Artificial Intelligence and Machine Learning in Bioinformatics

Kaitao Lai, Natalie Twine, and Aidan O'brien, CSIRO, North Ryde, NSW, Australia
Yi Guo, Western Sydney University, Penrith, NSW, Australia
Denis Bauer, CSIRO, North Ryde, NSW, Australia

Introduction

Machine Learning is defined as a computer science discipline where algorithms iteratively learn from observations to return insights from data without the need for programming explicit tests. Machine Learning peaked in Gartner's Hype Cycle for Emerging Technologies in 2017 and received substantial attention in its broader context of Artificial Intelligence (AI) with Google AlphaGo and Tesla Autopilot showcasing the advanced decision-making ability of such techniques.

The recent technological leaps in this space are due to two enabling trends. Firstly, computing is increasingly being shifted into the cloud, which enables the money that has traditionally been spent on hardware to now be invested in computing. This rent-based access results in the use of more powerful and specialized hardware. As machines learn through an iterative process of evaluating and updating internal evaluation measures, performing this on more appropriate systems removes traditional limitations and allows for more sophisticated methods to be trained. Secondly, the digital revolution has seen a dramatic increase in data collection about almost every aspect of life. As the ability of a machine to gain generalizable insights from presented examples is directly correlated with the dataset size, the recent increase has enabled the training of very sophisticated Machine Learning models.

These datasets are not only growing vertically, by capturing more events, but also horizontally by capturing more information about these events. The challenge of "big" and "wide" data is especially pronounced in the biomedical space where, for example, whole genome sequencing (WGS) technology enables researchers to interrogate all 3 billion base pairs of the human genome. With an expected 50% of the world's population likely to have been sequenced by 2025, the resulting datasets may surpass those generated in Astronomy, Twitter and YouTube combined (Stephens *et al.*, 2015). Machine Learning approaches are hence necessary to gain insights from these enormous and highly complex modern datasets. Here we will discuss applications in sequence annotation, disease gene association, and drug discovery.

Specifically, for analysing next-generation-sequencing data, Machine Learning has been applied to analyse RNA sequencing (RNA-seq) expression data, data from chromatin accessibility assays, such as DNase I hypersensitive site sequencing (DNase-seq), or chromatin immunoprecipitation followed by sequencing (ChIP-seq), and data on histone modification or transcription factor binding, to name a few. For more details, please see the excellent review by Libbrecht and Noble (2015).

Machine Learning approaches have been applied in life science fields well before the genomic revolution. For example, Machine Learning algorithms can learn to recognize patterns in DNA sequences (Libbrecht and Noble, 2015), such as pinpointing the locations of transcription start sites (TSSs) (Ohler *et al.*, 2002), identifying the importance of junk DNA in the genome (Algama *et al.*, 2017), and identifying untranslated regions (UTRs), introns and exons in eukaryotic chromosomes (Picardi and Pesole, 2010).

Enhancing the annotation of genomic regions can be achieved by using Machine Learning to combine datasets for functional gene annotation. Here, the input data can include the genomic sequence, gene expression profiles across various experimental conditions or phenotypes, protein-protein interaction data, synthetic lethality data, open chromatin data, and ChIP-seq data on histone modification or transcription factor binding (Libbrecht and Noble, 2015). Specifically transcription factors, as the master regulators of gene expression, have received attention and models have built for profiling their binding behaviour (Kummerfeld and Teichmann, 2006).

Moving up from two-dimensional sequence space, Machine Learning has also found application in predicting the 3D structure of proteins and RNA molecules from sequence, the design of artificial proteins or enzymes, and the automated analysis and comparison of biomacromolecules in atomic detail (Hamelryck, 2009).

Moving from descriptive applications to interpretative areas, Machine Learning has been used to gain insights into the molecular mechanisms of genetic diseases and susceptibilities. This is because of the growing awareness that complex interactions among genes and environmental factors are important in common human disease etiology. Traditional statistical methods are not well suited for identifying such interactions, especially when interactions occur between more than two genes, or when the data are high-dimensional (many attributes or independent variables). Machine Learning algorithms, including artificial neural networks (ANNs), cellular automata (CAs), random forests (RF), and multifactor dimensionality reduction (MDR), have been used for detecting and characterising susceptibility genes and gene interactions in common, complex, multifactorial human diseases (Mckinney *et al.*, 2006). However, the traditional implementations of these technologies reach their limit with modern dataset sizes, which we will discuss in the context of VariantSpark (O'Brien *et al.*, 2018) in Sections Random Forests (RF) and Feature Selection.

Machine learning techniques have been used to elucidate the functional interactions of genes by modelling and analysing gene regulatory networks (Schlauch *et al.*, 2017; Ni *et al.*, 2016). This machine learning discipline has helped answer biological questions about how molecular phylogenetics and evolution represents at whole-genome level, about how to identify protein biomarkers of diseases, and for disease-gene association discovery (Leung *et al.*, 2016).

Finally, arriving at applications in drug discovery, Machine Learning has been specifically applied in proteomics, which is the large-scale analysis of proteins, the main targets of drug discovery. Proteomics research provides applications for drug discovery,

including target identification and validation, identification of efficacy and toxicity biomarkers from readily accessible biological fluids, and investigation into the mechanisms of drug action or toxicity (Blunsom, 2004). Drug discovery is a continuous process that applies a variety of tools from diverse fields. Proteomics, genomics and some cellular and organismic approaches have been developed to accelerate the process (Abu-Jamous *et al.*, 2015a).

This article summarizes two main Machine Learning approaches: supervised learning and unsupervised learning. Apart from these predictive models, which aim to provide one result per sample, we also discuss generative methods, which are aimed at training a model to generate new data with similar properties to the training data. Lastly, we devote a section to Deep Learning, as one of the most recent developments in Machine Learning. However, we first discuss training regimes and quality control, for a more detailed description of this we refer the reader to the review by Libbrecht and Noble (2015).

Dataset for Machine Learning and Performance Measure

Choosing the right dataset is important for building a robust machine learning model. Typically, the dataset is divided into training set, validation set and test set. The training set is applied to build the models; the validation set is adopted to evaluate the generalization error of the final selected model and for optimizing the parameters; the test set is used as the final prove of the model's performance and used to report the generalization ability to unseen data.

Overfitting

In the machine learning task, a learning model is trained on a set of training data, but then it is applied to make predictions on new dataset. We need to consider the general situation that if the knowledge and data we have are not sufficient to completely determine the correct classifier. We take the risk of just constructing a classifier (or parts of this classifier) that is not grounded in reality, and is simply encoding random data points in the dataset (Domingos, 2012). The objective is to maximize its predictive accuracy on the new data set, and not necessary its accuracy on the training dataset. In fact, if we try our best to identify the very best fit to the training dataset, we run the risk of fitting the noise in the data by memorizing various peculiarities of the training data rather than finding a set of general predictive rules. This problem is called "overfitting" (Dietterich, 1995). Overfitting generally happens when the gap between the training and the test error is large (Valiant, 1984).

Domingos provided one way to understand overfitting by decomposing generalization error into bias and variance (**Fig. 1**). Consider a random guesser that chooses a number from a set of positive numbers irrespective of any input. Such a method will have high variance and high bias. Now if we limit the guesser to only returning the same number, it will have a low variance but is

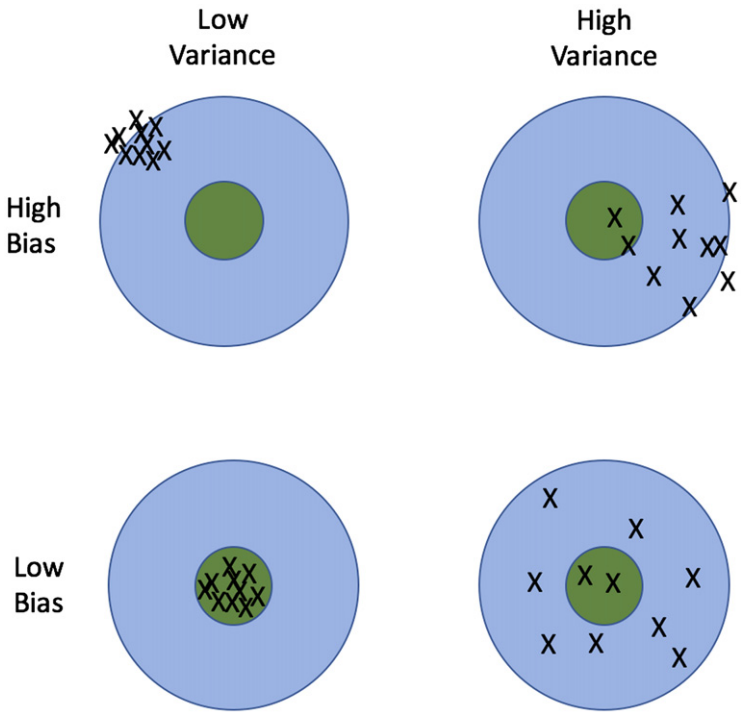


Fig. 1 Bias and variance in overfitting. Reproduced from Domingos, P., 2012. A few useful things to know about machine learning. Communications of the ACM 55, 78–87.

still biased. Conversely, we can reduce the bias by allowing the function to return positive and negative values. However, if we implement a function that actually takes the input into consideration and accurately predicts the target label, we can reduce both variance and bias.

Training Dataset

The most important consideration for generating a training dataset is to collect samples that span the entire problem space and represent predicted classes or values equally. Typically, it is easy to select samples for the commonly occurring classes or values (majority class). However, a classifier trained on such an imbalanced dataset would likely only predict the majority class. Consider a classifier for lung cancer with 351 patients, of which 95 had reoccurring cancer while 256 were cured. A model predicting no recurrence of lung cancer for all patients would reach an accuracy of 72.93% (256/351)*100. Despite this deceptively high classification accuracy it would tell 95 patients that their lung cancer was not going to reoccur (False Negatives), which is likely useless for a clinical setting. If the model needs to also accurately predict the minority class then it must have an equal representation of such samples in the dataset.

As collecting more of the minority class can be difficult, and presenting multiple copies of the same sample can lead to artefacts, undersampling the majority class is a common approach for processing imbalanced datasets. However, undersampling data has also drawbacks especially if the dataset is small to begin with. More specialized approaches have been developed taking the mechanisms in the different Machine Learning methods into account, e.g., for SVM (Akbani *et al.*, 2004).

Besides the equal representation of the prediction label (class or value), also important is the unbiased representation of samples. For example, the scenario of including the same sample multiple times can happen involuntarily especially if sample similarity cannot be determined easily (e.g., expression profile). Thorough analysis of the training dataset prior to training a Machine Learning method is hence strongly advisable.

Cross-Validation

Another potential source of bias can be the assignment of samples to training and test data. Any fixed split into training and testing data can lead to bias if the split happens to distribute samples unfavourable, e.g., all minority classes in the test set. N-fold cross-validation is a popular statistical method for randomizing the dataset and creating N equal size partitions: One from the Nth partition is picked for validation/testing, while the remaining N-1 sets are used for training the model then rotating the partitions until all have been used for testing. (Refaeilzadeh *et al.*, 2009).

However, in situations where the similarity between samples is hard to determine, standard N-fold cross validation may not be the right approach. Michiels *et al.* propose a multiple random validation strategy for prediction of cancer from microarrays. This strategy is about identifying a molecular signature (the subset of genes most differentially expressed in patients with different outcomes) in a training set of patients and to evaluate the proportion of misclassifications with this signature on an independent validation set of patients. They applied this strategy based on unique training and validation sets by using multiple random sets to study the stability of molecular signature and the proportion of misclassifications (Michiels *et al.*, 2005).

Measuring Performance of Classification

The use of the area under the receiver operating characteristic (ROC) curve (AUC) are generally used as a performance measure for machine learning algorithm on classification problem (Bradley, 1997). The receiver operating characteristic curve illustrates a binary classifier system as its discrimination threshold is varied. The ROC curve can be plotted with true positive rate (TPR) against the false positive rate (FPR) at various threshold settings from the so-called confusion matrix, which counts the correctly classified (True positives, True negative) as well as incorrect classifications (False positives, False negatives). The area under the ROC curve (AUROC) evaluates the accuracy of the test. An area of 1 represents a perfect test while an area of 0.5 demonstrates a random classifier. The precision-recall curve provides an alternative measure compensating for skewed datasets. The precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that have been retrieved over the total number of relevant instances. Both AUROC and the area under the precision-recall curve (AUPRC) can be measured using N-fold cross validation from training dataset **Table 1**.

Two methods, correlation coefficient and root-mean-square error, are commonly applied to measure the performance of Machine Learning algorithms in regression problems. The correlation coefficient quantifies the relationship between the predicted and observed labels and ranges from -1 to $+1$. A value of 0 means no relationship exists between the two continuous variables, and a value of 1 means a perfect relationship, with -1 denoting a perfect anti-correlation. The Pearson correlation coefficient is

Table 1 Confusion matrix

	Positive	*Negative*
Positive	True positive	False positive
Negative	False negative	True negative

used when both variables are normally distributed, otherwise the Spearman correlation coefficient is more appropriate. The correlation coefficient represents the association, not causal relationships (Mukaka, 2012). The root-mean-square error (RMSE), on the other hand, quantifies the standard deviation of the residuals (prediction errors). The residuals are an indicator of how far from the regression line data points are. RMSE is a frequently used as indicator of the differences between values predicted by a model or an estimator and the values actually observed.

Supervised Learning

Supervised learning can be performed for training datasets where the labels or targets are known for each sample, e.g., the disease status or the category (Maglogiannis, 2007). Think of this as a child-teacher scenario, where the child learns a subject by receiving feedback on the accuracy of the answer he/she gives. Similarly, a supervised model will learn, from the truth provided in the training set, to build a generalized model that can then be applied to predict the labels for new data. The labels can be categorical, such as ethnicity (O'Brien *et al.*, 2018), or continuous, such as survival rates (Shipp *et al.*, 2002). The former case is called classification while the latter is called regression.

The inputs to the model are called features, which are information points that describe the sample, such as an expression or genotype profile. So, a more general statement is that a supervised learning approach identifies a mapping method from one data space (features) to another data space (targets, i.e., labels) (Yang, 2010).

Another subcategory of supervised learning is feature selection. The objective here is to identify the set of features that are most associated with the labels, to either gain insights into the underlying mechanisms (e.g., biological insights) or remove redundant or noisy variables in order to improve accuracy.

In the subsequent sections, we discuss the different methodologies, such as Bayesian networks, trees, random forest, support vector machines, and artificial neural networks in the classification context. However, those technologies can also be applied in regression, which we discuss in a separate section, below.

Classification

Classification is a fundamental task in data analysis and pattern recognition that constructs a classifier, which assigns a class label to an instance with a set of features/attributes. Similarly to categorization (Cohen and Lefebvre, 2005; Frey *et al.*, 2011), classification is a general process for recognizing, differentiating, understanding, and grouping ideas and objects into classes. It has been widely used in computer science, e.g., in natural language processing (Jackson and Moulinier, 2007), prediction, and decision making (Heekeren *et al.*, 2004).

The induction of classifiers from data sets of pre-classified instances, usually called training data, is one of the fundamental tasks in Machine Learning (Stanke and Waack, 2003). The process of modelling from training data, i.e., building up the mapping from observed features/attributes to correctly predict the class from the available data, is called training or learning. Some common classification approaches are Bayesian classification, classification trees, random forest, support vector machines, nearest neighbours, artificial neural networks, ensemble models (combining classifiers), and so on.

Bayesian classification

Bayesian *classification* is a statistical classification that minimizes the probability of misclassification (Devroye *et al.*, 2013). Many Bayesian classification algorithms are common, and they are traceable to a common ancestor (Langley *et al.*, 1992). They come originally from a supervised algorithm, which is simply referenced as a Bayesian classifier, in pattern recognition (Duda *et al.*, 1973), that assigns a simple probabilistic summary for the data; This summary includes the conditional probabilities of the class labels given the attribute values (Langley *et al.*, 1992; Rish, 2001), which is called the posterior distribution. Bayesian classifiers employ models connecting attributes to class labels and incorporate prior knowledge so that Bayesian inference can be utilized to derive the posterior distribution.

Classification trees

Tree-based algorithms form a decision tree for features in which nodes represent a series of decisions which leaves represent the final class labels. Given an observation with many features, the inference is then by passing those values of features through the decision tree from the top layer to a leave where the prediction is made. Models are constructed by recursively partitioning the feature space, and fitting a simple prediction threshold for each partition. The partitioning is represented graphically as a tree (Loh, 2011). Classification trees are very intuitive in the sense that they can be easily visualized and interpreted. They are also easy to train thanks to their simplicity. The depth of the tree and number of splits in each layer are the main parameters used to curb the overfitting problem.

Random forests (RF)

The random forests (RF) method constructs an ensemble of tree predictors, where each tree is constructed on a subset randomly selected from the training data, with the same sampling distribution for all trees in the forest (Breiman, 2001). Random forest is a

popular nonparametric tree-based ensemble Machine Learning approach that merges the ideas of adaptive nearest neighbour clustering with bagging (Breiman, 1996) for effective data adaptive inference. RF can be applied to "wide" data ("large p, small n") problems, and accounts for correlation as well as interactions among features (Chen and Ishwaran, 2012). Data are identified as a fixed number of features which can be binary, categorical or continuous. Searching a good data demonstration is very domain specific and related to available measurements (Isabelle, 2006). In our gene editing CRISPR target sites on-target activity example, the features use the measurement of target site sequences, such as position-independent or position-dependent nucleotides and dinucleotides (Wilson, *et al*., CRISPR Journal accepted).

RF has been used for gene selection and classification of microarray data (Korf, 2004). More recently, RF have been used for predicting CRISPR-Cas9 on-target activities (Wilson *et al*., CRISPR Journal accepted). The RF model, called TUSCAN, is part of the GT-Scan2 suite for predicting target sites for CRISPR/Cas9 genome engineering. TUSCAN, uses sequence information describing the target site, such as global di-nucleotide frequency, or the presence of nucleotides at specific positions, to predict the activity of any given site. In total 621 features describe each site. For its final model TUSCAN performs feature selection to reduce the number of features to the 63 most important features (see Section Measuring Performance of Classification, Feature selection). TUSCAN can predict the activity of 5000 target sites in under 7 seconds, and is up to 7000 times faster than available methods, and is hence suitable for genome-wide screens.

Support vector machine (SVM)

Support vector machines (SVM) are a pattern classification technique proposed by Vapnik *et al.* (Boser *et al.*, 1992). SVM minimizes an upper bound on the generalization error through maximizing the margin between a hyperplane separating the data classes, and the data (Amari and Wu, 1999). The idea is to transform the data that is not linearly separable in its original space to a higher dimensional space where it can be separated by a simple hyperplane.

SVM has been used to identify target sequences in proteins and nucleic acids, for instance to identify SUMOylation sites (Bauer *et al.*, 2010) in proteins, or splice sites (Degroeve *et al.*, 2002) in primary mRNA. An accurate miR-E shRNA (the improved amiRNA backbone short hairpin RNAs) predictor has been developed using a sequential learning algorithm combining two support vector machine (SVM) classifiers trained on judiciously integrated data sets (Pelossof *et al.*, 2017).

A method has been developed using SVM for classification of tissue samples, consisting of ovarian cancer tissues, normal ovarian tissues and other normal tissues, and an exploration of the data for mislabelled or questionable tissue results. The tissue samples include ovarian cancer tissues, normal ovarian tissues and other normal tissues (Furey *et al.*, 2000).

Artificial neural networks

The motivation for neural networks was originally to mimic the working mechanism of the human brain. It is a graph computing model wherein computing units called neurons are organized in layers, and interconnected for passing information to each other (Kruse *et al.*, 2016). The first layer is the input layer which receives the raw data. The last layer is the output layer, which performs the final prediction task, e.g., classification, regression, and so on. The layers in between are hidden layers. To simplify optimization, the neurons in the same layer are not connected. Neural networks with three layers have the capability to approximate any function. However, the determination of the network architecture is not a trivial task, for example, the number of neurons in hidden layers and their activation functions. What about ANNs with no hidden layer?

Classic feedforward neural networks (FFNN) have been applied to predict protein site-directed recombination, which are breakpoint locations in a protein where introducing sequence from a homolog can yield improved activity (e.g., better heat stability) (Bauer *et al.*, 2006). Convolutional neural networks (CNN) uses a sequence of 2 operations, convolution and pooling, repeatedly on the input data. CNN are a subset of FFNN with a special structure, including sparse connectivity between layers and shared weights, which have surpassed conventional methods in modelling the sequence specificity of DNA-protein binding. In a systematic exploration of CNN architectures for predicting DNA sequence binding using a large compendium of transcription factor dataset, CNNs have been implemented as the best-performing architectures by varying CNN width, depth and pooling designs (Zeng *et al.*, 2016). A combination of embedding-based convolutional features (dense real value vectors, including word's feature vector and syntax word embedding) and traditional features has been developed for use with a softmax classifier to extract DDIs from biomedical literature for detecting drug-drug interactions (DDI) (Zhao *et al.*, 2016).

Combined classification approaches

The quality of *de novo* sequence assembly can be improved by Machine Learning methods using comparative features, such as N50 score and percent match, and non-comparative features, such as mismatch percentage and the *k*-mer frequencies, to classify overlaps as true or false prior to contig (a set of overlapping DNA segments) construction (Palmer *et al.*, 2010). A comprehensive evaluation of multicategory classification methods has been performed for microarray gene expression cancer diagnosis. The multicategory support vector machines (MC-SVMs) have been demonstrated as the most effective classifiers in performing accurate cancer diagnosis from gene expression data and outperform other popular machine learning algorithms, such as backpropagation and probabilistic neural networks. The classification performance of both MC-SVMs and other non-SVM learning algorithms can be improved significantly by gene selection techniques (Statnikov *et al.*, 2005).

Microbiology is the study of microscopic organisms in numerous sub-disciplines including virology, mycology, parasitology, and bacteriology. In microbiology, it was formerly necessary to grow pure cultures in the laboratory in order to study an organism. Since many organisms cannot be cultured, this created a cultivation bottleneck that has limited our view of microbial diversity.

Metagenomics provides a relatively unbiased view of the community structure (species richness and distribution) and the functional (metabolic) potential of a community (Hugenholtz and Tyson, 2008). Metagenomic methodologies are recognized as fundamental for understanding the ecology and evolution of microbial ecosystems. The development of approaches for pathway inference from metagenomics data is crucial to connecting phenotypes to a complex set of interactions stemming from a series of combined sets of genes or proteins. The role of symbiotic microbial populations in fundamental biochemical functions have been investigated based on the modelled biochemical and regulatory pathways within one cell type, one organism, or multiple species (De Filippo *et al.*, 2012).

Machine Learning methodologies are often used for supervised classification. Feature representations and selection may improve microbe classification accuracy by producing better models and predictions (Ning and Beiko, 2015). Microbial communities are crucial to human health. Beck *et al.* have tested three Machine Learning techniques including genetic programming (GP) (Moore *et al.*, 2007; Eiben and Smith, 2003), random forests (RFs), and logistic regression (LR) for their ability to classify microbial communities into bacterial vaginosis (BV) categories. Before constructing classification models, the microbes were collapsed into groups, based on correlations, by reducing the number of factors, such as environmental variables, or dynamic microbial interactions, and to increase the interpretability of the classification models. Genetic algorithm uses computational simulations of evolutionary processes to explore highly fit models; RF is efficient but may not be as flexible as GP; LR fits a linear model, and produces a linear combination of features and regression coefficients whose value for a given set of microbial communities and patient behaviour quantifies the likelihood that the patient had BV (Beck and Foster, 2014).

Regression

In Machine Learning, regression can be seen as being more general than classification. In regression (except Poisson regression), the labels, i.e., the targets of the model, are continuous quantities, instead of discrete or categorical values. The modelling process here seeks to find a function that maps from feature to target, and which can then be used to predict the labels of new unseen data with some accuracy.

Regression methods attempt to model the relationship between input, the independent variables, and output, the dependent or response variables, by constructing parametric equations in which the parameters are estimated from the training data. The most commonly used regression methods are linear models, which include linear regression (Freedman, 2009), and regularized linear regression (Mccullagh, 1984), as well as generalized linear models. Regularized regression methods fit linear models for which the number of coefficients are constrained. Regularized regression methods include ridge regression and the LASSO (Dasgupta *et al.*, 2011).

The LASSO technique was proposed by Tibshirani (1996). The LASSO and sparse least squares regression (SPLS) methods have been used for SNP selection in predicting quantitative traits. The performance in terms of r^2 (the square of Pearson correlation coefficient) for both LASSO and SPLS are almost identical in some scenarios. The LASSO produces a stable model, which is with less coefficients of each variable, because the LASSO method does not consider the effects of the correlation among SNPs, and also tends to reduce the coefficients of each variable with the shrinkage feature (shrink the feature vector) (Feng *et al.*, 2012). LASSO, along with other sparse regression models, shares the property of selecting variables and building the linear model at the same time. However, the caveat is the bias created by the regularization term, in terms of geometry, Bayesian statistics, and convex analysis, in the model.

Feature Selection

Data are demonstrated as a fixed number of features which can be binary, categorical or continuous. Identifying a good data demonstration is very domain specific and related to available measurements (Isabelle, 2006). Feature selection is crucial for the model building when there are many features in the data sampling process. Feature selection has been part of supervised learning in many real-world applications, although it can be applied to unsupervised learning scenarios as well. The high dimensional nature of many modelling tasks in bioinformatics, such as sequence analysis, microarray analysis, spectral analysis, and literature mining, has demanded the development of applications using feature selection techniques (Abu-Jamous *et al.*, 2015b). The goal is to select subset of features that can be used to better classify the given data objects (Abu-Jamous *et al.*, 2015a). A rigorous training and testing schema needs to be applied to find the statistical properties for the training set and test set, without biasing the approach by removing features that generalise well. (Libbrecht and Noble, 2015).

The three main motivations of feature selection are the following. First, to improve the overall accuracy of a prediction method by eliminating noisy or redundant features (Libbrecht and Noble, 2015). Second, to gain insights into the underlying mechanisms, e.g., answering an underlying biological question, such as identifying the genes that are associated with the corresponding functional label in order to gain insights into disease mechanisms (Glaab *et al.*, 2012; Tibshirani, 1996; Urbanowicz *et al.*, 2012). Both reasons were applied in TUSCAN (Wilson *et al.*, CRISPR Jounal accepted) with an average AUC of 0.63, where the feature selection reduced the number of uninformative features from 621 to 63, and improved the average Cross-Validation result of the model by 12% to $R=0.6$ ($p<0.05$, t-test), which in turn gave insights into the most predictive features to be important to Cas9 on-target activity, including a Guanine at position 24 (G24), a depletion of Thymine within the seed region (5–12 bases preceding the PAM) and GC content.

However, there is a third motivation, which is adopted due to computational limitations rather than for improving accuracy or gaining insights. That is, to enable a more complicated model to be fitted, which has constraints on the number of features it can

process. Typical examples of this scenario include genome wide association studies (GWAS), where a preliminary feature selection based on linear regression models is performed, followed by a multi-variate model to capture complex interactions (Boyle *et al.*, 2017). The difficulty here is that this selection process can remove features that may individually not be associated with the traits of interest, but would have been major modulating factors in the multi-variate model.

To address this issue, VariantSpark (O'Brien *et al.*, 2018) was developed, which is a RF model implemented on a more powerful compute paradigm (Apache Spark), which overcomes the limitations of traditionally implemented models. VariantSpark allows a multi-variate model to be built directly on the full set of input features, thereby not discarding possibly important features during the feature selection process. It then allows features to be ranked according to their importance, thereby supporting the two main aims of feature selection, namely improving accuracy and gaining insights in the underlying mechanisms.

Unsupervised Learning

In contrast to supervised learning, learning from instances where label information is unavailable or is not used in the modelling process is called unsupervised learning (Maglogiannis, 2007). The purpose of unsupervised learning is to identify patterns in the data, such as finding groups of similar samples or identifying a trend line.

Typically, data collected through automated processes are unlabelled data, where, especially in the life-science space, the data volumes and speed with which the data is changing prevents the creation of an expert annotated subset on which to train supervised methods. As such, extracting information and gaining insights from large volumes of unlabelled data has recently become very important. Here, we will discuss Clustering approaches, such as hierarchical clustering, K-means, and model-based clustering.

Clustering

Clustering aims to group data into categories so that the data in each category share some commonalities or exhibit some uniformity. Clustering is a useful approach, especially in the exploratory data analysis phase, during decision-making, and when serving as a pre-processing step in Machine Learning. Clustering is widely used in data mining, document retrieval and image segmentation (Krogh, 2000). Hierarchical clustering, k-means clustering, and mixture models are the most important clustering families. Cluster assignment is typically quantitative in contrast to statistical dimensionality reduction methods like principal component analysis (PCA) or multidimensional scaling (MDS), where groupings are qualitatively based upon visual inspection in 2 or 3-dimensional space, or more depending on the data.

Hierarchical clustering

Hierarchical clustering builds clusters by recursively partitioning the instances in either a top-down or bottom-up fashion. Hierarchical clustering methods can be divided into two types of clustering methods: agglomerative hierarchical clustering and divisive hierarchical clustering. In agglomerative hierarchical clustering, each object initially represents a cluster of its own, and clusters are successively merged to obtain the desired cluster structure using Ward's method. Ward recommended the criterion for selecting the pair of clusters to merge at each step is based on the optimized value of an objective function, which can be any function that reflects the investigator's purpose (Ward, 1963). In divisive hierarchical clustering, all objects initially belong to one cluster, and this cluster is successively divided into their own sub-clusters to obtain the desired cluster structure (Majoros *et al.*, 2004).

Hierarchical clustering algorithms have been used for gene sequences. A method named Unweighted Pair Group Method using arithmetic Averages (UPGMA) (average linking) is regarded the most widely used algorithm for hierarchical data clustering in computational biology. The entire collection of protein sequences has been automatically built a comprehensive evolutionary-driven hierarchy of proteins from sequence alone using a novel class of memory-constrained UPGMA (MC-UPGMA) algorithms (Loewenstein *et al.*, 2008).

k-means clustering

k-means clustering is a so-called partitional clustering algorithm, where samples for data sets are moved between clusters as the learning progresses. The algorithm begins by randomly selecting k samples and using their coordinates as the centroids (cluster centres). The algorithm then assigns each sample to its nearest centroid. This is based on a distance measure such as the Euclidian distance. Once every sample has been assigned to a centroid, each centroid is shifted to the mean of the samples assigned to that centroid. This process repeats iteratively (assign samples to centroids, adjust the centroids to reflect the mean of assigned samples) until certain stopping criteria are met. These criteria are either a maximum number of iterations, or a threshold defining a distance the centroids must move for them to not yet be considered as converged.

Many *k*-means implementations allow the user to define the above parameters (the value of *k*, the maximum number of iterations, and the distance measure). Defining the optimal value for *k* may be considered especially important, as it defines the number of clusters the algorithm will invariably build. Specifying the correct value for *k* can be a case of compromise, as increasing this value will decrease one of the error metrics (the within set sum of square error, or WSSSE), but may result in clusters

containing just one sample. One colloquial method to find the optimal value for k is the "elbow method". This method plots the WSSSE as a function of k. According to this method, the optimal value for k is the point on the graph where the relative decrease in the WSSSE drops for each increase in k (i.e., the "elbow" in the plot).

k-means, as a heuristic algorithm, may not find the global optimum. To find the global optimum in partition-based clustering, an exhaustive enumeration process of all possible partitions is required. However, this is computationally infeasible, and therefore greedy heuristics such as k-means are ideal for these clustering problems (Majoros *et al.*, 2004).

k-means clustering was used in the VariantSpark suite to cluster individuals by their ethnicity based on their genomic profile (O'brien *et al.*, 2015). When using genome sequencing data, this problem can be challenging as the number of variants across the input data can easily reach millions, even with only a moderate number (100–1000 s) of samples. This is due to many rare variants that are only present in a small set of samples. Because of this sparsity, variant data can be stored as sparse vectors. This is more efficient than using standard feature vectors, as sparse vectors need not store zero-values. Once VariantSpark has transformed variants from text files into sparse vectors, it can then cluster the individuals using the aforementioned K-means algorithm.

Generative Models

In this section we discuss generative models, which aim at generating novel data based on the patterns learned from the training data.

Mixture Models

Mixture models are also known as model-based clustering. Model-based clustering is a broad family of algorithms designed for modelling an unknown distribution as a mixture of simpler distributions, sometimes called basis distributions. The classification of mixture model clustering is based on the following four criteria: (i) the number of components in the mixture, including finite mixture model (parametric) and infinite mixture model (non-parametric); (ii) the clustering kernel, including multivariate normal models, or Gaussian mixture models (GMMs), the hidden Markov mixture models, Dirichlet mixture models, and other mixture models based on non-Gaussian distributions; (iii) the estimation method, including non-Bayesian methods and Bayesian methods; (iv) the dimensionality, including classes of factorising algorithms, such as mixture of factor analysers (MFA), MFA with common factor loadings, mixture of probabilistic principal component analysers, and so on (Abu-Jamous *et al.*, 2015b).

A widely used example of GMMs in bioinformatics is the VariantRecalibrator tool, which is part of the Genome Analysis Toolkit (GATK) (Depristo *et al.*, 2011). GATK is a suite of tools for genomic variant discovery in high-throughput sequencing data. VariantRecalibrator is, in fact, the first part of a two-stage process called VQSR (Variant Quality Score Recalibration). VariantRecalibrator uses GMMs to generate a continuous, co-varying estimate of the relationship between SNP annotations and the probability that a SNP is a genuine variant (versus a sequencing artefact). The GMM is estimated adaptively based on a set of true variants provided from highly validated sources (e.g., HapMap 3 sites, or the Omni 2.5M SNP chip array).

The second stage of VSQR is called ApplyRecalibration. Here, the adaptive GMM generated by VariantRecalibrator can be applied to both known and novel genetic variations discovered in the dataset at hand, thus evaluating the probability that each variant is real. Each variant is then annotated with a score called VQSLOD, which is the log-odds ratio of being a true variant versus being a false variant (sequencing error) under the trained GMM.

Probabilistic Graphical Models

Probabilistic graphical models use a graph-based representation to compactly encode a complex distribution over a high-dimensional space. The nodes correspond to the variables in the domain, and the edges correspond to direct probabilistic interactions between them (Burge and Karlin, 1997). Bayesian networks and Markov networks are important families of graphical representations of distributions.

Bayesian networks

Bayesian networks are directed acyclic graphs that efficiently represent a joint probability distribution over a set of random variables. In the graph, each vertex represents a random variable, and edges represent direct correlations between the variables. Each variable is independent of its non-descendants given the state of its parents. These independencies are then exploited to reduce the number of parameters for characterizing a probability distribution and computing posterior probabilities given the evidence (Stanke and Waack, 2003).

Bayesian networks have been applied to predict the cellular compartment to which a protein will be localized for its function (Bauer *et al.*, 2011). Here, the model integrates protein interactions (PPI), protein domains (Domains), post-translational modification sites (Motifs), and protein sequence data. For each protein, the network receives Boolean inputs of its random variables, e.g., 'Protein-interacts-with-Pml' = False, 'Protein-sequence-has-SUMO-site' = True, and 'Protein-associates-with-PML-bodies' = False, which are processed in its unobserved latent variables, which represent PPI, Domains, and Motifs. The sequence information is provided by a SVM classification over the protein sequence, and presented to the network as an output variable. The

compartment variable itself is hence located between the latent variables and the output variables, and the probabilities are inferred during the training. This has the benefit that the input of the latent variables (the variables with missing data because they are unobserved or missing) is not required to be present for all variables, which makes the resulting network robust against missing information.

Markov networks

Markov networks are also called Markov random fields (MRF) or undirected graphical models, are commonly used in the statistical Machine Learning domain to succinctly model spatial correlations. A Markov random field includes a graph G; the nodes represent random variables, and the edges define the independence semantics between the random variables. A random variable in a graph, G, is independent of its non-neighbours given the observed values for its neighbours (Krogh, 1997).

Probing cellular networks from different perspectives, using high-throughput genome-wide molecular assays, has become an important study in molecular biology. Probabilistic graphical models represent multivariate joint probability distributions through a product of terms with a few variables. These models are useful tools for extracting meaningful biological information from the resulting data sets (Friedman, 2004).

Hidden Markov Models

The hidden Markov model (HMM) is an important statistical tool for modelling data with sequential correlations in neighbouring samples, such as in time series data. Its most successful application has been in natural language processing (NLP). HMM have been applied with great success to problems such as part-of-speech tagging and noun-phrase chunking (Blunsom, 2004). In HMM, hidden variables are controlling the mechanism of how the data are generated. So, the attributes are directly affected by the hidden variables, for example, a segment of speech is dedicated to pronouncing a syllable, and this syllable can be seen as a value of a hidden variable.

HMMs have been used to resolve various problems of biological sequence analysis (Won *et al.*, 2007), including pairwise and multiple sequence alignment (Durbin *et al.*, 1998; Pachter *et al.*, 2002), base-calling (Liang *et al.*, 2007), gene prediction (Munch and Krogh, 2006), modelling DNA sequencing errors (Lottaz *et al.*, 2003), protein secondary structure prediction (Won *et al.*, 2007), fast ncRNA annotation (Weinberg and Ruzzo, 2006), ncRNA identification (Zhang *et al.*, 2006), ncRNA structural alignment (Yoon and Vaidyanathan, 2008), acceleration of RNA folding and alignment (Harmanci *et al.*, 2007), and many others (Yoon, 2009).

Pair HMMs can be used in dynamic programming (DP) for resolving alignment problems. A pair HMM emits a pairwise alignment in comparison with generalized HMMs (Durbin *et al.*, 1998). A combined approach named generalized pair HMM (GPHMM) has been developed in conjunction with approximate alignments, which allows users to state bounds on possible matches, for a reduction in memory (and computational) requirements, rendering large sequences on the order of hundreds of thousands of base pairs feasible. GPHMMs can be used for cross-species gene finding and have applications to DNA-cDNA and DNA-protein alignment (Pachter *et al.*, 2002).

HMMs have been widely applied for modelling genes. The *ab initio* HMM gene finders for eukaryotes include BRAKER1 (Hoff *et al.*, 2016), Seqping (Chan *et al.*, 2017), and MAKER-P (Campbell *et al.*, 2014). A procedure, GeneMarkS-T (Tang *et al.*, 2015), has been developed to generate a species-specific gene predictor from a set of reliable mRNA sequences and a genome. HMMs have demonstrated that species-specific gene finders are superior to gene finders trained on other species. Acyclic discrete phase-type distributions implemented using an HMM are well suited to model sequence length distributions for all gene structure blocks (Munch and Krogh, 2006). The state structure of each HMM is constructed dynamically from an array of sub-models that include only gene features from the training set. The comparison result from each individual gene predictor on each individual genome has demonstrated that species-specific gene finders are superior to gene finders trained on other species (Munch and Krogh, 2006).

A systematic approach, named EBSeq-HMM, using an HMM has been applied to modelling RNA-seq. In EBSeq-HMM, an auto-regressive HMM is developed to place dependence in gene expression across ordered conditions. This approach has been proved to be useful in identifying differentially expressed genes and in specifying gene-specific expression paths and inference regarding isoform expression (Leng *et al.*, 2015).

The prediction of the secondary structure of proteins is one of the most popular research topics in the bioinformatics community. The tasks of manual design of HMMs are challenging for the above prediction, an automated approach, using Genetic Algorithms (GA) has been developed for evolving the structure of HMMs. In the GA algorithm, the biologically meaningful building blocks of proteins (the set of 20 amino acids) are assembled as populations of HMMs. The space of Block-HMMs is discovered by mutation and crossover operators on 1662 random sequences, which are generated from the evolved HMM. The standard HMM estimation algorithm (the Baum-Welch algorithm) was applied to update model parameters after each step of the GA. This approach uses the grammar (probabilistic modelling) of protein secondary structures and transfers it into the stochastic context-free grammar of an HMM. This approach provides good performance of the probabilistic information on the prediction result under the single-sequence condition (Won *et al.*, 2007).

Non-coding RNAs (ncRNAs) are RNA molecules that are transcribed from DNA but not believed to be translated into proteins (Weinberg and Ruzzo, 2006). The ncRNA sequences play a role in the regulation of gene expression (Zhang *et al.*, 2006). The state-of-art methods, Covariance models (CMs), are an important statistical tool for identifying new members of a ncRNA gene family

in a large genome database using both sequence and RNA secondary structure information. Recent speed improvements through applying filters have been achieved (Weinberg and Ruzzo, 2006). In the development of detection methods for ncRNAs, Zhang *et al.* propose efficient filtering approaches for CMs to identify sequence segments and speed up the detection process. They built up the concept of a filter by designing efficient sequence based filters and provide figures of merit, such as G + C content, that allow comparison between filters. Zhang *et al.* (2006) also designed efficient sequence-based HMM filters to construct a new formulation of the CM that allows speeding up RNA alignment. This approach has been illustrated its efficiency and capability on both synthetic data and real bacterial genomes (Zhang *et al.*, 2006). Because many ncRNAs have secondary structures, an efficient computational method for representing RNA sequences and RNA secondary structure has been proposed for finding the structural alignment of RNAs based on profile context-sensitive hidden Markov models (profile-csHMMs) to identify ncRNA genes. The framework, based on profile-csHMMs , has been demonstrated to be effective for the computational analysis of RNAs and the identification of ncRNA genes (Yoon and Vaidyanathan, 2008).

The accuracy of structural predictions can be improved significantly by joint alignment and secondary structure prediction of two RNA sequences. Applying constraints that reduce computation by restricting the permissible alignments and/or structures further improves accuracy. A new approach has been developed for the purpose of establishing alignment constraints based on the posterior probabilities of nucleotide alignment and insertion. The posterior probabilities of alignment and insertion are computed for all possible pairings of nucleotide positions from the two sequences by a forward-backward calculation using a hidden Markov model. The co-incidence for nucleotide position pairs are obtained from these combined alignments, insertion posterior probabilities and the co-incidence probabilities are thresholded by a suitable alignment constraint, and this constraint is integrated with a free energy minimization algorithm for joint alignment and secondary structure prediction (Harmanci *et al.*, 2007).

A prediction method for a transcription factor prediction database has been implemented using profile HMMs of domains, and used for identifying sequence-specific DNA-binding transcription factors through sequence similarity. Transcription factor prediction based on HMMs of DNA-binding domains provides advantages. It is more sensitive than conventional genome annotation procedures because it uses the efficient multiple sequence comparison method of HMMs, and it recognizes only transcription factors that use the mechanism of sequence-specific DNA binding (Kummerfeld and Teichmann, 2006).

Deep Learning

Deep learning provides computational models composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These approaches have significantly improved the state-of-the-art in speech recognition, visual object recognition, and many other domains including biology, such as drug discovery and genomics. Deep learning has identified complex structures in large datasets by using the backpropagation algorithm. Backpropagation is an approach applied to estimate the error contribution of each neuron after the training data is processed. This algorithm can guide the network to change its internal parameters, to determine the representation in each layer from the representation in the previous layer (Lecun *et al.*, 2015).

As mentioned earlier, a standard neural network (NN) includes many simple, connected processors called neurons, each generating a sequence of real-valued activations. The input neurons become activated through sensors perceiving the environment, other neurons become activated through weighted connections from previously active neurons (Schmidhuber, 2015). However, simply stacking many layers of neurons together, i.e., making the model "deep", will not increase model performance in accuracy by much. There is a need for an effective training algorithm in deep structures. In the late 1990s, there were some attempts to alleviate this problem. The convolutional neural network (CNN) (Lecun *et al.*, 1998) was proposed in 1998 in which parameter sharing and sparsity were introduced and implemented as convolutional operations in layers close to the input layer. Despite its outstanding performance, it was largely ignored. In 2006, the Restricted Boltzman Machine (RBM) was introduced (Hinton *et al.*, 2006) where neurons are treated as probabilistic units defined by energy functions (similar to a Markov Random Field). It was trained layer by layer. This provided a practical tool to make the networks really deep, i.e., containing many layers, and therefore representing very complex models with a large number of parameters to tune.

Meanwhile, computational platforms moved from vertical scaling to horizontal scaling, leading to fast growth in parallel computing and the emergence of GPUs (general computing units). These new platforms made the training of large-scale networks possible, which, in turn, stimulated the growth of deep learning. Many very deep CNNs were proposed and successfully applied to computer vision problems. Furthermore, many other neural networks designed for temporal data such as speech went deep as well, resulting in deep recurrent neural networks (DRNN) where connections between units form a directed cycle. A particular form of RNNs called long short-term memory (LSTM) was very successful in natural language processing. Now deep learning models are incorporating other learning techniques, such as reinforcement learning (Van Otterlo and Wiering, 2012), which combines learning and generative models to solve complex problems such as game playing (AlphaGo), and design (Autodesk generative design), to name a few. Reinforcement learning is a field of machine learning inspired by behaviourist psychology. Reinforcement learning is the problem that an agent resolves it by learning behaviour through trail-and-error interactions with a dynamic environment (Kaelbling *et al.*, 1996).

Deep learning requires a large amount of data for the training process. However, this can be alleviated by adopting some transfer learning techniques, i.e., localizing pre-trained models, which have been trained on readily available related data sets, and then fine-tuned on higher quality data.

Deep learning has been used in omics, biomedical imaging, and biomedical signal processing for bioinformatics (Min *et al.*, 2016). The generation of different transcripts from single genes is guided by the alternative splicing (AS) process. A model has been implemented using a deep neural network, trained on mouse RNA-Seq data, that can predict splicing patterns in individual tissues, and differences in splicing patterns across tissues. Their framework uses hidden variables jointly representing features in genomic sequences and tissue types for predictions (Leung *et al.*, 2014). In biomedical imaging research, deep learning approaches have been demonstrated to have the capability to learn physiologically important representations, such as independent component analysis (ICA) and restricted Boltzmann machine (RBM), and detect latent relations in neuroimaging data (Plis *et al.*, 2014). Buggenthin *et al.* present a deep neural network that predicts lineage choice in differentiating primary hematopoietic progenitors using millions of image patches from brightfield microscopy and cellular movement. They combine a CNN with an RNN architecture to automatically detect local image features and retrieve temporal information about the single-cell trajectories. This approach provides a solution for the identification of cells with differentially-expressed lineage-specific genes without molecular labelling (Buggenthin *et al.*, 2017). In biomedical signal processing research, brain signals have been decoded with Deep Belief Networks, probabilistic generative models that are composed of multiple layers of latent variables, and identified higher correlations with neural patterns than Principal Component Analysis (PCA).

Conclusion

Here we have provided an overview of different Machine Learning technologies and their application in the bioinformatics space. As also covered by Libbrecht and Noble, the choice of methodology depends on the properties of the available data as well as the intended outcome (Libbrecht and Noble, 2015).

We have covered supervised versus unsupervised learning, exemplified by FFNN and K-means methods respectively. The supervised FFNN approach has been used for designing novel proteins with site-directed recombination (Bauer *et al.*, 2006), where the Machine Learning models could learn from annotated examples. In contrast, the unsupervised K-means clustering in VariantSpark groups patients based on their genomic profile, which for a global population represents the different ethnicity groups (O'brien *et al.*, 2015).

We also gave examples of choosing methodologies to achieve specific outcomes. For example we used a SVM for SUMOylation site prediction in protein sequence, as the intent was to develop the most accurate predictor (Bauer *et al.*, 2010). The trade-off for excellent performance here was the lack of insight into which biological feature contributes to the outcome. However, if gaining insights is the intent, as it was for the CRISPR target site prediction (Wilson *et al.*, CRISPR Jounal accepted), then a methodology such as RF, is more advisable as it provides a feature-importance score after training.

Throughout the article we referred to "Big Data", which is particularly attractive for Machine Learning. This is because Machine Learning relies on the iterative process of learning from examples, which requires data to be kept close to the compute resources, and ideally in memory. Recent developments in the distributed computing space have enabled this in a standardized framework, namely Apache Hadoop, and for better memory management, Apache Spark.

We therefore also discussed VariantSpark, a Machine Learning framework for high-dimensional complex data, such as genomic information (O'Brien *et al.*, 2018). It offers supervised and unsupervised Machine Learning methods, and specifically its RF implementation exceeds other Spark-based parallelization attempts in the volume of data, e.g., Google's Plant implementation (Panda *et al.*, 2009). As such, it can be applied to datasets with millions of features to fit multi-variate models, e.g., to perform Genome wide association studies (GWAS) capturing the complex interaction between genomic loci.

In conclusion, the Machine Learning field is an exciting and rapidly evolving space especially in life sciences, as data volumes here will outpace those of traditional Big Data disciplines, like astronomy or retail. Hence, dramatic breakthroughs in advanced Machine Learning or Artificial Intelligence can be expected from this domain over the next years.

References

Abu-Jamous, B., Fa, R., Nandi, A.K., 2015a. Feature Selection. Integrative Cluster Analysis in Bioinformatics. John Wiley & Sons, Ltd.

Abu-Jamous, B., Fa, R., Nandi, A.K., 2015b. Mixture Model Clustering. Integrative Cluster Analysis in Bioinformatics. John Wiley & Sons, Ltd.

Akbani, R., Kwek, S., Japkowicz, N., 2004. Applying support vector machines to imbalanced datasets. In: Proceedings of the Machine Learning, vol. 3201, pp. 39–50, ECML

Algama, M., Tasker, E., Williams, C., *et al.*, 2017. Genome-wide identification of conserved intronic non-coding sequences using a Bayesian segmentation approach. BMC Genomics 18.

Amari, S., Wu, S., 1999. Improving support vector machine classifiers by modifying kernel functions. Neural Networks 12, 783–789.

Bauer, D.C., Boden, M., Thier, R., Gillam, E.M., 2006. STAR: Predicting recombination sites from amino acid sequence. BMC Bioinformatics 7.

Bauer, D.C., Willadsen, K., Buske, F.A., *et al.*, 2011. Sorting the nuclear proteome. Bioinformatics 27, I7–I14.

Bauer, D.C., Buske, F.A., Bailey, T.L., Boden, M., 2010. Predicting SUMOylation sites in developmental transcription factors of Drosophila melanogaster. Neurocomputing 73, 2300–2307.

Beck, D., Foster, J.A., 2014. Machine learning techniques accurately classify microbial communities by bacterial vaginosis characteristics. PLOS ONE 9.

Blunsom, P., 2004. Hidden markov models. Lecture Notes 15, 18–19.

Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152, ACM.

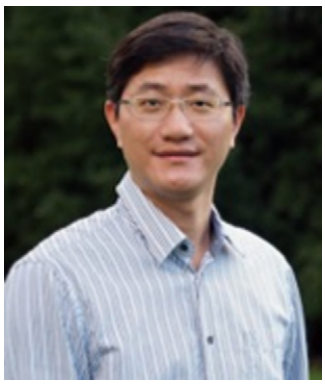
Boyle, E.A., Li, Y.I., Pritchard, J.K., 2017. An expanded view of complex traits: From polygenic to omnigenic. Cell 169, 1177–1186.

Bradley, A.P., 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.
Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.
Buggenthin, F., Buettner, F., Hoppe, P.S., *et al*., 2017. Prospective identification of hematopoietic lineage choice by deep learning. Nature Methods 14, 403–406.
Burge, C., Karlin, S., 1997. Prediction of complete gene structures in human genomic DNA. Journal of Molecular Biology 268, 78–94.
Campbell, M.S., Law, M.Y., Holt, C., *et al*., 2014. MAKER-P: A tool kit for the rapid creation, management, and quality control of plant genome annotations. Plant Physiology 164, 513–524.
Chan, K.L., Rosli, R., Tatarinova, T.V., *et al*., 2017. Seqping: Gene prediction pipeline for plant genomes using self-training gene models and transcriptomic data. BMC Bioinformatics 18.
Chen, X., Ishwaran, H., 2012. Random forests for genomic data analysis. Genomics 99, 323–329.
Cohen, H., Lefebvre, C., 2005. Handbook of Categorization in Cognitive Science. Elsevier.
Dasgupta, A., Sun, Y.V., Konig, I.R., Bailey-Wilson, J.E., Malley, J.D., 2011. Brief review of regression-based and machine learning methods in genetic epidemiology: The Genetic Analysis Workshop 17 experience. Genetic Epidemiology 35, S5–S11.
Degroeve, S., De Baets, B., Van De Peer, Y., Rouze, P., 2002. Feature subset selection for splice site prediction. Bioinformatics 18, S75–S83.
De Filippo, C., Ramazzotti, M., Fontana, P., Cavalieri, D., 2012. Bioinformatic approaches for functional annotation and pathway inference in metagenomics data. Briefings in Bioinformatics 13, 696–710.
Depristo, M.A., Banks, E., Poplin, R., *et al*., 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nature Genetics 43.491–498.
Devroye, L., Györfi, L., Lugosi, G., 2013. A Probabilistic Theory of Pattern Recognition. Springer Science & Business Media.
Dietterich, T., 1995. Overfitting and undercomputing in machine learning. ACM Computing Surveys 27, 326–327.
Domingos, P., 2012. A few useful things to know about machine learning. Communications of the ACM 55, 78–87.
Duda, R.O., Hart, P.E., Stork, D.G., 1973. Pattern Classification. New York: Wiley.
Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G., 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge university press.
Eiben, A.E., Smith, J.E., 2003. Introduction to Evolutionary Computing. Springer.
Feng, Z.Z., Yang, X.J., Subedi, S., Mcnicholas, P.D., 2012. The LASSO and sparse least squares regression methods for SNP selection in predicting quantitative traits. IEEE-ACM Transactions on Computational Biology and Bioinformatics 9, 629–636.
Freedman, D.A., 2009. Statistical Models: Theory and Practice. Cambridge university press.
Frey, T., Gelhausen, M., Saake, G., 2011. Categorization of concerns: A categorical program comprehension model. In: Proceedings of the 3rd ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools, pp. 73–82. ACM.
Friedman, N., 2004. Inferring cellular networks using probabilistic graphical models. Science 303, 799–805.
Furey, T.S., Cristianini, N., Duffy, N., *et al*., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16, 906–914.
Glaab, E., Bacardit, J., Garibaldi, J.M., Krasnogor, N., 2012. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. PLOS ONE 7.
Hamelryck, T., 2009. Probabilistic models and machine learning in structural bioinformatics. Statistical Methods in Medical Research 18, 505–526.
Harmanci, A.O., Sharma, G., Mathews, D.H., 2007. Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign. BMC Bioinformatics 8.
Heekeren, H.R., Marrett, S., Bandettini, P.A., Ungerleider, L.G., 2004. A general mechanism for perceptual decision-making in the human brain. Nature 431, 859–862.
Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural Computation 18, 1527–1554.
Hoff, K.J., Lange, S., Lomsadze, A., Borodovsky, M., Stanke, M., 2016. BRAKER1: Unsupervised RNA-seq-based genome annotation with GeneMark-ET and AUGUSTUS. Bioinformatics 32, 767–769.
Hugenholtz, P., Tyson, G.W., 2008. Microbiology – Metagenomics. Nature 455, 481–483.
Isabelle, G., 2006. Feature extraction foundations and applications. Pattern Recognition.
Jackson, P., Moulinier, I., 2007. Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization. John Benjamins Publishing.
Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285.
Korf, I., 2004. Gene finding in novel genomes. BMC Bioinformatics 5.
Krogh, A., 1997. Two methods for improving performance of an HMM and their application for gene finding. In: Proceedings of the Ismb-97 Fifth International Conference on Intelligent Systems for Molecular Biology, pp. 179–186.
Krogh, A., 2000. Using database matches with HMMGene for automated gene detection in Drosophila. Genome Research 10, 523–528.
Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., Steinbrecher, M., 2016. Computational Intelligence: A Methodological Introduction. Springer.
Kummerfeld, S.K., Teichmann, S.A., 2006. DBD: A transcription factor prediction database. Nucleic Acids Research 34, D74–D81.
Langley, P., Iba, W., Thompson, K., 1992. An analysis of Bayesian classifiers. In: AAAI-92 Proceedings: Tenth National Conference on Artificial Intelligence, pp. 223–228.
Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.
Leng, N., Li, Y., Mcintosh, B.E., *et al*., 2015. EBSeq-HMM: A Bayesian approach for identifying gene-expression changes in ordered RNA-seq experiments. Bioinformatics, 31. pp. 2614–2622.
Leung, M.K.K., Delong, A., Alipanahi, B., Frey, B.J., 2016. Machine learning in genomic medicine: A review of computational problems and data sets. Proceedings of the IEEE 104, 176–197.
Leung, M.K.K., Xiong, H.Y., Lee, L.J., Frey, B.J., 2014. Deep learning of the tissue-regulated splicing code. Bioinformatics 30, 121–129.
Liang, K.C., Wang, X.D., Anastassiou, D., 2007. Bayesian basecalling for DNA sequence analysis using hidden Markov models. IEEE-ACM Transactions on Computational Biology and Bioinformatics 4, 430–440.
Libbrecht, M.W., Noble, W.S., 2015. Machine learning applications in genetics and genomics. Nature Reviews Genetics 16, 321–332.
Loewenstein, Y., Portugaly, E., Fromer, M., Linial, M., 2008. Efficient algorithms for accurate hierarchical clustering of huge datasets: Tackling the entire protein space. Bioinformatics 24, I41–I49.
Loh, W.Y., 2011. Classification and regression trees. Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery 1, 14–23.
Lottaz, C., Iseli, C., Jongeneel, C.V., Bucher, P., 2003. Modeling sequencing errors by combining Hidden Markov models. Bioinformatics 19, Ii103–Ii112.
Maglogiannis, I.G., 2007. Emerging Artificial Intelligence Applications In Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies. Ios Press.
Majoros, W.H., Pertea, M., Salzberg, S.L., 2004. TigrScan and GlimmerHMM: Two open source ab initio eukaryotic gene-finders. Bioinformatics 20, 2878–2879.
Mccullagh, P., 1984. Generalized linear-models. European Journal of Operational Research 16, 285–292.
Mckinney, B.A., Reif, D.M., Ritchie, M.D., Moore, J.H., 2006. Machine learning for detecting gene-gene interactions: A review. Applied Bioinformatics 5, 77–88.
Michiels, S., Koscielny, S., Hill, C., 2005. Prediction of cancer outcome with microarrays: A multiple random validation strategy. Lancet 365, 488–492.
Min, S., Lee, B., Yoon, S., 2016. Deep learning in bioinformatics. Briefings in Bioinformatics.
Moore, J.H., Barney, N., Tsai, C.T., *et al*., 2007. Symbolic modeling of epistasis. Human Heredity 63, 120–133.
Mukaka, M.M., 2012. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. Malawi Medical Journal 24, 69–71.

Munch, K., Krogh, A., 2006. Automatic generation of gene finders for eukaryotic species. BMC Bioinformatics 7.
Ning, J., Beiko, R.G., 2015. Phylogenetic approaches to microbial community classification. Microbiome 3.
Ni, Y., Aghamirzaie, D., Elmarakeby, H., *et al.*, 2016. A machine learning approach to predict gene regulatory networks in seed development in arabidopsis. Frontiers in Plant Science 7.
O'brien, A.R., Saunders, N.F.W., Guo, Y., *et al.*, 2015. VariantSpark: Population scale clustering of genotype information. BMC Genomics 16.
O'Brien, A., Szul, P., Dunne, R., *et al.*, 2018. Cloud-based machine learning enables whole-genome epistatic association analyses. in preparation.
Ohler, U., Liao, G.C., Niemann, H., Rubin, G.M., 2002. Computational analysis of core promoters in the Drosophila genome. Genome Biology 3.RESEARCH0087.
Pachter, L., Alexandersson, M., Cawley, S., 2002. Applications of generalized pair hidden Markov models to alignment and gene finding problems. Journal of Computational Biology 9, 389–399.
Palmer, L.E., Dejori, M., Bolanos, R., Fasulo, D., 2010. Improving de novo sequence assembly using machine learning and comparative genomics for overlap correction. BMC Bioinformatics 11.
Panda, B., Herbach, J.S., Basu, S., Bayardo, R.J., 2009. PLANET: Massively parallel learning of tree ensembles with MapReduce. Proceedings of the VLDB Endowment 2, 1426–1437.
Pelossof, R., Fairchild, L., Huang, C.H., *et al.*, 2017. Prediction of potent shRNAs with a sequential classification algorithm. Nature Biotechnology 35, 350–353.
Picardi, E., Pesole, G., 2010. Computational methods for ab initio and comparative gene finding. Methods in Molecular Biology 609, 269–284.
Plis, S.M., Hjelm, D.R., Salakhutdinov, R., *et al.*, 2014. Deep learning for neuroimaging: A validation study. Frontiers in Neuroscience 8.
Refaeilzadeh, P., Tang, L., Liu, H., 2009. Cross-validation. In: Liu, L., Özsu, M.T. (Eds.), Encyclopedia of Database Systems. Boston, MA: Springer US.
Rish, I., 2001. An empirical study of the naive Bayes classifier. IJCAI 2001 workshop on empirical methods in artificial intelligence, pp. 41–46, IBM.
Schlauch, D., Paulson, J.N., Young, A., Glass, K., Quackenbush, J., 2017. Estimating gene regulatory networks with pandaR. Bioinformatics 33, 2232–2234.
Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61, 85–117.
Shipp, M.A., Ross, K.N., Tamayo, P., *et al.*, 2002. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature Medicine 8, 68–74.
Stanke, M., Waack, S., 2003. Gene prediction with a hidden Markov model and a new intron submodel. Bioinformatics 19, Ii215–Ii225.
Statnikov, A., Aliferis, C.F., Tsamardinos, I., Hardin, D., Levy, S., 2005. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 21, 631–643.
Stephens, Z.D., Lee, S.Y., Faghri, F., *et al.*, 2015. Big data: Astronomical or genomical? PLOS Biology 13.
Tang, S.Y.Y., Lomsadze, A., Borodovsky, M., 2015. Identification of protein coding regions in RNA transcripts. Nucleic Acids Research 43.
Tibshirani, R., 1996. Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society Series B-Methodological 58, 267–288.
Urbanowicz, R.J., Granizo-Mackenzie, A., Moore, J.H., 2012. An analysis pipeline with statistical and visualization-guided knowledge discovery for Michigan-style learning classifier systems. IEEE Computational Intelligence Magazine 7, 35–45.
Valiant, L.G., 1984. A theory of the learnable. Communications of the ACM 27, 1134–1142.
Van Otterlo, M., Wiering, M., 2012. Reinforcement learning and Markov decision processes. In: Wiering, M., Van Otterlo, M. (Eds.), Reinforcement Learning: State-of-the-Art. Berlin, Heidelberg: Springer.
Ward, J.H., 1963. Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58, 236.
Weinberg, Z., Ruzzo, W.L., 2006. Sequence-based heuristics for faster annotation of non-coding RNA families. Bioinformatics 22, 35–39.
Won, K.J., Hamelryck, T., Pruegel-Bennett, A., Krogh, A., 2007. An evolutionary method for learning HMM structure: Prediction of protein secondary structure. BMC Bioinformatics 8.
Wilson, L.O.W., Reti, D., O'Brien, A.R., Dunne, R.A., Bauer, D.C., 2018. High activity target-site identification using phenotypic independent CRISPR-Cas9 core functionality. The CRISPR Journal accepted.
Yang, Z.R., 2010. Machine Learning Approaches to Bioinformatics. World scientific.
Yoon, B.J., 2009. Hidden Markov models and their applications in biological sequence analysis. Current Genomics 10, 402–415.
Yoon, B.J., Vaidyanathan, P.P., 2008. Structural alignment of RNAs using profile-csHMMs and its application to RNA homology search: Overview and new results. IEEE Transactions on Automatic Control 53, 10–25.
Zeng, H.Y., Edwards, M.D., Liu, G., Gifford, D.K., 2016. Convolutional neural network architectures for predicting DNA-protein binding. Bioinformatics 32, 121–127.
Zhang, S.J., Borovok, I., Aharonowitz, Y., Sharan, R., Bafna, V., 2006. A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements. Bioinformatics 22, E557–E565.
Zhao, Z.H., Yang, Z.H., Luo, L., Lin, H.F., Wang, J., 2016. Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. Bioinformatics 32, 3444–3453.

Biographical Sketch



Dr. Kaitao Lai is a postdoctoral fellow at the Transformational Bioinformatics Team in Australian eHealth Research Centre and CSIRO Health and Biosecurity Business Unit. His expertise is in genome informatics, high throughput genomic data analysis, computational genome engineering, genome editing, as well as big data analysis and elastic cloud computing.He received his PhD in Bioinformatics for Plant Biotechnology and worked on microbial genomic and metagenomics data analysis, pan-genome analysis in previous postdoctoral fellow position. He is currently working on the development of predictors for CRISPR-Cpf1 in genome editing by training a Random Forests (RF) Machine Learning method, and facilitating the adaptation of the latest developments in computing infrastructure (e.g., Spark for big data analytics) and cloud technology (e.g., AWS Lambda) for research projects and commercial software products.

Dr. Natalie A Twine is a postdoctoral fellow at the Transformational Bioinformatics Team in Australian eHealth Research Centre and CSIRO Health and Biosecurity Business Unit. She works with big data technologies to understand the genetic basis of ALS. This is a collaborative project with Macquarie University and the international consortium, Project MinE. Dr. Twine has expertise in high throughput genomic and transcriptomic data analysis, clinical genomics, genetics and big data analysis. She obtained her PhD in Bioinformatics from University of New South Wales and has previously worked at UNSW, Kings College London and University College London. Natalie has 19 peer-reviewed publications (6 as senior author) with 835 citations and h-index of 12.

Aidan O'Brien is a joint PhD student between CSIRO and the John Curtin School of Medical Research at the Australian National University. His PhD project is aimed at developing sophisticated Machine Learning models to facilitate accurate CRISPR knock-in applications. He is working together with Australia's premier CRISPR facility to validate his models on novel datasets and enable new application areas. He graduated from the University of Queensland with a Bachelor of Biotechnology (1st class honours) in 2013. In his previous work, he developed GT-Scan and VariantSpark. Aidan has 4 journal publications (3 first author) with 61 citations (h-index 3). He received the "Best student and postdoc" award at CSIRO in 2015 and attracted $180K in funding to date as AI.

Dr. Yi Guo received the B. Eng. (Hons.) in instrumentation from the North China University of Technology in 1998, and the M. Eng. in automatic control from Central South University in 2002. From 2005, he studied Computer Science at the University of New England, Armidale, Australia, focusing on dimensionality reduction for structured data with no vectorial representation. He received a PhD degree in 2008. From 2008 until 2016, he was with CSIRO, working as a computational statistician on various projects in spectroscopy, remote sensing and materials science. He joined the Centre for Research in Mathematics, Western Sydney University in 2016. His recent research interests include Machine Learning, computational statistics and big data.

Dr. Denis Bauer is the team leader of the transformational bioinformatics team in CSIRO's ehealth program. She has a PhD in Bioinformatics from the University of Queensland and held Post-doctoral appointments in biological Machine Learning at the Institute for Molecular Bioscience and in genetics at the Queensland Brain institute. Her expertise is in computational genome engineering and BigData compute systems. She is involved in national and international initiatives tasked to include genomic information into medical practice, funded with $200M. She has 31 peer-reviewed publications (14 as first or senior author) with 7 in journals of IF>8 (e.g., Nat Genet.) and H-index 12. To date she has attracted more than $6.5Million in funding as Chief investigator.