APPLYING MACHINE LEARNING ALGORITHMS FOR THE ANALYSIS OF BIOLOGICAL SEQUENCES AND MEDICAL RECORDS

BY

SHAOPENG GU

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Mathematics

Specialization in Statistics

South Dakota State University

2019

THESIS ACCEPTANCE PAGE

SHAOPENG GU

This thesis is approved as a creditable and independent investigation by a candidate for the master's degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

DocuSigned by:
Xijin Ge
51ECE8AE9AA7415...
11/27/2019 | 09:26 CST

Xijin Ge

Advisor Date

DocuSigned by:
Kurt Cogswell
DA1993DDD148408...
11/27/2019 | 07:37 PST

Kurt Cogswell

Department Head Date

DocuSigned by:
Nicole B Lounsbury
35D846164547411...
11/27/2019 | 12:30 CST

Dean, Graduate School Date

ACKNOWLEDGEMENTS

I would like to give my endless thanks to my advisor, Dr. Xijin Ge, who encouragingly guided me through this research topic. Thanks for having me as your graduate student, lab member and giving a lot of valuable suggestions to complete this work. As an experienced and professional researcher in the data science and bioinformatics area, his brilliant insights help me solve many detailed problems.

Besides my advisor, I would like to thank my former advisor, Dr. Qin Ma for guiding me to the bioinformatics area. As a student who graduated from the Electrical Engineering major, a lack of biology knowledge is my top challenge during my master's study. In the past two years, Dr. Ma taught me many things not only relate to research skills but also relate to my future career. Thanks to Dr. Ma's patience and encouragement, I obtain enough time and passion to move forward for the degree.

I also want to thank my co-workers for this research topic: Dr. Bin Yu and his student, Cheng Chen. Dr. Yu brought many ideas and suggestions for enhancing the quality of my works. Cheng Chen helps me to understand biological information of DNA and RNA sequencing data through the feature extraction process.

I would like to give a special thanks to the department head, Dr. Cogswell for helping me review all materials covered by core courses of the degree.

Thanks, my committee, Dr. Rhoda Burrows. Thank you for being my committees, thanks for your precious time and valuable comments.

My sincere thanks also go all BMBL members. Dr. Adam McDermaid provides many insightful pieces of advice regarding the background of bioinformatics and future

career opportunities. Anjun Ma, one of my best friends, taught me a lot of biological knowledge. Cankun Wang, Minxuan Sun, Yiran Zhang and Yirong Wang, thank you all for your valuable comments.

Thanks to Dr. Jing Zhao, our cooperation in analyzing medical data led me to the healthcare area. Without this valuable opportunity, getting a job offer from the healthcare system will never happen.

Finally, infinite gratitude to my sweet family. My wife, Huayinyin Wang, thanks for your accompanying and strong supports for my daily life. My daughter, Caroline, brought my life more happiness and colors.

CONTENTS

LISTS OF FIGURES

LIST OF TABLES

ABSTRACT

APPLYING MACHINE LEARNING ALGORITHMS FOR THE ANALYSIS OF BIOLOGICAL SEQUENCES AND MEDICAL RECORDS

SHAOPENG GU

2019

The modern sequencing technology revolutionizes the genomic research and triggers explosive growth of DNA, RNA, and protein sequences. How to infer the structure and function from biological sequences is a fundamentally important task in genomics and proteomics fields. With the development of statistical and machine learning methods, an integrated and user-friendly tool containing the state-of-the-art data mining methods are needed. Here, we propose SeqFea-Learn, a comprehensive Python pipeline that integrating multiple steps: feature extraction, dimensionality reduction, feature selection, predicting model constructions based on machine learning and deep learning approaches to analyze sequences. We used enhancers, RNA N6-methyladenosine sites and protein-protein interactions datasets to evaluate the validation of the tool. The results show that the tool can effectively perform biological sequence analysis and classification tasks.

Applying machine learning algorithms for Electronic medical record (EMR) data analysis is also included in this dissertation. Chronic kidney disease (CKD) is prevalent across the world and well defined by an estimated glomerular filtration rate (eGFR). The progression of kidney disease can be predicted if future eGFR can be accurately estimated using predictive analytics. Thus, I present a prediction model of eGFR that was built using Random Forest regression. The dataset includes demographic, clinical and

laboratory information from a regional primary health care clinic. The final model included eGFR, age, gender, body mass index (BMI), obesity, hypertension, and diabetes, which achieved a mean coefficient of determination of 0.95. The estimated eGFRs were used to classify patients into CKD stages with high macro-averaged and micro-averaged metrics.

CHAPTER 1: Introduction for Sequencing Data Analysis

1.1 Next-Generation Sequencing

The appearance of next-generation sequencing (NGS) technology has significantly improved the quantities and qualities of biological sequences [1]. NGS provides advanced technology with many advantages: ultra-high throughput, speed, scalability and friendlily cost [2]. With NGS, the duration for sequencing an entire human genome is reduced from a decade to a single day [3] and its cost dropped from $300000 to less than $1000 [4]. The most recent released version, 232 of GenBank in NCBI contains 213,387,758 sequences and WGS in NCBI includes 1,022,913,321 sequences [5]. Analyzing biological sequences help researches to explore the structural and functional properties of sequences [6, 7], disease diagnosis [8-10], drug target development, biotechnology [11] and many others.

1.2 Machine Learning in Sequencing Data Analysis

Computational biological sequences analysis tools are urgently needed because an ever-widening gap emerges between these data and their annotations. Recently, applying machine learning algorithms for the analysis of biological sequences became a popular trend [12]. In essence, many problems can be considered as a binary or multi-class prediction tasks [13, 14], include DNA N6-methyladenosine site [15, 16], RNA N6-methyladenosine site [17], RNA-binding protein identification [18], protein function site [19], protein fold recognition [20, 21], protein-protein interaction prediction [22-24], etc.

1.3 Feature Extraction of Sequencing Data

Billions of short raw reads are generated for each sample through NGS in FASTA data format [25], which cannot directly be used for classification purposes. Thus, the step of feature extraction is required to transform reads of sequences to the mathematical data matrix using different approaches based on sequencing, physicochemical, evolutional and structural properties [26].

1.4 Feature Selection

With an increasing number of classification algorithms has been introduced, selecting the most important features to reach accurate and efficient performances becomes a new challenge [27]. Some extracted feature vectors show high dimensionality, which can cause time-consuming and overfitting issues. Therefore, selecting those features that contribute most to classification is an essential step in the sequencing data analysis [28]. Some powerful feature selection algorithms that can be used include the Chi-squared test [29], SVM-RFE [30], Lasso [31], Pearson correlation [32], ReliefF [33], and so on.

1.5 Dimensionality Reduction

Besides many supervised methods, some unsupervised learning methods such as K-means [34], PCA [35] and TSNE [36], are introduced. Dimensionality reduction can project raw feature space with high dimensionality to a new feature space via the linear or non-linear combination. Dimensionality reduction and feature selection both can reduce the model's complexity, computational resource cost and execution time, prevent overfitting issue and improve the accuracy of prediction to provide more reliable predictions.

1.6 Model Construction

Classification is a supervised learning approach to classify new observations based on the given data in machine learning. Some popular and well-developed classification algorithms are widely used in many different fields, such as SVM [37], RandomForest [38], LightGBM [39], XGBoost [40], Adaboost [41] and KNN [42], etc. Every classifier has its characters thus there is not a best classifier but only an appropriate classifier. Therefore, training multiple classifiers simultaneously can help researchers to find the best classifier.

1.7 Sequencing Data Analysis Tool

There are several computational tools are available in the public. Some tools focus only on extracting features from one or more types of sequencing data. For instance, repDNA [43], Pse-in-one 2.0 [44], PyFeat [45] and PROFEAT [46] are tools only for feature extraction. To my knowledge, there are three computational tools: IFeature [47], iLeran [48] and BioSeq-Analysis2.0 [49] that integrating multiple steps for sequencing data analysis, but the integrated classifiers and feature selection methods are not sufficient and updated. In addition, deep learning is a very powerful computational tool for classification tasks via layer by layer learning [50]. Some popular deep learning methods show convincing performances for prediction but they are not included in these packages [51].

CHAPTER 2: SeqFea-Learn – An Integrated Python Package for the Analysis of Biological Sequences

2.1 Overall Design of SeqFea-Learn

To develop a comprehensive pipeline for the classification of biological sequences, we integrated 20 feature selection methods, 16 dimensionality reduction methods and 13 classification models. In addition, this tool also contains a total of 60 methods to extract features from DNA, RNA and protein sequences, Figure 1. Compared with other software packages, SeqFea-Learn has the following advantages:

- A large variety of feature selection methods, including regularization, statistics, information, tree, and recursive feature elimination-based approaches.
- 13 classification algorithms include three deep learning approaches.
- Enhanced graphical visualization of results, including a box plot of classification accuracy and ROC curves.

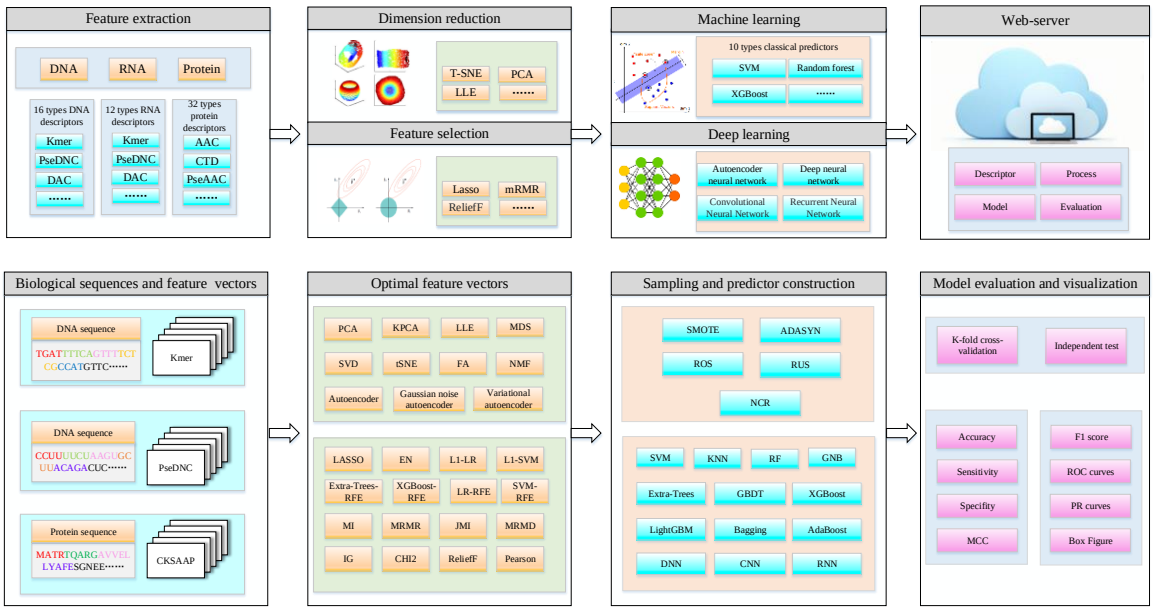

Figure 1. The pipeline of SeqFea-Learn. The Python package contains feature extraction, feature selection, dimensionality reduction, and model construction from sequences. The input is the DNA, RNA or protein sequences in the FASTA format. The outputs will provide generated feature vectors, prediction accuracy comparison, and suggestion of the best model for researchers.

2.2 Detailed Methods in SeqFea-Learn

The DNA, RNA and protein sequence S with L residues can be regarded as:

$$S = R_1 R_2 \cdots R_{L-1} R_L \tag{1}$$

where R_L represents the $L\text{-}th$ residue.

2.2.1 Feature Extraction

The step of feature extraction consists of 16 feature extraction methods for DNA and 12 feature extraction methods for RNA; 32 feature extraction methods for protein sequences, which are shown in Table 1 and Table 2, respectively.

Table 1. List of 16 DNA feature extraction methods and 12 RNA feature extraction methods

DNA Feature Extraction Methods	**RNA Feature Extraction Methods**	**Extraction Method Description**
Kmer	Kmer	DNA or RNA sequence are represented as the occurrence frequencies of k neighboring nucleic acids [55, 56]
Reverse Compliment Kmer (RCKmer)	Reverse Compliment Kmer (RCKmer)	A variant of Kmer descriptor by removing the reverse compliment Kmer [55, 57]
Pseudo Dinucleotide	Pseudo Dinucleotide	Incorporating the contiguous local sequence-order and global sequence-order information [58]

Composition (PseDNC)	Composition (PseDNC)	
Pseudo k-tuple Nucleotide Composition (PseKNC)	-	Extending the PseDNC by incorporating k-tuple nucleotide composition [59]
Dinucleotide Based Auto Covariance (DAC)	Dinucleotide Based Auto Covariance (DAC)	Measuring the correlation of the same physicochemical index between two dinucleotides separated by lag along the sequence [60, 61]
Dinucleotide Based Cross Covariance (DCC)	Dinucleotide Based Cross Covariance (DCC)	Measuring the correlation of two different physicochemical indices between two dinucleotides separated by lag nucleic acids [60, 61]
Dinucleotide Based Auto-cross Covariance (DACC)	Dinucleotide Based Auto-cross Covariance (DACC)	Combining of DAC and DCC [43]
Trinucleotide Based Auto Covariance (TAC)	-	Measuring the correlation of the same physicochemical index between trinucleotides separated by lag nucleic acids [43]
Trinucleotide Based Cross Covariance (TCC)	-	Measuring the correlation of two different physicochemical indices between two trinucleotides separated by lag nucleic acids [43]
Trinucleotide Based Auto-Cross Covariance (TACC)	-	Combining of TCC and TACC [43]
Nucleic Acid Composition (NAC)	Nucleic Acid Composition (NAC)	Calculating the frequency of each nucleic acid type in nucleotide sequence [48]
Di-Nucleotide Composition (DNC)	Di-Nucleotide Composition (DNC)	Containing 16 NAC descriptors [48]
Tri-Nucleotide Composition (TNC)	Tri-Nucleotide Composition (TNC)	Containing 64 NAC descriptors [48]

zCurve Mathematical Formula (zCurve)	zCurve Mathematical Formula (zCurve)	Calculating three components in three axis in genomic sequence analysis [45]
MonoKGap Theoretical Description (MonoKGap)	MonoKGap Theoretical Description (MonoKGap)	Calculating features based on the value of kgap [45]
MonoDiKGap Theoretical Description (MonoDiKGap)	MonoDiKGap Theoretical Description (MonoDiKGap)	Calculating features based on value of 4 ∗kgap [45]

Table 2. List of 32 Protein feature extraction methods and their description

Protein Feature Extraction	**Extraction Method Description**
Amino Acid Composition (AAC)	Calculating the frequencies of 20 kinds of amino acids [62]
Dipeptide Composition (DC)	transforming the variable length of proteins to fixed length feature vectors [62]
Composition of K-Spaced Amino Acid Pairs (CKSAAP)	Extracting important intrinsic correlation information of protein sequences in multidimensional space [63-65]
Grouped Dipeptide Composition (GDC)	A variation of the DPC descriptor which generates 25 descriptors [66]
Grouped Tripeptide Composition (GTC)	Another variation of TPC descriptor which generates 125 descriptors [66]
Conjoint Triad (CT)	Calculating the frequency of occurrence of each triad [67]
K-Spaced Conjoint Triad (KSCTriad)	Combining CT and considers the continuous amino acid units that are separated by any *k* residues [68]
Composition (C)	Calculating composition descriptors
Transition (T)	Calculating transition descriptors
Distribution (D)	Calculating distribution descriptors [69-71]

Encoding Based on Grouped Weight (EBGW)	Capturing the continuity and discontinuity features based on grouped weight coding [72]
Auto Covariance (AC)	Measuring the correlation of the same property between two residues separated by distance of *l*[73]
Moreau-Broto autocorrelation (Morean-Broto)	Measuring the physiochemical and position information between two amino acid [74]
Moran Autocorrelation (Moran)	Measuring the physiochemical information of adjacent amino acid [75]
Geary Autocorrelation (Geary)	Measuring the physiochemical information and generate positive values [76, 77]
Quasi-Sequence-Order (QSO)	Obtaining the sequence distribution patters for a specific physicochemical property [78]
Pseudo-Amino Acid Composition (PseAAC)	Extracting the physicochemical information and sequence order information [79, 80]
Amphiphilic Pseudo-Amino Acid Composition (APAAC)	Extracting the type-2 pseudo amino acid composition [79, 80]
Amino Acid Composition PSSM (ACC-PSSM)	Calculating process of amino acid composition PSSM [81, 82]
Dipeptide Composition PSSM (DPC-PSSM)	Extracting the sequence-order information in the PSSM [82]
Bi-gram PSSM (Bi-PSSM)	Calculating the frequency of the transition between amino acids [83]
Auto Covariance PSSM (AC-PSSM)	Measuring the correlation of the same property between two residues separated by lag [84]
Pseudo PSSM (PsePSSM)	Calculating the PsePSSM feature vector according to the pseudo amino acid composition [85]
AB-PSSM	Calculating feature vector based on averaged PSSM over blocks [86]
Secondary Structure Composition (SSC)	Calculating feature based normalized count of frequency of the structural motifs present at the amino-acid residue positions [87]
Accessible Surface Area composition (ASA)	Calculating feature based on normalized sum of accessible surface area [87]
Torsional Angles Composition (TAC)	Calculating features based four different types of torsional angles [87]
Torsional Angles bigram (TA-bigram)	Calculating feature based on the bigram of the torsional angles [87]

Structural Probabilities bigram (SP-bigram)	Calculating feature based on structural probabilities for each position of amino acid residue [87]
Torsional Angles Auto-Covariance (TAAC)	Calculating feature from the torsional auto-covariance [87]
Structural Probabilities Auto-Covariance (SPAC)	Calculating feature from the structural probabilities [87]

2.2.2 Feature Selection and Dimensionality Reduction

SeqFea-Learn integrated steps of feature selection and dimensionality reduction, which are shown in Table 3.

Table 3. Feature selection and dimensionality reduction methods

Feature Selection Method	**Description**	**Dimensionality Reduction Method**	**Description**
Lasso	Using Lasso liner model to recursively eliminate features [31, 88]	K-means	Clustering data by separating samples in n groups of equal variances [34]
ElasticNet	Using ElasticNet model to recursively eliminate features [89]	T-SNE	Visualizing high-dimensional data [36]
L1-SVM	Using SVM with L1 penalty model to recursively eliminate features [90]	Principal Component Analysis (PCA)	Linear dimensionality reduction using singular value decomposition [35]
CHI2	Retrieving best features based on x^2 test [91]	Kernel PCA (KPCA)	Non-linear dimensionality reduction through use of kernels [35]
Pearson Correlation (PC)	Retrieving best features based on Pearson correlation [32]	Locally linear embedding (LLE)	Reducing projection of data which preserves distances within local neighborhoods [105]
ExtraTree	Using ExtraTree model to recursively eliminate features [92]	Truncated Singular Value Decomposition (TSVD)	Linear dimensionality reduction by means of truncated singular value decomposition [106]

xgBosst	Using xgBoost model to recursively eliminate features [93]	Non-negative matrix factorization (NMF)	Reducing dimension by finding two non-negative matrix [107]
SVM-RFE	Using linear SVM model to recursively eliminate features [100]	Multi-dimensional Scaling (MDS)	Reducing dimension by modeling data as distances in a geometric space [108]
LOG-RFE	Using Logistic Regression model to recursively eliminate features [94]	Independent Component Analysis (ICA)	Reducing dimension by finding components with some sparsity [109]
Mutual Information (MI)	Retrieving best features based mutual information [95]	Factor Analysis (FA)	Reducing dimension by performing a maximum likelihood estimate [110]
Minimum Redundancy Maximum Relevance (MRMR)	Selecting features that still having high correlation to the classification variable [96]	Agglomerate Feature (AF)	Recursively merges feature instead of samples [111]
Joint Mutual Information (JMI)	Retrieving best features based joint mutual information [97]	Gaussian Random Projection (GRP)	Reducing the dimension by projecting the original input space using the Gaussian distribution [112]
Maximum Relevance Maximum Distance (MRMD)	Retrieving best features by measuring relevance and redundancy between features [98]	Sparse Random Projection (SRP)	Reducing dimension by projecting the original input space using a sparse random matrix [113]
ReliefF	Retrieving best features by calculating and ranking a feature score for each feature [33]	Autoencoder	Reducing the dimension using encode and decode neural network [114]
Trace Ratio	Retrieving best features by calculating the corresponding score in trace ratio form [99]	Gaussian Noise Autoencoder (GNA)	Corrupting input before being passed to autoencoder neural network [115]
Gini Index	Retrieving best features by constructing the measure function based on Gini-Index [100]	Variational Autoencoder (VA)	Neural network can be trained with stochastic gradient descent [116]

Spectral Feature Selection (SPEC)	Retrieving best features based on structure induced [101]	-	-
Fisher Score	Retrieving best features based on scores of features under the Fisher criterion [102]	-	-
T Score	Retrieving best features based on their t-score [103]	-	-
Information Gain (IG)	Retrieving best features based on their information gain [104]	-	-

2.2.3 Models Construction

SeqFea-Learn integrated 10 popular classifiers include SVM, KNN, RF, LightGBM, XGBoost, Adaboost [118], Extra-Tree, Gaussian Naïve Bayes (GNB) [119], GBDT [117]. The tool also integrated three deep learning methods, including deep neural network (DNN) [52], convolutional neural network (CNN) [53], and recurrent neural network (RNN) [54].

2.2.4 Cross-validation and Models Evaluation

Stratified 5-Folds cross-validator is used for obtaining classification accuracy and plotting ROC curves. All models are evaluated using classification accuracy that reflects the fraction of correct predictions:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (2)$$

Most structural and functional of sequences predictions are binary classification and the accuracy can be calculated by:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

where TP, TN, FP and FN in the above equations represent true positive, true negative, false positive and false negative, respectively.

$$Accuracy = \frac{TP(i)+TN(i)}{TP(i)+TN(i)+FP(i)+FN(i)} \quad (4)$$

where i means i^{th} classes.

2.3 Application of SeqFea-Learn

Three prediction tasks were performed for DNA, RNA and protein sequences respectively to evaluate our tool. These classification performances are comparable and even more effective than the state-of-the-art approaches, which indicate our proposed python package is competitive for the analysis of biological sequences.

2.3.1 Enhancers Classification

Enhancers play an important role in analyzing gene expression. The dataset contains we used 1484 enhancer samples and 1484 non-enhancer samples [120]. We applied five DNA feature extraction methods: PSTNP, Kmer, pseDNC, BE and DNC to construct predictors. We also found that fusing these feature descriptors as one mixed descriptor can effectively represent the information and improve classification performance. The highest AUCs of 13 predictors are shown in Table 4. Compare to BioSeq-Analysis 2.0 (AUC: 0.82), our tool shows a better classification performance.

Table 4. AUC based on different feature descriptors for enhancer

Feature Extraction Methods	PSTNP	Kmer	pseDNC	BE	DNC	Five Descriptors Fusion
Highest AUC of 13 Predictors	0.90 (SVM)	0.84 (SVM)	0.84 (DNN)	0.83 (GNB)	0.84 (RNN)	0.91 (DNN)

All of 20 feature selection methods are applied to the fused vector. These selected feature vectors are then used to construct 13 classifiers for finding the best. Based on our observation, the selected feature vector using the Extra-Tree method can achieve better prediction performance, Figure 2. The execution time of modeling is significantly reduced, Table 5.

Table 5. Comparison of number of features and modeling execution time of enhancers

	Fused feature vector	Selected feature vector based Extra-Tree
Number of Features	1296	50
Execution Time	34m 42s	4m 5s

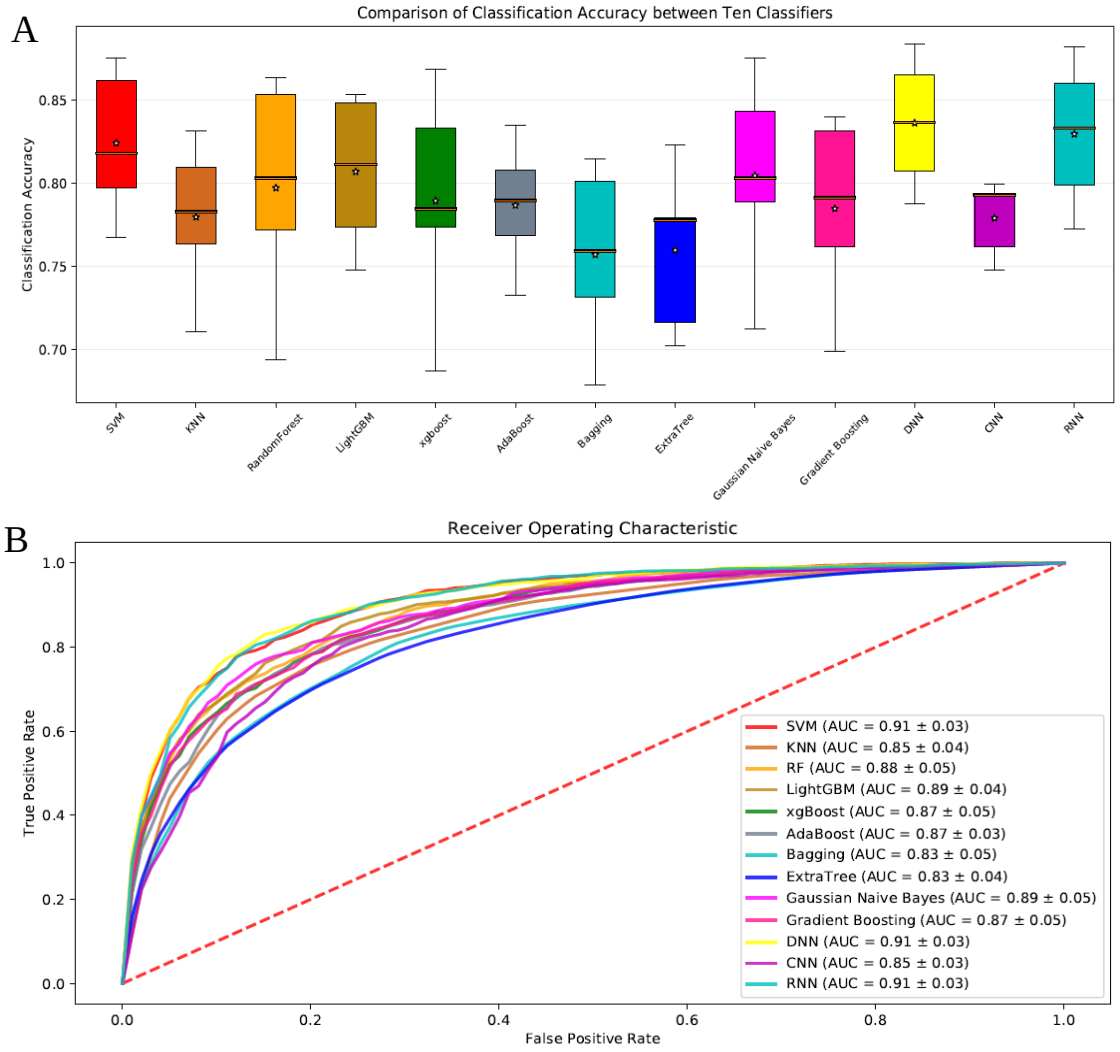

Figure 2. The boxplot of classification accuracies (A) and ROC curves (B) of DNA enhancers using various classifiers with Extra-Tree feature selection method. (A) 13 classifiers all achieve satisfactory accuracy, and SVM, DNN, RNN obtain superior performance than other classifiers. (B) The ROC curves of 13 classifier indicate DNN and RNN achieved better results.

2.3.2 RNA N6-methyladenine Sites Prediction

N6-methyladenosine (m6A) refers to methylation of the adenosine nucleotide acid at the nitrogen-6 position. It is highly related to a series of biological processes, such as

splicing events, mRNA exporting, nascent mRNA synthesis, nuclear translocation and translation process [17]. The m6A dataset contains 2614 sequences, where 1307 represents true methyladenosine sites, and the remaining 1307 are false methyladenosine sites. BioSeq-Analysis2.0 achieves 0.73 AUC with RandomForest classifier. Similarly, the fused feature vector shows a better classification performance, Table 6. After the step of feature selection and model construction, the vector using the ReliefF feature selection method displays better predictions, Figure 3.

Table 6. AUC based on different feature descriptors for RNA N6-methyladenine sites

Feature Extraction Methods	**PSTNP**	**PseDNC**	**DNC**	**TNC**	**MonoKGap**	**Five Descriptors Fusion**
Highest AUC of 13 Predictors	**0.88 (SVM)**	**0.69 (DNN)**	**0.68 (SVM)**	**0.71 (DNN)**	**0.66 (DNN)**	**0.89 (SVM)**

Table 7. Comparison of number of features and modeling execution time of RNA 6mA data

	Fused feature vector	**Selected feature vector based Extra-Tree**
Number of Features	**186**	**50**
Execution Time	**6m 13s**	**4m 26s**

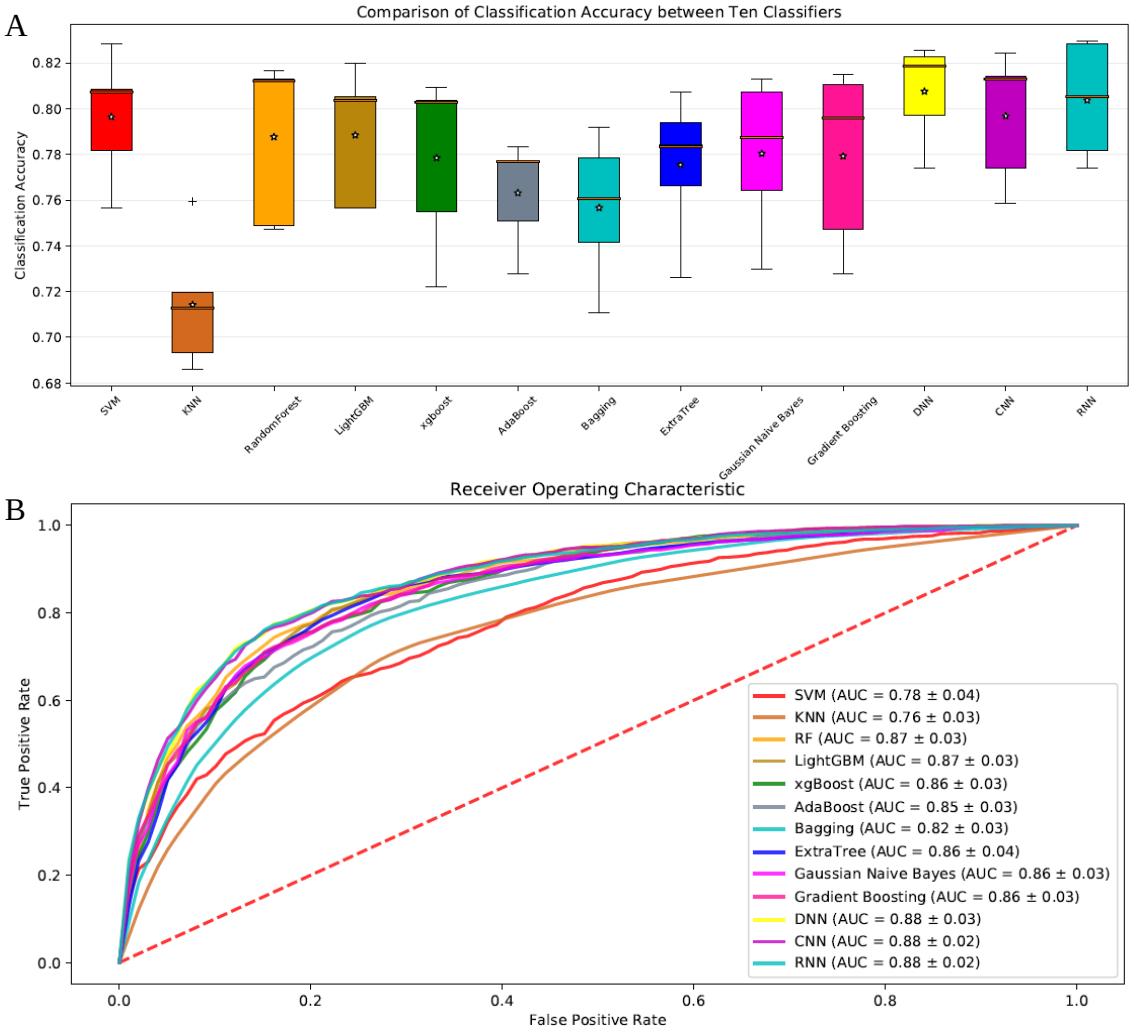

Figure 3. The boxplot accuracies (A) and ROC curves (B) under different classifiers on RNA N6-methyladenine sites dataset via ReliefF feature selection. (A) The boxplot of 13 classifiers and deep learning methods achieve better performance and the KNN is the worst. (B) The ROC curves of 13 classifier and DNN, CNN and RNN obtain the best prediction performance.

2.3.3 Protein-protein interactions prediction

The analysis of protein-protein interactions (PPIs) can help to understand the protein function, construct the complete interactome and study the signaling pathways. In this

section, the dataset includes 5594 PPI samples and 5594 non-PPI samples [153]. We fused CTDC, CTDT, CTDD, EBGW, Geary, PseAAC, PsePSSM, abPSSM to obtain the feature representation information. After comparing all selected feature vectors' predicting performances (Table 8 and 9), the MRMR feature selection method shows a better performance, Figure 4.

Table 8. AUC based on different feature descriptors for protein-protein interactions data

Feature Extraction Methods	**CTDC**	**CTDT**	**EBGW**	**Geary**	**PseAAC**	**PsePSSM**	**abPSSM**	**Five Descriptors Fusion**
Highest AUC of 13 Predictors	**0.92 (CNN)**	**0.96 (RF)**	**0.96 (GBDT)**	**0.91 (RNN)**	**0.95 (DNN)**	**0.96 (LightGBM)**	**0.94 (DNN)**	**0.98 (LightGBM)**

Table 9. Comparison of number of features and modeling execution time of PPIs

	Fused feature vector	**Selected feature vector based MRMR**
Number of Features	**2066**	**200**
Execution Time	**1006m 20s**	**110m 30s**

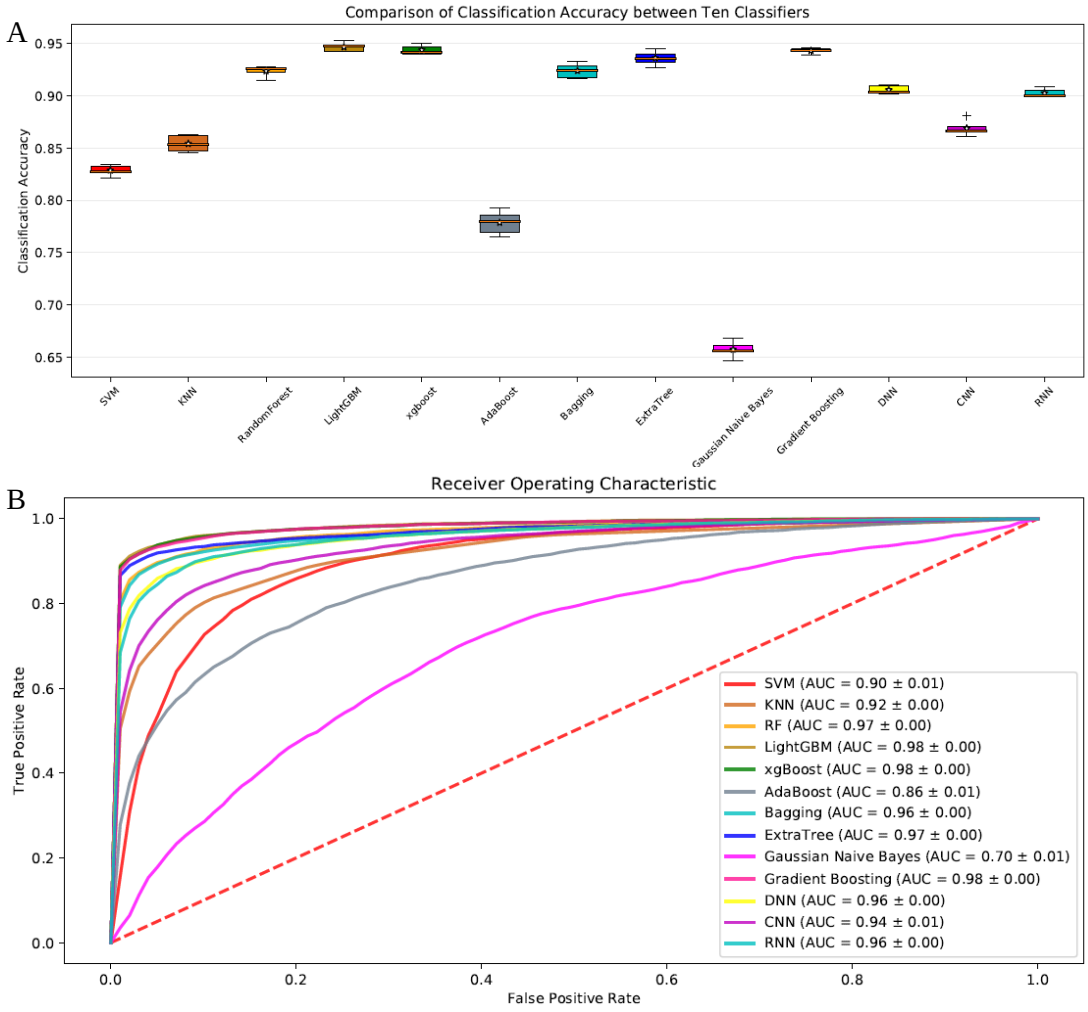

Figure 4. The boxplot accuracies (A) and ROC curves (B) under different classifiers on protein-protein interactions dataset via MRMR feature selection. (A) The boxplot of 13 classifiers and LightGBM achieve better performance and the GNB is the worst. (B) The ROC curves of 13 classifiers and LightGBM and xgBoost obtain the best prediction performance.

2.4 Summary and Conclusion

With the rapid increase of DNA, RNA and protein sequences, the analysis and process of the biological sequences are urgently needed. Therefore, we developed an intuitive and comprehensive Python package and web server called SeqFea-Learn to perform steps of feature extraction, feature selection, dimensionality reduction, and model construction to predict the structure and function of unseen sequences. SeqFea-Learn for the first time integrated 20 types of feature selection methods and 16 kinds of dimensionality reduction approach to deal with dimensionality disaster and prevent overfitting issues. It also offers 10 popular classifiers and 3 deep learning frameworks to satisfy users' needs. The tool will generate visible results to provide a user clear idea to compare and select the best classifier. To further test the validity, we perform three predicting tasks: enhancers, RNA N6-methyladenine sites and protein-protein interactions prediction. Integrated feature selection and dimensionality reduction methods reduce as much as 80% modeling time. These classification performances indicate SeqFea-Learn is an effective and accurate biological sequencing analysis tool compared with other state-of-the-art approaches.

CHAPTER 3: Predicting Outcomes of Chronic Kidney Disease from EMR Data Based on Random Forest Regression

3.1 Chronic Kidney Disease and eGFR

The increasing incidence of chronic kidney disease (CKD) in the United States and around the world lays an enormous burden on healthcare [121, 122]. By December 2015, there were 703,243 prevalent patients with End Stage Renal Disease (ESRD), with the unadjusted incident rate of 378 per million [123]. In 2017, there were approximately 500,000 patients on different dialysis modalities (91% are on hemodialysis), 20,000 received transplants [123]. Treatments that are effective in patients with advanced CKD also increase health care costs and lead to adverse effects [124]. Thus, it is essential to identify earlier stage CKD and prevent its progression to ESRD [125]. However, the biggest challenge is that most people do not have any signs or symptoms in the early stages and go undetected until an advanced stage.

Early identification and targeted intervention of CKD have attracted considerable attention from clinicians and researchers since both have the potential to reduce the number of patients progressing to ESRD and lower the mortality rate related to CKD and associated healthcare costs [126]. With the growing availability of Electronic Medication Record (EMR) data, various predictive models for disease progression have been developed to facilitate the decision-making process of health care providers [124, 127, 128]. Choi et al. classified disease progression models into two categories based on the extent of targeted diseases: models focusing on a specific disease and those focusing on a broader range of conditions. Among those disease-specific progression models, some are validating specific hypotheses of disease progression based on experts' knowledge [124, 129, 130], while others are driven by the application of advanced statistical methods [131-

133]. Approaches that can be generalized to model the progression of multiple diseases have been proposed, where statistical methods and machine learning techniques are widely used [134, 135]. For kidney disease, different models have been developed in predicting CKD stages to ESRD over time and in predict variations of GFR in patients [126, 128, 136, 137].

Estimated glomerular filtration rates (eGFRs) have been used in primary care to assist the early detection and staging of CKD [138, 139]. The eGFR formula [140] is:

$$eGFR = 141 * \min\left(\frac{S_{Cr}}{K}, 1\right)^{\alpha} * \max\left(\frac{S_{Cr}}{K}, 1\right)^{-1.209} * 0.993^{age} * 1.018[if\ female] * 1.159[if\ African\ American] \quad (5)$$

where eGFR (estimated glomerular filtration rate) = mL/min/1.73 m2; SCr (standardized serum creatinine) = mg/dL, κ = 0.7 (females) or 0.9 (males), α = −0.329 (females) or −0.411 (males), min = indicates the minimum of SCr/κ or 1, max = indicates the maximum of SCr/κ or 1, and age = years.

Although routine reporting of eGFR had positive effects in clinical practice, including prevention of CKD progression and reduction of CKD related complications, there are still concerns in its negative effects caused by overdiagnosis [138]. Studies have begun using an alternative measurement, such as eGFR decline derived from eGFR, to evaluate and predict CKD progression [141, 142]. Researchers investigated the association between eGFR change and ESRD risk and mortality risk respectively, where age and gender factors were taken into account [141, 143, 144]. Large eGFR decline were associated with greater hazard ratios of ESRD in several clinical trials [145, 146]. However, a smaller eGFR changes, which is a reflection of the short-term treatment effect of kidney disease, is underexamined [141].

3.2 Machine Learning in EMR Data Analysis

The application of statistical models and machine learning techniques have been rapidly growing in estimating health and disease outcomes [147]. Cerqueira et al.developed a model using the Cox proportional hazard regression in predicting the risks that pre-dialysis pediatric patients progress to ESRD from CKD [128]. Decruyenaere et al. compared the performances of machine learning methods with logistic regression in predicting the occurrence of delayed renal graft in renal transplant patients [148]. Their results showed that support vector machine outperformed logistic regression in terms of sensitivity. Kumar compared six machine learning classifiers (Random Forest, Sequential Minimal Optimization, NaiveBayes, Radial Basis Function, Multilayer Perceptron Classifier, and SimpleLogistic) in CKD classification and identified that Random forest outperformed the other classifiers [149].

Since GFR is the best test in measuring the level of kidney function [123, 126], the renal function of a CKD patient can be predicted if their GFR variations can be predicted. Consequently, the time to reach GFR thresholds corresponding to stages of CKD can be anticipated. An integrated expert system has been used in predicting future GFR based on selected clinical variables and demonstrated reliable accuracy [126]. However, there is still a lack of efficient methods for predicting the individual level timeframe of CKD progression. Specifically, Random Forest Regression, featured with a reduction in overfitting and less variance, has not been used to predict the progression of renal function yet. This study predicted future eGFR values using Random Forest regression based on real-world EMR data representing the general population in the upper Midwest. The main aim of this study is to propose an efficient and reliable clinical tool that allows

us to identify patients at risk of ESRD at an earlier stage. Such a tool can offer primary care physicians the opportunity to preemptively suggest the preventive strategies that can attenuate the development of this challenging disease in patients that reside in our agricultural communities.

3.3 Methods

3.3.1 Data Acquisition

The dataset used in this study comes from real-world clinical data. We built up a cohort consisting of 120,495 patients aged from 20 to 80 in Sioux Falls, SD, region that receiving primary care from Sanford Health. By consulting with the nephrologist, we pulled out data elements influencing GFR variations for this cohort from the comprehensive Sanford EMR database for years 2009–17. None of the identifiable information was extracted to protect patients' privacy. We are focusing on the progression of CKD, so only the "clinical" encounter data was included. Those data elements contain patients' eGFR records for years 2009–17, the ICD-10 codes [150] for CKD, Hypertension, Diabetes, and Obesity, and their demographic information comprising Age, Gender, and Race. A detailed description of the data elements is given in Table 10.

Table 10. Predictor and covariate data type breakdown

Feature	**Data elements**
Predictor	
eGFR	All clinical encounter eGFR data with testing dates were pulled out for each patient
Covariates	
Age	Continuous
Gender	Categorical
Race/Ethnicity	Categorical

BMI	Continuous
Hypertension	Flagged for each patient (ICD-10: I10, I11, I12, I13, I15, I16)
Diabetes	Flagged for each patient (ICD-10: E08, E09, E10, E11, E13)
Obesity	Flagged for each patient (ICD-10: E66.9)

3.3.2 Data Pre-processing

The extracted data were formatted into three separate tables: (1) eGFR table with rows representing patients and columns containing eGFR for multiple years; (2) Demographic table consisting of demographic information; and (3) Disease table composed of diagnosis status of hypertension, diabetes, and obesity. The processing of these data tables is illustrated in Figure 5 and described below.

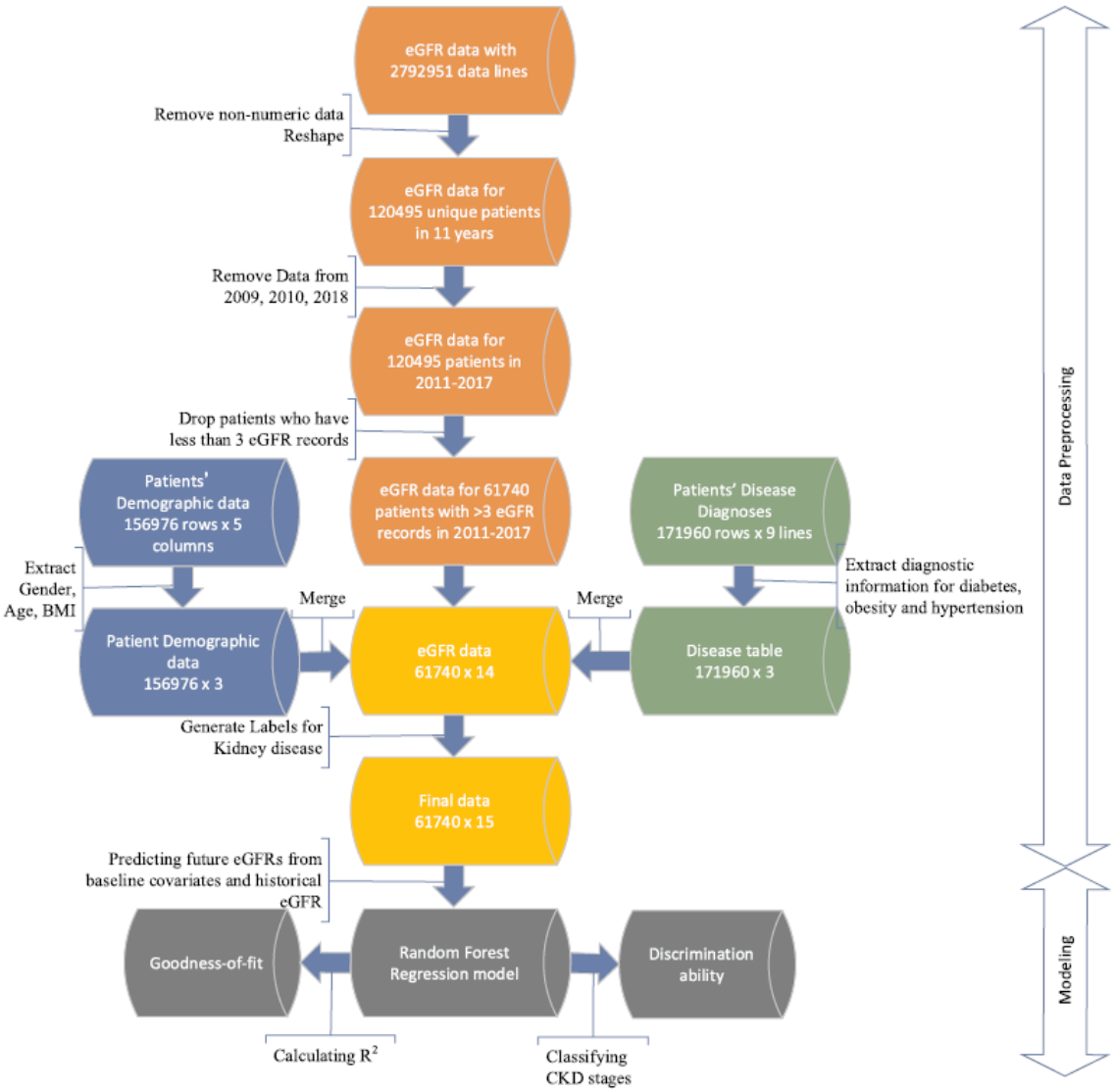

Figure 5. Workflow of the data preprocessing, including initial eGFR data, demographic and disease information, and data merging and filtering. This process resulted in 61,740 samples with 15 variables each.

1. The eGFR table has 120,495 unique patients and 10 columns, each of which representing eGFR records in years 2009–18. First, the non-numeric eGFR records (e.g. “>90”)) were considered as missing data and marked as “NA.” For patients with

more than one eGFR values in a specific year, the median of these values was calculated and kept for that year in the table.

2. More than 95% eGFR records are missing in 2009 and 2010, so data from these two years were omitted. Since the data in 2018 was not complete when the data was extracted, we also excluded the records in this year. Patient lines were removed from the data if they have no more than three available records from 2011 to 2017. The final eGFR table has 61,740 unique patients and 7 years eGFR data for each patient with at least three eGFR values.
3. Next, the different CKD stages were determined by eGFR values in the physical laboratory. Therefore, the CKD stages true labels were created using eGFR. The minimum eGFR value in each of the years between 2011 and 2017 was evaluated first, and then the CKD stages labels were produced based on the following equation:

$$Practical\ CKD\ Stage = \begin{cases} 1.\ if\ \min eGFR\ \geq 90 \\ 2.\ if\ 60 \leq \min eGFR < 90 \\ 3.\ if\ 30 \leq \min eGFR < 60 \\ 4.\ if\ 15 < \min eGFR \leq 30 \\ 5.\ if\ \min eGFR \leq 15 \end{cases} \quad (6)$$

4. The true labels were also merged into eGFR matrix based on their index (patient ID).
5. The current eGFR matrix includes 61,740 unique patients, and each patient has 7 years eGFR values from 2011 to 2017 and labels for the CKD stage from 1 to 5. The final data table was created by merging the eGFR table with the demographic table and the disease table by matching their patient IDs.

3.4 Construction of Random Forest Regression Model

The longitudinal design of this study enables the estimation the future eGFR value from the past eGFR values adjusted by clinical covariates. We selected Random Forest regression as the primary model because of its efficiency and accuracy to predict 1 year, 2 years and 3 years eGFRs from the historical eGFR records between years 2011–14.

Baseline covariates and predictors: The variables included in the analysis were baseline eGFR, age, gender, ethnicity, body mass index (BMI), hypertension, diabetes, obesity.

Outcome: eGFR values in the year 2015, 2016, and 2017 were considered as the outcome variable. This is based on the consensus that GFR is the best measure of kidney function.

Model development: the inputs of this model are the attributes of the *i*th patient denoted by a vector $Xi = (xi1,\ldots, xin)$ which includes eGFR values from multiple years and other covariates listed in Table 1. The output is the future eGFR for the *i*th patient denoted by *Gij* where *j* indicating a future year.

In the computational experiment, we used the processed dataset with 61,740 unique patients. For building the model in predicting eGFR of 2015, the patient must have recorded eGFR in 2015, and at least two recorded eGFR between 2011 and 2014. Similar requirements were used in predicting eGFR of 2016 and 2017. Other years' eGFR values were imputed and filled by the median eGFR value of each patient. All models were built using scikit-learn package [151]. The parameters of Random Forest Regressor were determined using the grid-search method. Only two parameters, number of estimators and maximum number of features, were tuned because they can determine

numbers of trees in forest and how the tree will split and grow. We also randomly split the dataset and repeat the training process five times with different sets to avoid over fitting for our models.

3.5 Assessment of model performance

3.5.1 Goodness-of-fit

The model fit of the proposed Random Forest Regression was measured using the coefficient of determination $R2$ to show how well the fitted eGFR value approximates the real eGFR value. $R2$ is a measure used to represent the percent of variation explained, i.e., the proportion of variance in the dependent variable that can directly be attributed to variance in the independent variables. An $R2$ of 1 would indicate all changes we see in the dependent variable are caused by changing our independent variables, whereas an $R2$ of 0 means no such direct impact. We also checked the residual plot since randomly distributed residuals indicate the model fits the data well.

3.5.2 Discrimination

The estimated eGFR values were used to classify patients into different CKD stages based on Eq. (1). Both micro-average and macro average were generated to illustrate the classification accuracy of the Random Forest model.

3.6 Results

In Random Forest regression analysis, the predicting accuracy was enhanced by optimizing the values of hyperparameters, where the default values and the optimized values of the hyperparameters were shown in Table 11. The predicted versus observed eGFR values in years 1–3 were plotted for both the default and optimized hyperparameters in Figure 6. The $R2$ was increased from default to optimized

hyperparameters in each of the three years. The Root of Mean Squared Error (RMSE) in Figure 6 illustrated that the optimized hyperparameters provided a more accurate prediction that the default values. It is also worse noticing that the prediction accuracy decreased over time. With the optimal parameters, we further examined the importance of the features included in the analysis whose results were given in Figure 7. It is not surprising that previous eGFR records played essential roles than other features since eGFR is decreasing continuously over time. Although the information of age and BMI are considered in estimating GFR using the eGFR formula, predictions based solely on the previous eGFR are not sufficient. Age and BMI, as illustrated in Figure 7, still contribute to 4.7–9% to the future three years of eGFR respectively. All the other features, including Race, Gender, Obesity, Hypertension, and Diabetes, accounted for a total of 2.7–3.9% of the variances.

Table 11. Hyperparameters used in the Random Forest Regression for the default and optimized models.

	Default	**Optimized**
# of trees	10	100
Max depth	None	None
Max sample split	2	2
Min samples leaf	1	1
Max features	11	8
Bootstrap	True	True

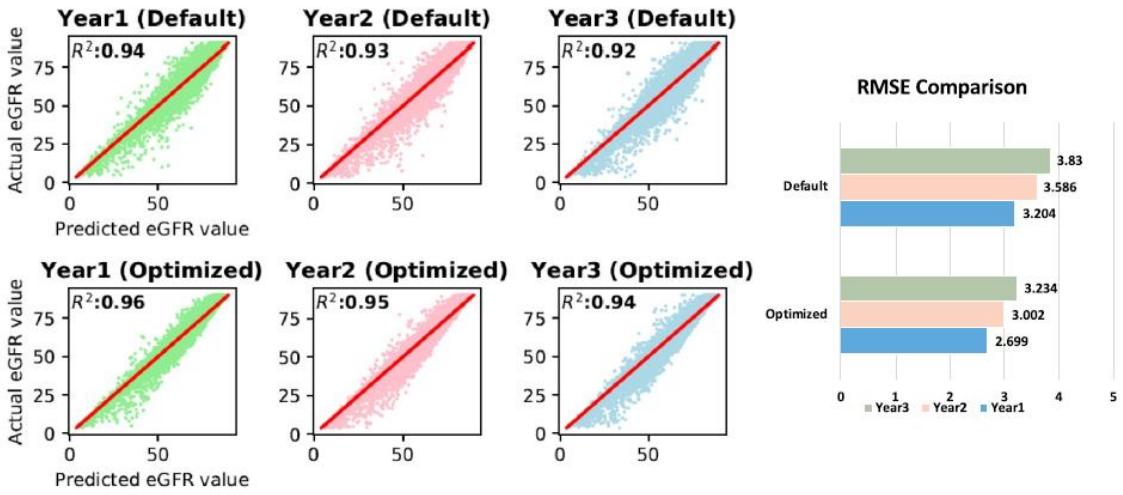

Figure 6. Goodness of fit based on R^2 of the Random Forest Regression model in predicting eGFR in year 1 to year 3 for the default and optimized models. RMSE comparison for each year is also provided for the default and optimized models.

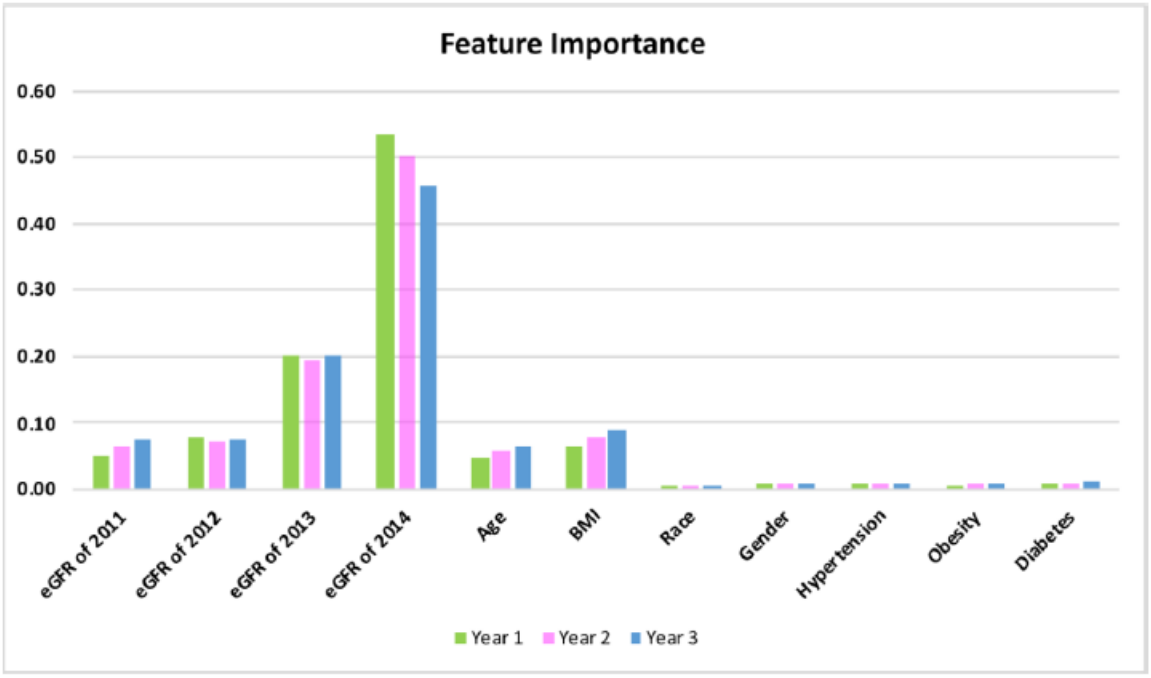

Figure 7. Feature importance in predicting eGFR values in years 1-3 using optimized parameter values in Random Forest Regression.

3.7 Conclusion and Discussion

In this study, we proposed a model in predicting future eGFR values, which is based on Random Forest regression that can efficiently learn from the real world EMR data and accurately predict future patient outcomes. We validated this model on an EMR dataset extracted from a health system located in the Great Plains. The computational experiment achieved an average $R2$ of 0.95 over three years with small variation. And an 88% Macro Recall and a 96% Macro Precision by averaging over three years were obtained by dividing patients into different CKD stages using estimated eGFRs. Besides, we identified the crucial features that contribute to the variation of future eGFRs, which include recent eGFR records, Age and BMI. Therefore, our proposed predictive model of eGFR has excellent potential to be developed into a clinical decision support tool to assist doctors in providing preventive advice to patients.

One of the limitations of this work is that only patients with numeric eGFR records were included, which exclude those patients without CKD symptoms in the study period. However, those excluded patients can serve as a control group whose clinical information can be incorporated into the predictive model to adjust the parameter estimations. Also, the current study only contained historical eGFRs, demographic characteristics, and relevant disease diagnoses. Studies have shown that an individual's genetic and phenotypic characteristics both affect their risk in developing kidney disease, including genetic mutations, a family history, gender, ethnicity, age, obesity, socioeconomic status, smoking, nephrotoxins, acute kidney injury, diabetes mellitus, and hypertension [152]. Thus, we are planning to address those issues in future studies to improve the practicability of the predictive model of eGFR in support of patient care.

APPENDIX: SeqFea-Learn Tutorial

SeqFea-Learn - An intergrated python package for DNA, RNA and Protein sequecing data analysis

Include Feature extraction, Feature selection, Dimensionality Reduction, Models Construction for sequencing data.

Table of Contents

Installation

The package is developed using Python 3(Python Version 3.0 or above) and it can be run on Linux operatinig system. We strongly recommand user to install Ancaonda Python 3.7 or above version to avoid installing other packages.

After installaing Anaconda, the following packages need to be installed:

1. xgboost
2. skrebate
3. lightgbm

The source code is freely available at: https://github.com/ashinandjay/FeatureSelection

To install our tool, first download the zip file manually from github, or use the code below in Unix:

```
cd your_folder_path
wget https://github.com/ashinandjay/FeatureSelection/archive/master.zip
```

Unzip the file:

```
unzip master.zip
```

Data Preparation

The DNA, RNA or protein sequence data (FASTA format) and their labels (txt format) are required for using our feature selection tool.

DNA Feature Extraction

The tool includes 16 feature extraction methods for DNA sequencing data.

DNA Extraction Method	DNA Extraction Number
Kmer	1
Reverse Compliment Kmer	2
Pseudo dinucleotide composition	3
Pseudo k-tuple nucleotide composition	4
Dinucleotide-based auto covariance	5
Dinucleotide-based cross covariance	6
Dinucleotide-based auto-cross covariance	7

Trinucleotide-based auto covariance	8
Trinucleotide-based cross covariance	9
Trinucleotide-based auto-cross covariance	10
Nucleic acid composition	11
Di-nucleotide composition	12
Tri-nucleotide composition	13
zcurve	14
monoMonoKGap	15
monoDiKGap	16

DNA_Feature_Extraction require two inputs: DNA Extraction number and DNA sequencing data.

Run DNA_Feature_Extraction.py:

```
DNA_Feature_Extraction.py [DNA Extraction number] [DNA sequencing data]
```

Example: Use **kmer** method to extract features from DNA sequencing data

```
DNA_Feature_Extraction.py 1 DNA_sequencing.txt
```

RNA Feature Extraction

The tool includes 12 feature extraction methods for RNA sequencing data.

RNA Extraction Method	RNA Extraction Number
Kmer	1
Reverse Compliment Kmer	2
Pseudo dinucleotide composition	3
Dinucleotide-based auto covariance	4
Dinucleotide-based cross covariance	5
Dinucleotide-based auto-cross covariance	6
Nucleic acid composition	7
Di-nucleotide composition	8
Tri-nucleotide composition	9
zcurve	10
monoMonoKGap	11
monoDiKGap	12

RNA_Feature_Extraction require two inputs: RNA Extraction number and RNA sequencing data.

Run RNA_Feature_Extraction.py:

```
RNA_Feature_Extraction.py [RNA Extraction number] [RNA sequencing data]
```

Example: Use **kmer** method to extract features from RNA sequencing data

```
RNA_Feature_Extraction.py 1 RNA_sequencing.txt
```

Protein Feature Extraction

The tool includes 32 feature extraction methods for Protein sequencing data.

Protein Extraction Method	Protein Extraction Number
Amino acid composition	1
Composition of k-spaced amino acid pairs	2
Dipeptide composition	3
Grouped dipeptide composition	4
Grouped tripeptide composition	5
Cojoint triad	6
k-spaced cojoint triad	7
Composition	8
Transition	9
Distribution	10
Encoding based on grouped weight	11
Auto covariance	12
Moran autocorrelation	13
Geary autocorrelation	14
Quasi-sequence-order	15
Pseudo-amino acid composition	16
Amphiphilic pseudo-amino acid composition	17
Amino Acid Composition PSSM	18
Dipeptide composition PSSM	19
Pseudo PSSM	20
Auto covariance PSSM	21
Cross covariance PSSM	22
Auto Cross covariance PSSM	23
Bigram-PSSM	24
AB-PSSM	25
Secondary structure composition	26
Accessible surface area composition	27
Torsional angles composition	28
Torsional angles bigram	29
Structural probabilities Bigram	30
Torsional angles auto-covariance	31
Structural probabilities auto-covariance	32

Protein_Feature_Extraction require two inputs: Protein Extraction number and Protein sequencing data.

Run Protein_Feature_Extraction.py:

```
Protein_Feature_Extraction.py [Protein Extraction number] [Protein sequencing data]
```

Example: Use **Amino acid composition** method to extract features from Protein sequencing data

```
Protein_Feature_Extraction.py 1 Protein_sequencing.txt
```

Feature Selection

Our Feature Selection tool contains 20 supervised selection methods.

Feature Selection Method	Feature Selection Number
Lasso	1
Elastic Net	2
L1-SVM	3
CHI2	4
Pearson Correlation	5
ExtraTree	6
XGBoost	7
SVM-RFE	8
LOG-RFE	9
Mutual Information	10
Minimum Redundancy Maximum Relevance	11
Joint Mutual Information	12
Maximum-Relevance-Maximum-Distance	13
ReliefF	14
Trace Ratio	15
Gini index	16
SPEC	17
Fisher Score	18
T Score	19
Information Gain	20

For using our Feacture Selection Tool, Four inputs are required: 1. Feauture selection number (See the table above) 2. Number of feature to select (how many number of feature you want) 3. Feature Vectors (Feature extraction output file) 4. Label Vectors (labels for sequencing)

Run Feature_Selection.py:

```
Feature_Selection.py [Feauture selection number] [Number of feature to select] [Feature Vectors] [Label Vectors]
```

Example: Using **Lasso** method to select **3** features

```
Feature_Selection.py 1 3 Feaute_Vectors.csv label.txt
```

Dimensionality Reduction

Our Feature Reduction tool contains 16 unsupervised dimensionality reduction methods.

Dimensionality Reduction Method	Feature Reduction Number
K-means	1
T-SNE	2
Principal Component Analysis	3
Kernel PCA	4
Locally-linear embedding	5
Singular Value Decomposition	6
Non-negative matrix factorization	7
Multi-dimensional Scaling	8
Independent Component Analysis	9
Factor Analysis	10

Agglomerate Feature	11
Gaussian random projection	12
Sparse random projection	13
Autoencoder	14
Gaussian Noise Autoencoder	15
Variational Autoencoder	16

For using our Feacture Reduction Tool, Three inputs are required: 1. Feauture Reduction number (See the table above) 2. Number of Clusters (how many number of Clusters you want) 3. Feature Vectors (Feature extraction output file)

Run Feature_Reduction.py:

```
Feature_Reduction.py [Feauture Reduction number] [Number of Clusters to select] [Feature Vectors]
```

Example: Using PCA method to select 3 clusters

```
Feature_Reduction.py 3 3 Feaute_Vectors.csv
```

Feature Evaluation

Feature selection method can be evaluated using 10 classifiers and 3 deep learning methods include SVM, KNN, RandomForest, LightGBM, XGBoost, Adaboost, Bagging, ExtraTree, gaussian Naïve Bayes, gradient boosting, DNN, CNN and RNN predictors. The classification accurcay comparison files (plot and table) will be generated in same folder of code.

Run Feature_Evaluation.py:

```
Feature_Evaluation.py [Feauture selection output] [Label Vectors]
```

Example: evaluating Lasso selection method

```
Feature_Evaluation.py Lasso.csv label.txt
```

REFERENCES

1. Selvaraj, S. and J. Natarajan, *Microarray Data Analysis and Mining Tools.* Bioinformation, 2011. **6**(3): p. 95-99.
2. Garber, M., et al., *Computational methods for transcriptome annotation and quantification using RNA-seq.* Nature methods, 2011. **8**(6): p. 469-477.
3. Behjati, S., & Tarpey, P. S. (2013). What is next generation sequencing? *Archives of Disease in Childhood - Education & Practice Edition*, *98*(6), 236–238. doi: 10.1136/archdischild-2013-304340.
4. Cost of Next-Generation Sequencing. (n.d.). Retrieved from https://www.illumina.com/science/technology/next-generation-sequencing/beginners/ngs-cost.html.
5. GenBank and WGS Statistics. (n.d.). Retrieved from https://www.ncbi.nlm.nih.gov/genbank/statistics/.
6. Wang, Z., M. Gerstein, and M. Snyder, *RNA-Seq: a revolutionary tool for transcriptomics.* Nature reviews genetics, 2009. **10**(1): p. 57-63.
7. Ozsolak, F. and P.M. Milos, *RNA sequencing: advances, challenges and opportunities.* Nature reviews genetics, 2011. **12**(2): p. 87-98.
8. Prince, M.E., et al., *Identification of a subpopulation of cells with cancer stem cell properties in head and neck squamous cell carcinoma.* Proc Natl Acad Sci U S A, 2007. **104**(3): p. 973-8.
9. Navin, N., et al., *Tumour evolution inferred by single-cell sequencing.* Nature, 2011. **472**(7341): p. 90-4.
10. Xu, X., et al., *Single-cell exome sequencing reveals single-nucleotide mutation characteristics of a kidney tumor.* Cell, 2012. **148**(5): p. 886-95.
11. D'Agaro, E. (2017). NGS genome annotation profiling using data analysis workflows. *Journal of Biotechnology*, *256*. doi: 10.1016/j.jbiotec.2017.06.039.
12. Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, *16*(6), 321–332. doi: 10.1038/nrg3920.
13. Karchin, R., Karplus, K., & Haussler, D. (2002). Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, *18*(1), 147–159. doi: 10.1093/bioinformatics/18.1.147.
14. Cai, C. (2003). SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Research*, *31*(13), 3692–3697. doi: 10.1093/nar/gkg600.
15. Feng, P., Yang, H., Ding, H., Lin, H., Chen, W., & Chou, K.-C. (2019). iDNA6mA-PseKNC: Identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. *Genomics*, *111*(1), 96–102. doi: 10.1016/j.ygeno.2018.01.005.
16. Sasnauskas, G. (2018). DNA binding domain of restriction endonuclease McrBC in complex with N4-methylcytosine DNA. doi: 10.2210/pdb6gcf/pdb.
17. Lynch, S., & Kool, E. (2015). N6-Methyladenosine RNA. doi: 10.2210/pdb2mvs/pdb.
18. Han, L. Y. (2004). Prediction of RNA-binding proteins from primary sequence by a support vector machine approach. *Rna*, *10*(3), 355–368. doi: 10.1261/rna.5890304.
19. Lee, J. S., & Ondrechen, M. J. (2011). Electrostatic Properties for Protein Functional Site Prediction. *Protein Function Prediction for Omics Era*, 183–196. doi: 10.1007/978-94-007-0881-5_10.

20. Fischer, D., & Eisenberg, D. (2008). Protein fold recognition using sequence-derived predictions. *Protein Science*, *5*(5), 947–955. doi: 10.1002/pro.5560050516.
21. Yan, K., Xu, Y., Fang, X., Zheng, C., & Liu, B. (2017). Protein fold recognition based on sparse representation based classification. *Artificial Intelligence in Medicine*, *79*, 1–8. doi: 10.1016/j.artmed.2017.03.006.
22. Han, M., Song, Y., Qian, J., & Ming, D. (2018). Sequence-based prediction of physicochemical interactions at protein functional sites using a function-and-interaction-annotated domain profile database. *BMC Bioinformatics*, *19*(1). doi: 10.1186/s12859-018-2206-2.
23. Lo, S. L., Cai, C. Z., Chen, Y. Z., & Chung, M. C. M. (2005). Effect of training datasets on support vector machine prediction of protein-protein interactions. *Proteomics*, *5*(4), 876–884. doi: 10.1002/pmic.200401118.
24. Constructing Reliable Protein-Protein Interaction (PPI) Networks. (2017). *Computational Prediction of Protein Complexes from Protein Interaction Networks*, 15. doi: 10.1145/3064650.3064653.
25. Zhang, H. (2016). Overview of Sequence Data Formats. *Methods in Molecular Biology Statistical Genomics*, 3–17. doi: 10.1007/978-1-4939-3578-9_1.
26. Feature Extraction and Classification. (2014). *Medical Diagnosis Using Artificial Neural Networks*, 159–169. doi: 10.4018/978-1-4666-6146-2.ch011.
27. Deep Learning in Python: Preventing Overfitting. (2019). doi: 10.4135/9781526493453.
28. Introduction to Feature and Gene Selection. (2011). *Feature Selection and Ensemble Methods for Bioinformatics*, 117–122. doi: 10.4018/978-1-60960-557-5.ch008.
29. An Introduction to the Chi-Square Test. (2017). doi: 10.4135/9781473980525.
30. Wang, J., Shan, G., Duan, X., & Wen, B. (2011). Improved SVM-RFE feature selection method for multi-SVM classifier. *2011 International Conference on Electrical and Control Engineering*. doi: 10.1109/iceceng.2011.6058060.
31. Missing-Value, M.-V. (n.d.). The Lasso for Linear Models. *Statistical Learning with Sparsity*, 22–43. doi: 10.1201/b18401-4.
32. Applications Of Pearson Correlation To Measurement Theory. (n.d.). *Correlation and Regression*, 67–96. doi: 10.4135/9781412983815.n4.
33. Yang, F., Cheng, W., Dou, R., & Zhou, N. (2011). An improved feature selection approach based on ReliefF and Mutual Information. *International Conference on Information Science and Technology*. doi: 10.1109/icist.2011.5765246.
34. Suryavanshi, A. S. (2016). A Survey Paper on Modified Approach for Kmeans Algorithm. *International Journal of Emerging Trends in Science and Technology*. doi: 10.18535/ijetst/v3i02.04.
35. A Comparative Studyon Kernel PCA and PCA Methods for Face Recognition. (2016). *International Journal of Science and Research (IJSR)*, *5*(5), 1844–1847. doi: 10.21275/v5i5.nov163830.
36. Priam, R. (2018). Symmetric Generative Methods and tSNE: A Short Survey. *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. doi: 10.5220/0006684303560363.
37. Qi, X., Silvestrov, S., & Nazir, T. (2017). Data classification with support vector machine and generalized support vector machine. doi: 10.1063/1.4972718.

38. Vanicek, T. (2019). Classification in major depressive disorder using randomForest and various cortical and subcortical gray matter measures. *Intrinsic Activity*, *7*(Suppl. 1). doi: 10.25006/ia.7.s1-a3.49.
39. Wang, D., Zhang, Y., & Zhao, Y. (2017). LightGBM. *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics - ICCBB 2017*. doi: 10.1145/3155077.3155079.
40. Sun, J., Wang, S., & Du, J. (2017). Research on Classification Model of Equipment Support Personnel Based on Collaborative Filtering and Xgboost Algorithm. *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*. doi: 10.1109/iccsec.2017.8446746
41. Zhang, Y., & He, P. (2010). A revised AdaBoost algorithm: FM-AdaBoost. *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. doi: 10.1109/iccasm.2010.5623209.
42. Kramer, O. (2013). K-Nearest Neighbors. *Dimensionality Reduction with Unsupervised Nearest Neighbors Intelligent Systems Reference Library*, 13–23. doi: 10.1007/978-3-642-38652-7_2.
43. Liu, B., Liu, F., Fang, L., Wang, X., & Chou, K.-C. (2014). repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics*, *31*(8), 1307–1309. doi: 10.1093/bioinformatics/btu820.
44. Liu, B., Wu, H., & Chou, K.-C. (2017). Pse-in-One 2.0: An Improved Package of Web Servers for Generating Various Modes of Pseudo Components of DNA, RNA, and Protein Sequences. *Natural Science*, *09*(04), 67–91. doi: 10.4236/ns.2017.94007.
45. Muhammod, R., Ahmed, S., Farid, D. M., Shatabda, S., Sharma, A., & Dehzangi, A. (2019). PyFeat: a Python-based effective feature generation tool for DNA, RNA and protein sequences. *Bioinformatics*, *35*(19), 3831–3833. doi: 10.1093/bioinformatics/btz165.
46. Li, Z. R., Lin, H. H., Han, L. Y., Jiang, L., Chen, X., & Chen, Y. Z. (2006). PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Research*, *34*(Web Server). doi: 10.1093/nar/gkl305.
47. Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., … Song, J. (2018). iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, *34*(14), 2499–2502. doi: 10.1093/bioinformatics/bty140.
48. Chen, Z., Zhao, P., Li, F., Marquez-Lago, T. T., Leier, A., Revote, J., … Song, J. (2019). iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Briefings in Bioinformatics*. doi: 10.1093/bib/bbz041.
49. Liu, B., Gao, X., & Zhang, H. (2019). BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches. *Nucleic Acids Research*. doi: 10.1093/nar/gkz740.
50. Ketkar, N. (2017). Introduction to Deep Learning. *Deep Learning with Python*, 1–5. doi: 10.1007/978-1-4842-2766-4_1.

51. Deep Learning in Python: Different Types of Deep Learning Network. (2019). doi: 10.4135/9781526493439.
52. Komorowski, J., Kurzejamski, G., & Sarwas, G. (2019). DeepBall: Deep Neural-Network Ball Detector. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. doi: 10.5220/0007348902970304.
53. Convolutional Neural Network. (2018). *Encyclopedia of Social Network Analysis and Mining*, 418–418. doi: 10.1007/978-1-4939-7131-2_100208.
54. Gelenbe, E. (1992). Learning in the Recurrent Random Neural Network. *Neural Networks*, 1–12. doi: 10.1016/b978-0-444-89330-7.50004-1.
55. Noble, W.S., Kuehn, S., Thurman, R., Yu, M., and Stamatoyannopoulos, J. (2005) Predicting the in vivo signature of human gene regulatory sequences. Bioinformatics, 21, i338-343.
56. Lee, D., Karchin, R. and Beer, M.A. (2011) Discriminative prediction of mammalian enhancers from DNA sequence. Genome research, 21, 2167-2180.
57. Gupta, S., Dennis, J., Thurman, R.E., Kingston, R., Stamatoyannopoulos, J. and Noble, W.S. (2008) Predicting human nucleosome occupancy from primary sequence. PLoS computational biology, 4, e1000134.
58. Chen, W., Feng, P.M., Lin, H. and Chou, K.C. (2013) iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. Nucleic Acids Res, **41**, e68.
59. Guo, S.H., Deng, E.Z., Xu, L.Q., Ding, H., Lin, H., Chen, W. and Chou, K.C. (2014) iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. Bioinformatics, **30**, 1522-1529.
60. Dong, Q., Zhou, S. and Guan, J. (2009) A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. Bioinformatics, 25, 2655-2662.
61. Guo Y., Yu L., Wen Z. and Li, M. (2008) Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. Nucleic Acids Res. 36, 3025-3030.
62. Bhasin, M. and Raghava, G.P. (2004) Classification of nuclear receptors based on amino acid composition and dipeptide composition. J. Biol. Chem., 279, 23262-23266.
63. Chen, K., Jiang Y., Du, L., Kurgan, L. and (2009) Prediction of integral membrane protein type by collocated hydrophobic amino acid pairs. J. Comput. Chem., 30, 163-172.
64. Chen, K., Kurgan, L. and Rahbari, M. (2007) Prediction of protein crystallization using collocation of amino acid pairs. Biochem. Biophys. Res. Commun., 355, 764-769.
65. Chen, K., Kurgan, L.A. and Ruan, J. (2007) Prediction of flexible/rigid regions from protein sequences using k-spaced amino acid pairs. BMC Struct. Biol., 7, 25.
66. Lee, T.Y., Lin, Z.Q., Hsieh, S.J., Bretaña, N.A. and Lu, C.T. (2011) Exploiting maximal dependence decomposition to identify conserved motifs from a group of aligned signal sequences. Bioinformatics, 27, 1780-1787.

67. Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li Y. and Jiang, H. (2007) Predicting protein-protein interactions based only on sequences information. Proc. Natl. Acad. Sci. U. S. A., 104, 4337-4341.
68. Göktepe, Y.E. and Kodaz, H. (2018) Prediction of protein-protein interactions using an effective sequence based combined method. Neurocomputing, 303, 68-74.
69. Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, X., and Chen, Y. Z. (2003) SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res., 31, 3692-3697.
70. Cai, C.Z., Han, L.Y., Ji, Z.L. and Chen, Y.Z. (2004) Enzyme family classification by support vector machines. Proteins, 55, 66-76.
71. Dubchak, I., Muchnik, I., Holbrook, S.R. and Kim, S.H. (1995) Prediction of protein folding class using global description of amino acid sequence. Proc. Natl. Acad. Sci. U. S. A., 92, 8700-8704.
72. Zhang, Z.H., Wang, Z.H., Zhang, Z.R. and Wang, Y. X. (2006) A novel method for apoptosis protein subcellular localization prediction combining encoding based on grouped weight and support vector machine. FEBS Lett., 580, 6169-6174.
73. Guo Y., Yu L., Wen Z. and Li, M. (2008) Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. Nucleic Acids Res. 36, 3025-3030.
74. Horne, D.S. (1988) Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. Biopolymers, 27, 451-477.
75. Sokal, R.R. and Thomson, B.A. (2006) Population structure inferred by local spatial autocorrelation: an example from an Amerindian tribal population. Am. J. Phys. Anthropol, 129, 121-131.
76. Feng, Z.P. and Zhang, C.T. (2000) Prediction of membrane protein types based on the hydrophobic index of amino acids. J. Protein Chem., 19, 269-275.
77. Lin, Z. and Pan, X.M. (2001) Accurate prediction of protein secondary structural content. J. Protein Chem., 20, 217-220.
78. Du, X., Sun, S., Hu, C., Yao, Y., Yan, Y. and Zhang Y. (2017) DeepPPI: boosting prediction of protein-protein interactions with deep neural networks. J. Chem. Inf. Model, 57, 1499-1510.
79. Chou K.C. (2001) Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins, 43, 246-255.
80. Chou, K.C. (2005) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics, 21, 10-19.
81. Liu, T., Geng, X., Zheng, X., Li, R. and Wang, J. (2012) Accurate prediction of protein structural class using auto covariance transformation of PSI-BLAST profiles. Amino acids, 42, 2243-2249.
82. Liu, T., Zheng, X. and Wang, J. (2010) Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile, Biochimie, 92, 1330-1334.
83. Saini, H. Raicar, G., Lal, S.P. Dehzangi, A., Imoto, S. and Sharma, A. (2016) Protein fold recognition using genetic algorithm optimized voting scheme and profile bigram. J Softw., 11, 756–767.

84. Dong, Q., Zhou, S. and Guan, J. (2009) A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. Bioinformatics, 25, 2655-2662.
85. Shen, H.B., and Chou, K.C. (2007) Nuc-PLoc: a new web-server for predicting protein subnuclear localization by fusing PseAA composition and PsePSSM. Protein Eng. Des. Sel. 20, 561.
86. Jeong, J.C., Lin, X. and Chen, X.W. (2011) On position-specific scoring matrix for protein function prediction. IEEE/ACM transactions on computational biology and bioinformatics, 8, 308-315.
87. Rayhan, F., Ahmed S., Shatabda S., Farid D.M., Mousavian Z., Dehzangi A. and Rahman M.S. (2017) iDTI-ESBoost: identification of drug target interaction using evolutionary and structural features with boosting. Sci. Rep., 7, 17731.
88. Libal, U. (2011). Feature selection for pattern recognition by LASSO and thresholding methods - a comparison. *2011 16th International Conference on Methods & Models in Automation & Robotics*. doi: 10.1109/mmar.2011.6031338.
89. Teisseyre, P. (2017). CCnet: Joint multi-label classification and feature selection using classifier chains and elastic net regularization. *Neurocomputing*, *235*, 98–111. doi: 10.1016/j.neucom.2017.01.004.
90. Maldonado, S., Montoya, R., & López, J. (2016). Embedded heterogeneous feature selection for conjoint analysis: A SVM approach using L1 penalty. *Applied Intelligence*, *46*(4), 775–787. doi: 10.1007/s10489-016-0852-5.
91. Liu, H., & Setiono, R. (n.d.). Chi2: feature selection and discretization of numeric attributes. *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*. doi: 10.1109/tai.1995.479783.
92. Tree-Structured Classifier. (n.d.). *SpringerReference*. doi: 10.1007/springerreference_66000.
93. Li, J. (2018). Analysis of the Rise and Fall of International Futures Based on Xgboost Algorithm. *Finance*, *08*(05), 211–220. doi: 10.12677/fin.2018.85025.
94. Introduction to the Logistic Regression Model. (2005). *Applied Logistic Regression*, 1–30. doi: 10.1002/0471722146.ch1.
95. Lajevardi, S. M., & Hussain, Z. M. (2009). Feature selection for facial expression recognition based on mutual information. *Exhibition*. doi: 10.1109/ieeegcc.2009.5734265.
96. Zhao, X.-M. (2013). Maximum Relevance/Minimum Redundancy (MRMR). *Encyclopedia of Systems Biology*, 1191–1192. doi: 10.1007/978-1-4419-9863-7_432.
97. Liu, H., & Ditzler, G. (2017). A fast information-theoretic approximation of joint mutual information feature selection. *2017 International Joint Conference on Neural Networks (IJCNN)*. doi: 10.1109/ijcnn.2017.7966441.
98. Shirzad, M. B., & Keyvanpour, M. R. (2015). A feature selection method based on minimum redundancy maximum relevance for learning to rank. *2015 AI & Robotics (IRANOPEN)*. doi: 10.1109/rios.2015.7270735.
99. Wang, H., Yan, S., Xu, D., Tang, X., & Huang, T. (2007). Trace Ratio vs. Ratio Trace for Dimensionality Reduction. *2007 IEEE Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/cvpr.2007.382983.

100. Test, E., Zigic, L., & Kecman, V. (2013). Feature ranking using Gini index, scatter ratios, and nonlinear SVM RFE. *2013 Proceedings of IEEE Southeastcon*. doi: 10.1109/secon.2013.6567380.
101. Zhao, Z. A., & Liu, H. (2011). Spectral Feature Selection for Data Mining. doi: 10.1201/b11426.
102. A Hybrid Feature Selection Method Based On Fisher Score And Genetic Algorithm. (2016). *Journal of Mathematical Sciences: Advances and Applications*, *37*(1), 51–78. doi: 10.18642/jmsaa_7100121627.
103. Polat, K., & Güneş, S. (2009). A new feature selection method on classification of medical datasets: Kernel F-score feature selection. *Expert Systems with Applications*, *36*(7), 10367–10373. doi: 10.1016/j.eswa.2009.01.041.
104. Dhir, C. S., Iqbal, N., & Lee, S.-Y. (2007). Efficient feature selection based on information gain criterion for face recognition. *2007 International Conference on Information Acquisition*. doi: 10.1109/icia.2007.4295788.
105. Sun, X., & Lu, Y. (2016). Locally Linear Embedding based on Rank-order Distance. *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*. doi: 10.5220/0005658601620169.
106. Garnadi, A. D. (2017). Kompresi Citra Menggunakan Truncated Singular Value Decomposition (TSVD), Sebuah Eksplorasi Numerik. doi: 10.31227/osf.io/uqrwe.
107. Larsen, J. S., & Clemmensen, L. K. H. (2015). Non-negative Matrix Factorization for Binary Data. *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. doi: 10.5220/0005614805550563.
108. Chen, Y. (n.d.). Multi-dimensional scaling and MODELLER based evolutionary algorithms for protein model refinement. doi: 10.32469/10355/43046.
109. R., G. (2012). Introduction: Independent Component Analysis. *Independent Component Analysis for Audio and Biosignal Applications*. doi: 10.5772/52324.
110. Data Reduction: Factor Analysis and Cluster Analysis. (n.d.). *Statistical Methods for Geography*, 193–209. doi: 10.4135/9781849209953.n10.
111. Clustering: Practice - Agglomerative Clustering. (2018). doi: 10.4135/9781526469205.
112. Fong, H., Zhang, Q., & Wei, S. (2007). Image Reconstruction Based on Sub-Gaussian Random Projection. *Fourth International Conference on Image and Graphics (ICIG 2007)*. doi: 10.1109/icig.2007.28.
113. Kim, Y., & Toh, K.-A. (2008). Sparse random projection for efficient cancelable face feature extraction. *2008 3rd IEEE Conference on Industrial Electronics and Applications*. doi: 10.1109/iciea.2008.4582897.
114. Yu, J., Huang, Y., & Liu, C. (2017). Classification method for uncertain data based on sparse denoising autoencoder neural network. *Advances in Modelling and Analysis B*, *60*(1), 210–223. doi: 10.18280/ama_b.600113.
115. Gualtierotti, A. F. (2015). Likelihoods for Signal Plus Gaussian Noise Versus Gaussian Noise. *Detection of Random Signals in Dependent Gaussian Noise*, 1087–1160. doi: 10.1007/978-3-319-22315-5_17.
116. Hsu, W.-N., & Glass, J. (2018). Scalable Factorized Hierarchical Variational Autoencoder Training. *Interspeech 2018*. doi: 10.21437/interspeech.2018-1034.

117. Al-Timemy, A. H. (2017). Boosting-based decision tree for improved screening of vibroarthrographic signals. *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*. doi: 10.1109/icabme.2017.8167551.
118. Grossmann, E. (2004). AdaTree: Boosting a Weak Classifier into a Decision Tree. *2004 Conference on Computer Vision and Pattern Recognition Workshop*. doi: 10.1109/cvpr.2004.296.
119. Padmanaban, H. (2016). Comparative Analysis of Naive Bayes and Tree Augmented Naive Bayes Models. doi: 10.31979/etd.n7jg-e3uh.
120. Gruskin, E. A., & Rich, A. (1993). B-DNA to Z-DNA structural transitions in the SV40 enhancer: Stabilization of Z-DNA in negatively supercoiled DNA minicircles. *Biochemistry*, *32*(9), 2167–2176. doi: 10.1021/bi00060a007.
121. Coresh, J. (2007). Prevalence of Chronic Kidney Disease and Associated Risk Factors—United States, 1999-2004. *Jama*, *297*(16), 1767. doi: 10.1001/jama.297.16.1767.
122. Webster, A. C., Nagler, E. V., Morton, R. L., & Masson, P. (2017). Chronic Kidney Disease. *The Lancet*, *389*(10075), 1238–1252. doi: 10.1016/s0140-6736(16)32064-5.
123. US Renal Data System 2016 Annual Data Report: Epidemiology of Kidney Disease in the United States. (2017). *American Journal of Kidney Diseases*, *69*(3). doi: 10.1053/j.ajkd.2017.01.036.
124. Tangri, N. (2011). A Predictive Model for Progression of Chronic Kidney Disease to Kidney Failure. *Jama*, *305*(15), 1553. doi: 10.1001/jama.2011.451.
125. Locatelli, F., Vecchio, L. D., & Pozzoni, P. (2002). The importance of early detection of chronic kidney disease. *Nephrology Dialysis Transplantation*, *17*(suppl 11), 2–7. doi: 10.1093/ndt/17.suppl_11.2.
126. Norouzi, J., Yadollahpour, A., Mirbagheri, S. A., Mazdeh, M. M., & Hosseini, S. A. (2016). Predicting Renal Failure Progression in Chronic Kidney Disease Using Integrated Intelligent Fuzzy Expert System. *Computational and Mathematical Methods in Medicine*, *2016*, 1–9. doi: 10.1155/2016/6080814.
127. Taal, M., & Brenner, B. (2006). Predicting initiation and progression of chronic kidney disease: Developing renal risk scores. *Kidney International*, *70*(10), 1694–1705. doi: 10.1038/sj.ki.5001794.
128. Cerqueira, D. C., Soares, C. M., Silva, V. R., Magalhães, J. O., Barcelos, I. P., Duarte, M. G., … Oliveira, E. A. (2014). A Predictive Model of Progression of CKD to ESRD in a Predialysis Pediatric Interdisciplinary Program. *Clinical Journal of the American Society of Nephrology*, *9*(4), 728–735. doi: 10.2215/cjn.06630613.
129. Winter, W. D., Dejongh, J., Post, T., Ploeger, B., Urquhart, R., Moules, I., … Danhof, M. (2006). A Mechanism-based Disease Progression Model for Comparison of Long-term Effects of Pioglitazone, Metformin and Gliclazide on Disease Processes Underlying Type 2 Diabetes Mellitus. *Journal of Pharmacokinetics and Pharmacodynamics*, *33*(3), 313–343. doi: 10.1007/s10928-006-9008-2.
130. Ito, K., Ahadieh, S., Corrigan, B., French, J., Fullerton, T., & Tensfeldt, T. (2010). Disease progression meta-analysis model in Alzheimers disease. *Alzheimers & Dementia*, *6*(1), 39–53. doi: 10.1016/j.jalz.2009.05.665.
131. Liu, Y.-Y., Ishikawa, H., Chen, M., Wollstein, G., Schuman, J. S., & Rehg, J. M. (2013). Longitudinal Modeling of Glaucoma Progression Using 2-Dimensional

Continuous-Time Hidden Markov Model. *Advanced Information Systems Engineering Lecture Notes in Computer Science*, 444–451. doi: 10.1007/978-3-642-40763-5_55.

132. Jackson, C. H., Sharples, L. D., Thompson, S. G., Duffy, S. W., & Couto, E. (2003). Multistate Markov models for disease progression with classification error. *Journal of the Royal Statistical Society: Series D (The Statistician)*, *52*(2), 193–209. doi: 10.1111/1467-9884.00351.
133. Zhou, J., Liu, J., Narayan, V. A., & Ye, J. (2012). Modeling disease progression via fused sparse group lasso. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 12*. doi: 10.1145/2339530.2339702.
134. Wang, X., Sontag, D., & Wang, F. (2014). Unsupervised learning of disease progression models. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 14*. doi: 10.1145/2623330.2623754.
135. Choi, E., Du, N., Chen, R., Song, L., & Sun, J. (2015). Constructing Disease Network and Temporal Progression Model via Context-Sensitive Hawkes Process. *2015 IEEE International Conference on Data Mining*. doi: 10.1109/icdm.2015.144.
136. Obrador, G. T., Mahdavi-Mazdeh, M., & Collins, A. J. (2011). Establishing the Global Kidney Disease Prevention Network (KDPN): A Position Statement From the National Kidney Foundation. *American Journal of Kidney Diseases*, *57*(3), 361–370. doi: 10.1053/j.ajkd.2010.12.006.
137. Rucci, P., Mandreoli, M., Gibertoni, D., Zuccala, A., Fantini, M. P., Lenzi, J., … Flachi, M. (2013). A clinical stratification tool for chronic kidney disease progression rate based on classification tree analysis. *Nephrology Dialysis Transplantation*, *29*(3), 603–610. doi: 10.1093/ndt/gft444.
138. Levin, A., & Stevens, P. E. (2011). Early detection of CKD: the benefits, limitations and effects on prognosis. *Nature Reviews Nephrology*, *7*(8), 446–457. doi: 10.1038/nrneph.2011.86.
139. Levey, A. S., Becker, C., & Inker, L. A. (2015). Glomerular Filtration Rate and Albuminuria for Detection and Staging of Acute and Chronic Kidney Disease in Adults. *Jama*, *313*(8), 837. doi: 10.1001/jama.2015.0602.
140. Abraham, A. G., Schwartz, G. J., Furth, S., Warady, B. A., & Muñoz, A. (2009). Longitudinal Formulas to Estimate GFR in Children with CKD. *Clinical Journal of the American Society of Nephrology*, *4*(11), 1724–1730. doi: 10.2215/cjn.01860309.
141. Coresh, J., Turin, T. C., Matsushita, K., Sang, Y., Ballew, S. H., Appel, L. J., … Levey, A. S. (2014). Decline in Estimated Glomerular Filtration Rate and Subsequent Risk of End-Stage Renal Disease and Mortality. *Jama*, *311*(24), 2518. doi: 10.1001/jama.2014.6634.
142. Klein, M. (2012). GFR Decline and Mortality Risk among Patients with Chronic Kidney Disease. *Yearbook of Medicine*, *2012*, 161–162. doi: 10.1016/j.ymed.2012.08.020.
143. Hallan, S. I., Matsushita, K., Sang, Y., Mahmoodi, B. K., Black, C., Ishani, A., … For The Chronic Kidney Disease Prognosis Consortium. (2012). Age and Association

of Kidney Measures With Mortality and End-stage Renal Disease. *Jama*, *308*(22), 2349. doi: 10.1001/jama.2012.16817.

144. Nitsch, D., Grams, M., Sang, Y., Black, C., Cirillo, M., Djurdjev, O., … Hemmelgarn, B. R. (2013). Associations of estimated glomerular filtration rate and albuminuria with mortality and renal failure by sex: a meta-analysis. *Bmj*, *346*(jan29 1). doi: 10.1136/bmj.f324.

145. Heerspink, H. J. L., Weldegiorgis, M., Inker, L. A., Gansevoort, R., Parving, H.-H., Dwyer, J. P., … Zeeuw, D. D. (2014). Estimated GFR Decline as a Surrogate End Point for Kidney Failure: A Post Hoc Analysis From the Reduction of End Points in Non–Insulin-Dependent Diabetes With the Angiotensin II Antagonist Losartan (RENAAL) Study and Irbesartan Diabetic Nephropathy Trial (IDNT). *American Journal of Kidney Diseases*, *63*(2), 244–250. doi: 10.1053/j.ajkd.2013.09.016.

146. Inker, L. A., Heerspink, H. J. L., Mondal, H., Schmid, C. H., Tighiouart, H., Noubary, F., … Levey, A. S. (2014). GFR Decline as an Alternative End Point to Kidney Failure in Clinical Trials: A Meta-analysis of Treatment Effects From 37 Randomized Trials. *American Journal of Kidney Diseases*, *64*(6), 848–859. doi: 10.1053/j.ajkd.2014.08.017.

147. Wainberg, M., Merico, D., Delong, A., & Frey, B. J. (2018). Deep learning in biomedicine. *Nature Biotechnology*, *36*(9), 829–838. doi: 10.1038/nbt.4233.

148. Decruyenaere, A., Decruyenaere, P., Peeters, P., Vermassen, F., Dhaene, T., & Couckuyt, I. (2015). Prediction of delayed graft function after kidney transplantation: comparison between logistic regression and machine learning methods. *BMC Medical Informatics and Decision Making*, *15*(1). doi: 10.1186/s12911-015-0206-y.

149. Tekale, S., Shingavi, P., & Wandhekar, S. (2018). Prediction of Chronic Kidney Disease Using Machine Learning Algorithm. *Ijarcce*, *7*(10), 92–96. doi: 10.17148/ijarcce.2018.71021.

150. Roberts, M. C., & Evans, S. C. (2013). Using the International Classification of Diseases System (ICD-10). *Psychologists Desk Reference*, 72–76. doi: 10.1093/med:psych/9780199845491.003.0013.

151. Nelli, F. (2015). Machine Learning with scikit-learn. *Python Data Analytics*, 237–264. doi: 10.1007/978-1-4842-0958-5_8.

152. Kazancioğlu, R. (2013). Risk factors for chronic kidney disease: an update. *Kidney International Supplements*, *3*(4), 368–371. doi: 10.1038/kisup.2013.79.

153. Chen, C., Zhang, Q., Ma, Q., & Yu, B. (2019). LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion. *Chemometrics and Intelligent Laboratory Systems*, *191*, 54–64. doi: 10.1016/j.chemolab.2019.06.003

CURRICULUM VITAE

Shaopeng Gu

605-691-1562 | https://www.linkedin.com/in/shaopenggu1990/ |
shaopenggu190@gmail.com

EDUCATION

M.S. in Mathematics – Statistics Specialization **May 2017-Dec 2019**

South Dakota State University, Brookings, SD

- Coursework: Statistical Inference, Regression analysis, Computational Intelligence, Statistical Programming, Measure & Probability Theory, Advanced Calculus

B.S. in Electrical Engineering

South Dakota State University, Brookings, SD

EMPLOYMENT

Computational Bioinformatics Analyst **Sep 2019—present**

Sanford Health

Graduate Research Assistant **May 2018—Aug 2019**

Bioinformatics and Mathematical Biosciences Lab, South Dakota State University

- Using machine learning methods to analyze and predict the outcomes of EMR data
- Develop a feature selection Python package for DNA, RNA and Protein sequence
- research and develop RNA-seq pipeline tools

Graduate Research Assistant **Aug 2017—May 2018**

Department of Electrical Engineering and Computer Science, South Dakota State University

- Develop machine learning methods to analyze electricity consuming data
- Develop computational tool to identify complex attack in smart grid security

PUBLICATIONS

1. Russel Wilke, Mohammad Qamar, Roxana Lupu, Shaopeng Gu, Jing Zhao, Chronic Kidney Disease in Agricultural Communities. ***The American Journal of Medicine***. doi: 10.1016/j.amjmed.2019.03.036
2. Zhao Jing, Shaopeng Gu, Adam McDermaid. Predicting outcomes of chronic kidney disease from EMR data based on Random Forest Regression. ***Mathematical Biosciences***. doi: 10.1016/j.mbs.2019.02.001
3. Adam McDermaid, Xin Chen, Yiran Zhang, Cankun Wang, Shaopeng Gu, Juan Xie, Qin Ma. A new machine learning-based framework for mapping uncertainty analysis in RNA-Seq read alignment and gene expression estimation. ***Frontiers in Genetics***. doi: 10.3389/fgene.2018.00313
4. Ashish Dubey, Nirmal Adhikari, Swaminathan Venkatesan, Shaopeng Gu, Devendra Khatiwada, Qi Wang, Lal Mohammad, Mukesh Kumar, Qiquan Qiao, Shelf life stability comparison in air for solution processed pristine PDPP3T polymer and doped spiro-OMeTAD as hole transport layer for perovskite solar cell. ***Data in Brief***. doi: 10.1016/j.dib.2016.02.021
5. Nirmal Adhikari, Ashish Dubey, Eman A. Gaml, Bjorn Vaagensmith, Khan Mamun Reza, Sally Adel Abdelsalam Mabrouk, Shaopeng Gu, Jiantao Zai, Xuefeng Qian, Qiquan Qiao, Crystallization of Perovskite Film for Higher Performance Solar Cells by Controlling Water Concentration in Methyl Ammonium Iodide Precursor Solution. ***Nanoscale***. doi: 10.1039/c5nr06687e
6. Ashish Dubey, Nirmal Adhikari, Swaminathan Venkatesan, Shaopeng Gu, Devendra Khatiwada, Qi Wang, Lal Mohammad, Mukesh Kumar, Qiquan Qiao, Solution processed pristine PDPP3T polymer as hole transport layer for efficient perovskite solar cells with slower degradation. ***Solar Energy Materials and Solar Cells***. doi: 10.1016/j.solmat.2015.10.008

SKILLS

- 5+ years working of programming languages (**Python**, R, SQL, MATLAB, SAS, C++, Perl)

- 2+ years of large-scale data mining and analysis with statistical & network modeling and machine learning methods
- 2 years of mathematical & statistical algorithm development
- 1 year of next-generation sequencing data analysis