Babu, et al., Editors. 2014, Springer India. p. 609-615.
5. Vanneschi, L., et al., A comparison of machine learning techniques for survival prediction in breast cancer. BioData Mining, 2011. 4(1): p. 12.
6. Capriotti, E. and R.B. Altman, A new disease-specific machine learning approach for the prediction of cancer-causing missense variants. Genomics, 2011. 98(4): p. 310-317.
7. Menden, M.P., et al., Machine Learning Prediction of Cancer Cell Sensitivity to Drugs Based on Genomic and Chemical Properties. PLoS ONE, 2013. 8(4): p. 1-7.
8. Guo, P., et al., Mining gene expression data of multiple sclerosis. PloS one, 2014. 9(6): p. e100052.

9. Granados, E.A.O., et al. Characterizing genetic interactions using a machine learning approach in Colombian patients with Alzheimer's disease. in Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on. 2013.
10. Scheubert, L., et al., Tissue-based Alzheimer gene expression markers-comparison of multiple machine learning approaches and investigation of redundancy in small biomarker sets. BMC Bioinformatics, 2012. 13(1): p. 266.
11. Ban, H.-J., et al., Identification of Type 2 Diabetes-associated combination of SNPs using Support Vector Machine. BMC Genetics, 2010. 11(1): p. 26.
12. Burstein, D., et al., Genome-Scale Identification of Legionella pneumophila Effectors Using a Machine Learning Approach. PLoS Pathogens, 2009. 5(7): p. 1-12.
13. Tretyakov, K., Machine Learning Techniques in Spam Filtering, in Data Mining Problem-oriented Seminar, U.o.T. Institute of Computer Science, Editor. 2004: Estonia. p. 19.
14. Witten, I.H., E. Frank, and M.A. Hall, Data Mining. 3 ed. 2011, Burlington, MA, USA: Elsevier.
15. Kotsiantis, S.B., Supervised Machine Learning. Informatica, 2007. 31: p. 19.
16. Larrañaga, P., et al., Machine learning in bioinformatics. Briefings in Bioinformatics, 2006. 7(1): p. 86-112.
17. Jauhari, S. and S.A.M. Rizvi, Mining Gene Expression Data Focusing Cancer Therapeutics: A Digest. Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 2014. 11(3): p. 533-547.
18. Tong, D.L., et al., Artificial Neural Network Inference (ANNI): A Study on Gene-Gene Interaction for Biomarkers in Childhood Sarcomas. PLoS ONE, 2014. 9(7): p. 1-13.
19. Gunther, F., I. Pigeot, and K. Bammann, Artificial neural networks modeling gene-environment interaction. BMC Genetics, 2012. 13(1): p. 37.
20. Abo-Zahhad, M., et al., Integrated Model of DNA Sequence Numerical Representation and Artificial Neural Network for Human Donor and Acceptor Sites Prediction. International journal of information technology and computer science, 2014. 6(8): p. 51-57.