Studies in Big Data 103

Sanjiban Sekhar Roy
Y.-H. Taguchi *Editors*

Handbook of Machine Learning Applications for Genomics

Springer

Studies in Big Data

Volume 103

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

The series "Studies in Big Data" (SBD) publishes new developments and advances in the various areas of Big Data- quickly and with a high quality. The intent is to cover the theory, research, development, and applications of Big Data, as embedded in the fields of engineering, computer science, physics, economics and life sciences. The books of the series refer to the analysis and understanding of large, complex, and/or distributed data sets generated from recent digital sources coming from sensors or other physical instruments as well as simulations, crowd sourcing, social networks or other internet transactions, such as emails or video click streams and other. The series contains monographs, lecture notes and edited volumes in Big Data spanning the areas of computational intelligence including neural networks, evolutionary computation, soft computing, fuzzy systems, as well as artificial intelligence, data mining, modern statistics and Operations research, as well as self-organizing systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

The books of this series are reviewed in a single blind peer review process.

Indexed by SCOPUS, EI Compendex, SCIMAGO and zbMATH.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at https://link.springer.com/bookseries/11970

Sanjiban Sekhar Roy · Y.-H. Taguchi
Editors

Handbook of Machine Learning Applications for Genomics

Springer

Editors
Sanjiban Sekhar Roy
School of Computer Science
and Engineering
Vellore Institute of Technology University
Vellore, Tamil Nadu, India

Y.-H. Taguchi
Department of Physics
Chuo University
Tokyo, Japan

ISSN 2197-6503 ISSN 2197-6511 (electronic)
Studies in Big Data
ISBN 978-981-16-9157-7 ISBN 978-981-16-9158-4 (eBook)
https://doi.org/10.1007/978-981-16-9158-4

This book is dedicated to my father,

Niharendu Shekhar Roy,

With love.

Growing up, my father taught me many valuable lessons, specifically: to be a fighter in every tough situation in life, to be kind to everyone, and to never give up. His words have shaped my life completely. My father is the kindest person ever I have seen; never have seen him talking bad about others, always saw him helping others , and he is brutally honest in every situation no matter what the cost to himself.

I love you, dad.

—Sanjiban Sekhar Roy

Preface

We live in the era of machine learning, and the applications of machine learning are not only limited to computer science it is now being used in a variety of domains. Among all, genomics has found huge applications of machine learning for its advancements. In the recent decade, the fast expansion of biological data is an important step for the development of genomics. As the use of machine learning increases towards the massive data of genome, it further helps us to enhance our knowledge about genomics. Machine learning perhaps can interpret the large genome data in the best possible ways and help to annotate a broad variety of sequences of the genome. The topics that have been included in this book will cater interest to academicians, clinical practitioners working in the field of functional genomics and machine learning. Also, graduate, postgraduates and Ph.D. scholars working in these fields will immensely be benefited. This edited book has dealt with the following chains of works on the applications of machine learning in the field of genomics.

- Multiomics data analysis of cancers using tensor decomposition and principal component analysis-based unsupervised feature extraction.
- Machine Learning for Protein Engineering
- Statistical Relational Learning for Genomics Applications: A State-of-the-Art Review
- A Study of Gene Characteristics and Their Applications using Deep Learning
- Computational Biology in the lens of CNN
- Drug repositioning using tensor decomposition-based unsupervised feature extraction
- Challenges of long non-coding RNAs in human disease diagnosis and therapies: Bio-computational approaches
- Protein sequence classification using Convolutional Neural Network and Natural Language Processing
- Machine Learning for Metabolic Networks Modelling: A State-of-the-Art Survey
- Tensor decomposition and principal component analysis-based unsupervised feature extraction applied to single cell analysis
- Machine learning: a tool to shape the future of medicine

The intention of compiling this book is to present a good idea about both theory and practice related to the above-mentioned applications before the readers by showcasing the usages of machine learning, and deep learning in genomics data and other related fields. Besides, the book shall be useful to the people working with predictive modelling of genomic research. It will immensely help people working in medical industries, research scholars in the educational institute and scientists in genom research labs.

We hope that readers will be benefited significantly in learning about the state of the art of machine learning applications in the domain of genomics and its related field.

Keep reading, learning and inquiring.

Vellore, Tamil Nadu, India
September 2020

Sanjiban Sekhar Roy
Associate Professor
sanjibansroy@ieee.org

Contents

Multiomics Data Analysis of Cancers Using Tensor Decomposition and Principal Component Analysis Based Unsupervised Feature Extraction

Y.-H. Taguchi

Abstract Multiomics data analyses using principal component analysis as well as tensor decomposition based unsupervised feature extraction were performed toward various cancers. It turned out that it is a very effective method to understand how multiomics contributes to cancer progressions.

1 Introduction

Cancers are ever difficult problems to be attacked by bioinformatics because of its complexity. In contrast to general diseases, they are deeply related with the malfunctions of genetic systems. They are associated with aberrant gene expression, aberrant promoter methylation, aberrant microRNA expression and so on. No single omics data set is enough to understand the biology of cancers. In this regard, so-called multiomics analysis is highly required.

Multiomics analysis means integration of various omics data, including gene expression, promoter methylation and microRNA expression. Since aberrant expression of cancers can be observed in multiomics levels, multiomics data analysis is very fitted to study cancers. In this chapter, we introduce how we can integrate multiomics data set of cancers, using principal component analysis as well as tensor decomposition, which are called as principal component analysis and tensor decomposition based unsupervised feature extraction.

Y.-H. Taguchi (✉)
Department of Physics, Chuo University, Tokyo, Japan
e-mail: tag@granular.com

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_1

2 Integrated Analysis of SNP and DNA Methylation

SNP, which is an abbreviation of Single Nucleotide Polymorphism, means the replacement of single nucletide in DNA. Generally, SNP is believed to cause the alteration of amino acid sequence of protein which is translated from mRNA transcripted from DNA, since SNP can change codon. Nevertheless, most of SNPs known to affect cancers are not located within exon that is a part of DNA converted to protein but located outside exon. Thus, the functionality of SNP toward cancer progression has not yet been well known. In this section, I would like to report how we can interpret of function in cancer of SNPs based upon the relationship with DNA metylation.

Previously [3], we have performed integrated analysis of SNP and DNA methylation in esophageal squamous cell carcinoma (ESCC). More than 90% of esophageal cancer is ESCC [1]. Tobacco (smoking or chewing) and alcohol are known to be two major risk factors for ESCC [5]. Oka et al. [8] found aberrantly methylated 37 genes in ESCC, five genes among which were found to be related to period of smoking. Nevertheless, how aberrant methylation affects the progression of ESCC was unclear. Thus, if we can relate aberrant methylation in ESCC to SNP in ESCC, which is naturally a cause of general cancers, it is very helpful to understand the mechanism by which aberrant methylation of ESCC can cause ESCC.

The data set we analyzed in the previous study [3] was composed of 30 blood samples, 30 normal tissues and 30 ESCCs of ESCC patients. DNA methylation as well as SNPs were measured by two SNP arrays, Nsp with 262,339 probes and Sty with 238,379 probes, respectively. From the data science point of views, it is extreme *large p small n* problems, since the number of p (=features) is more than 10^3 larger than that of$n$ (=samples); standard methods adapted to tackle *large p small n* [2], which assumes at most $p/n \sim 10$, cannot be effective.

In order to tackle this difficult problem, we invented principal component analysis (PCA) based unsupervised feature extraction (FE) [12]. Suppose that $x_{ij} \in \mathbb{R}^{N \times M}$ represents amount of SNP or DNA methylation associated with ith genes at jth sample among $90(=M)$ samples described in the above; $1 \leq j \leq 30$ stand for blood, $31 \leq j \leq 60$ stand for normal tissues, and $61 \leq j \leq 90$ stand for ESCC. N is the number of probes, $N = 262{,}339$ for Nsp array and $N = 238{,}379$ for Sty array. PCA was applied to x_{ij} such that PC loading was attributed to samples whereas PC score was attributed to probes. Please note that is differs from the usual usage of PCA where PC scores are attributed to samples while PC loading are attributed to genes. These two fundamentally differ from each other; $\sum_j x_{ij} = 0$ is assumed for usual usage of PCA whereas $\sum_i x_{ij} = 0$ is assumed for the current usage of PCA [12]. Since these two result in distinct results, they cannot be reproduced from another implementation. $u_{\ell i} \in \mathbb{R}^{N \times N}$ and $v_{\ell j} \in \mathbb{R}^{M \times M}$ are supposed to represent ℓth PC scores attributed to ith gene and PC loading attributed to jth sample.

First of all, we need to find which $v_{\ell j}$ represents the distinction between blood, normal tissue and ESCC samples. Since blood is generally believed to be mutation free, aberrant SNP and DNA methylation identified by the comparisons between blood and ESCC are supposed to be caused by somatic mutation and methylation

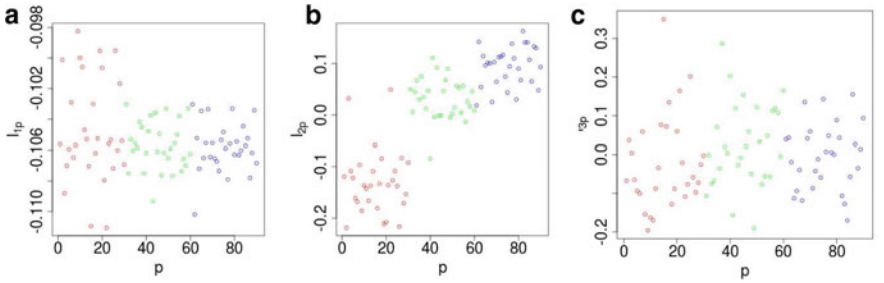

Fig. 1 PC loading attributed to samples, j, $v_{\ell j}$ when PCA was applied to SNP data measured by Nsp array. **a** $\ell = 1$ **b** $\ell = 2$ **c** $\ell = 3$. $1 \leq j \leq 30$ (red open circle): blood, $31 \leq j \leq 60$ (green open circle): normal tissue, $61 \leq j \leq 90$ (blue open circle): ESCC

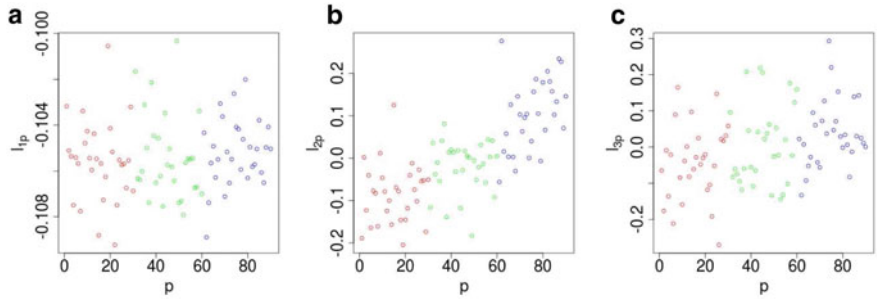

Fig. 2 PC loading attributed to samples, j, $v_{\ell j}$ when PCA was applied to DNA methylation data measured by Nsp array. Other notations are the same as Fig. 1

through cancer progression. Thus, identification of these anomaly would help us to identify genes associated with cancer progression. In addition to this, since normal tissue samples are taken from neighboring normal tissues adjusted to ESCC by surgery, SNPs and DNA methylation in normal tissue are expected in between ESCC and blood sample. Thus, the dependence of $v_{\ell j}$ upon j is expected to be monotonic from blood to ESCC through normal tissue. Figure 1 shows the first three PC loading $v_{\ell j} (1 \leq \ell \leq 3)$ when PCA was applied to SNP data measured by Nsp array. It is obvious that only second PC loading, v_{2j}, exhibits dependence upon distinction between blood, normal tissue and ESCC. Interestingly, contribution of v_{2j} is as small as 3%. In conventional usage of PCA, components having such a small amount of contribution should be discarded. Nonetheless, as can be seen in the following, this small amount of contribution is biologically important.

Next, we investigated PC loading $v_{\ell j}$ when PCA was applied to DNA methylation measured by Nsp array (Fig. 2). Again, only second PC loading, v_{2j}, associated with contribution of as small as 3%, exhibits dependence upon distinction between blood, normal tissue and ESCC.

Now we can select genes associated with the monotonic dependence from blood to ESCC through normal sample using u_{2i} associated with v_{2j}, since is having larger

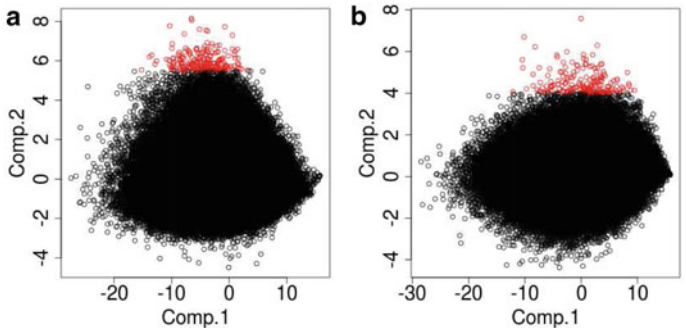

Fig. 3 PC score attributed to genes, i, $u_{\ell i}$ when PCA was applied to **a** SNP and **b** DNA methylation data measured by Nsp array. Horizontal and vertical axes represent $\ell = 1, 2$, respectively. Red parts are selected top 300 outliers along vertical axis

contribution to u_{2i} is expected to have more association with the monotonic dependence. Figure 3 shows the scatter plots of $u_{\ell i} (\ell = 1, 2)$. Since we are interested in outlier genes along u_{2i}, we selected top 50, 100, 150, 200, 250, and 300 outlier genes.

The first evaluation of if integrated analysis is successful is to see how many probes are selected between SNPs and DNA methylation. N is 10^5 whereas selected genes are 10^2, thus it is unlikely there are commonly selected probes by chance, since DNA methylation and SNP are completely different measurement. In spite of that, there are 68 probes commonly selected among 300 outlier probes selected in SNP and DNA methylation. Considering the only 10^2 probes selected among 10^5 probes, as many as 68 probes commonly selected cannot be accidental. This excellent overlaps of selected probes between SNP and DNA methylation suggests the success of our integrated analysis.

In order to further confirm the success of our analysis, we compare x_{ij}s of commonly selected probed between SNP and DNA methylation. Before drawing these, we screened probes where all three associated P-values adjusted by the BH criterion [12] are less than 0.05, when three pairwise one-sided t-tests (tumor tissue vs. normal tissue, normal tissue vs. blood, tumor tissue vs. blood) are applied. Figure 4 shows the scatter plot of commonly selected x_{ij} Probes between top N outliers in SNPs and DNA methlylations ($N = 50, 100, 150, 200, 250, 300$). Although there are no reasons that they must be highly correlated, it is obvious that they are highly correlated and it cannot be accidental. This strong correlation also suggests the success of our integrated analysis. It is likely that somatic mutation associated with these genes are driven by DNA methylation; it is coincident with the observation that methylation caused by smoking is a risk factor of ESCC.

We have repeated the same analysis to data measured by Sty array. Figure 5 shows the PC loading $v_{\ell j} (1 \leq \ell \leq 4)$ when PCA was applied to SNPs measured by Sty array. In contrast to Nsp array, two PC loading, v_{3j} and v_{4j}, are dependent upon the distinction between blood, normal tissue, and ESCC; contributions of these two PC loading are as small as 2% and 1%, respectively. Figure 6 shows the PC loading

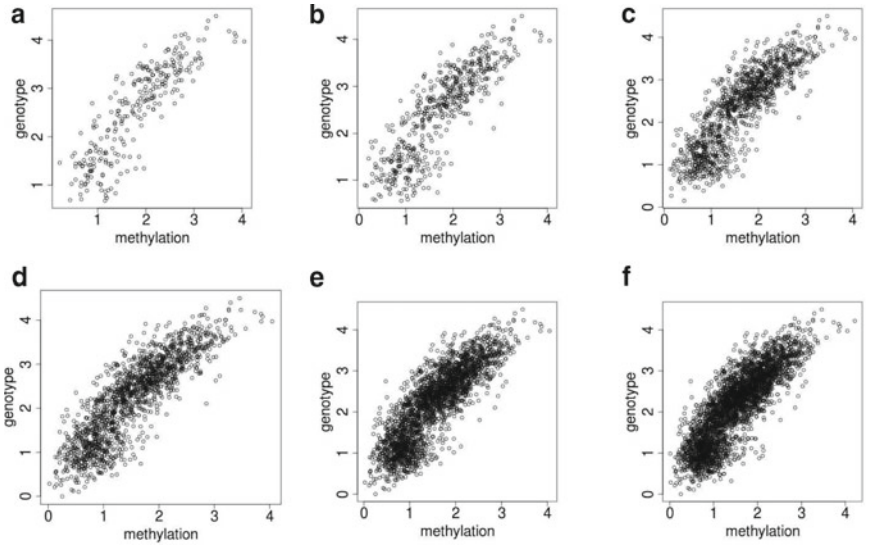

Fig. 4 Scatter plots of commonly selected x_{ij} between DNA methylation and SNPS in **a** top 50 outliers, **b** top 100 outliers, **c** top 150 outliers, **d** top 200 outliers, **e** top 250 outliers, **f** top 300 outliers. Nsp array was used

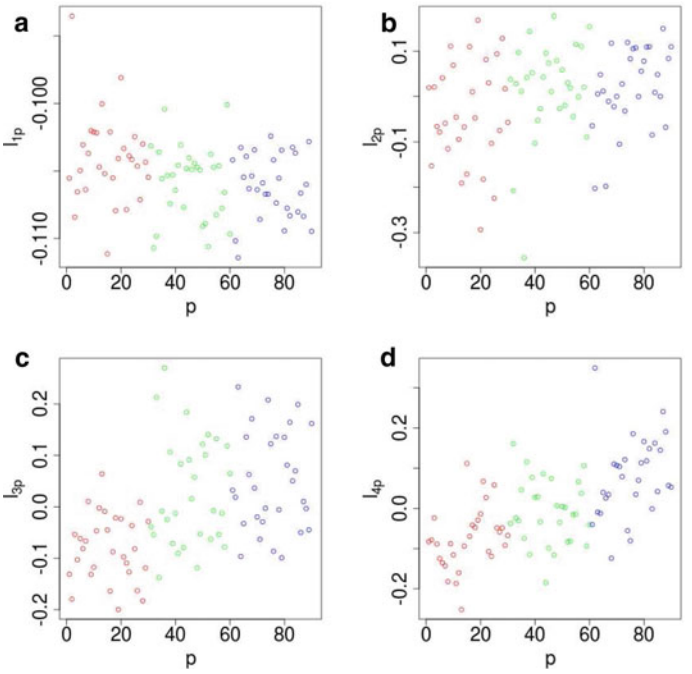

Fig. 5 PC loading attributed to samples, j, $v_{\ell j}$ when PCA was applied to SNP data measured by Sty array. **a** $\ell = 1$ **b** $\ell = 2$ **c** $\ell = 3$ **d** $\ell = 4$. Other notations are the same as Fig. 1

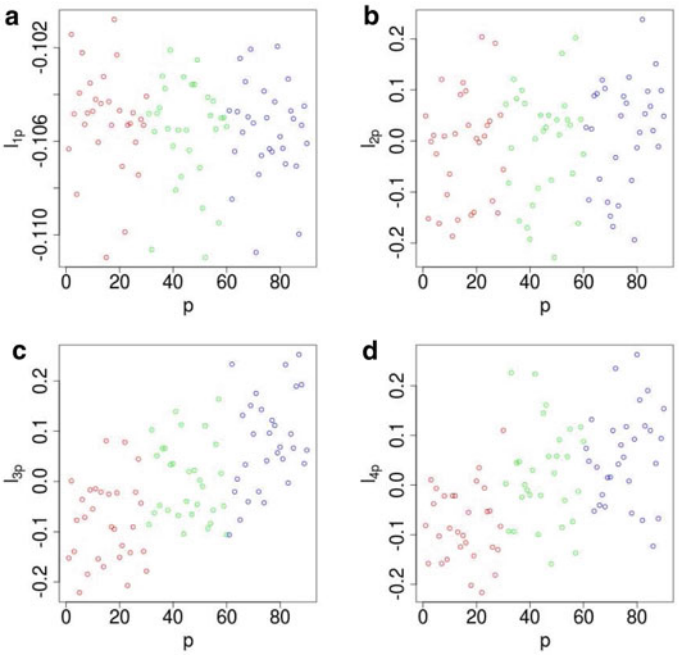

Fig. 6 PC loading attributed to samples, j, $v_{\ell j}$ when PCA was applied to DNA methylation data measured by Sty array. **a** $\ell = 1$ **b** $\ell = 2$ **c** $\ell = 3$ **d** $\ell = 4$. Other notations are the same as Fig. 1

$v_{\ell j} (1 \leq \ell \leq 4)$ when PCA was applied to DNA methylation measured by Sty array. Again, two PC loading, v_{3j} and v_{4j}, are dependent upon the distinction between blood, normal tissue, and ESCC; contributions of these two PC loading are as small as 1%.

Now we would like to decide which pairs of PC scores attributed to SNPs and DNA methylation should be used for selecting probes. Since there are no one to one correspondence of PC loading between SNPs and methylation in contrast to Nsp array, we tried all of pairs and found that the best pairs are The 4th PC score, u_{4i}, for SNPS versus the 3rd PC score, u_{3i}, for DNA methylation (pair 1) and the 3rd PC score, u_{3i}, for SNPS versus the 4th PC score, u_{4i}, for DNA methylation (pair 2). Figure 7 shows the scatter plots of $u_{\ell i}$s as vertical axes whereas u_{1i}s as horizontal axes when top 300 outlier probes are selected. Between them, 81 and 50 probes were commonly selected for pair 1 and pair 2, respectively. Again, considering the number of probes of 10^5 whereas the selected probes are 10^2, this number of commonly selected probes cannot be accidental. Thus, again, our integrated strategy could successfully identify probes associated with SNPs and DNA methylation simultaneously.

Again in order to confirm the success of proposed analysis, we also tried to see if commonly selected x_{ij} are significantly correlated or not between SNPs and DNA methylation for pair 1 (Fig. 8). It is obvious that they are highly correlated. Thus, from this point, our analysis is very successful.

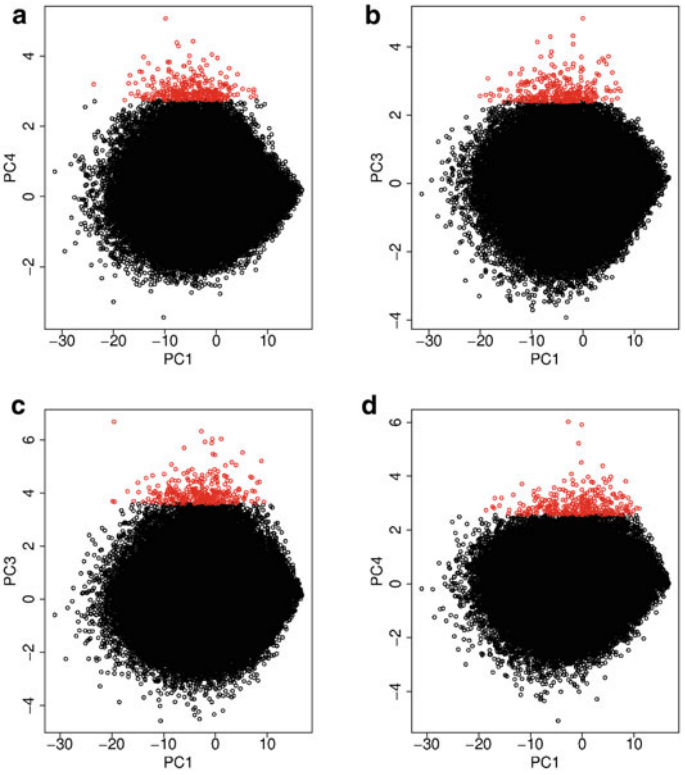

Fig. 7 PC score attributed to genes, i, $u_{\ell i}$ when PCA was applied to **a**, **b** SNP and **c**, **d** DNA methylation data measured by Nsp array. Horizontal axes represent $\ell = 1$. Vertical axes represent **a** $\ell = 4$, **b** $\ell = 3$, **c** $\ell = 3$, **d** $\ell = 4$. Red parts are selected top 300 outliers along vertical axis

In the above, independent of employed array, our strategy successfully limited number ($\sim 10^2$) of selected probes associated with SNPs and DNA methylation coincident with distinction among blood, normal tissue and ESCC while SNPs and DNA methylation are mutually correlated. Before going further, we would like to confirm the superiority of the proposed strategy toward other conventional methods. We have commonly selected 68, 81, and 50 probes between top 300 outliers in SNPs and DNA methylation as shown in Figs. 3 and 7 using three integrated analyses. We would like to evaluate other methods based upon the number of commonly selected probes between top 300 outliers in SNPs and DNA methylation. Generally speaking, since there are three categorical classes, it is not straight forward to select probes coincident with the distinction between three classes. Nevertheless, fortunately, we have already known that there are limited number of probes whose SNPs as well as DNA methylation are well ordered from blood to ESCC through normal tissue. Thus, we decided to select probes whose values are also coincident with this order. In order that, we introduce dummy variable, y_j as

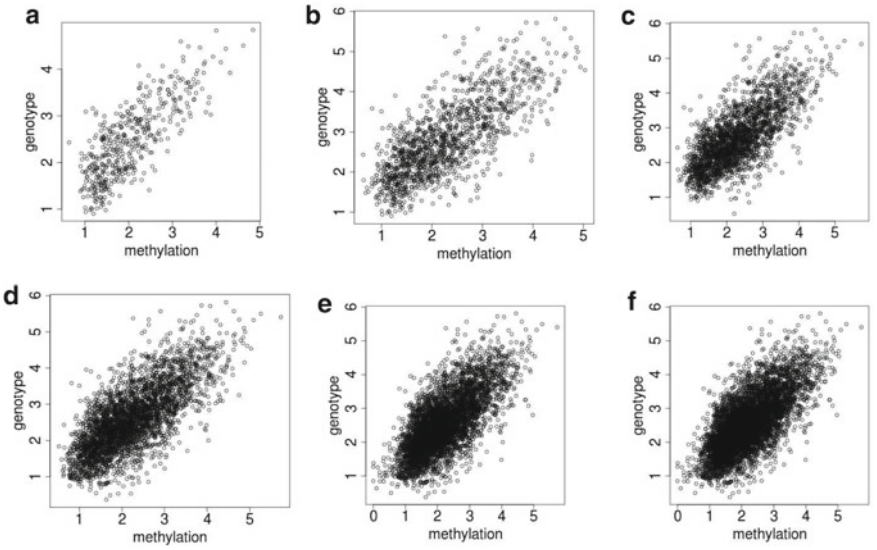

Fig. 8 Scatter plots of commonly selected x_{ij} between DNA methylation and SNPS in **a** top 50 outliers, **b** top 100 outliers, **c** top 150 outliers, **d** top 200 outliers, **e** top 250 outliers, **f** top 300 outliers. Sty array was used. Only for pair 1

$$y_j = \begin{cases} 1 \ \ (1 \leq j \leq 30), \ \text{blood} \\ 2 \ (31 \leq j \leq 60), \ \text{normal tissue} \\ 3 \ (61 \leq j \leq 90), \ \text{ESCC} \end{cases} \tag{1}$$

At first, we computed the Pearson's correlation coefficient between X_{ij} and y_j for each i (probe) and listed top ranked 300 probes associated with larger correlation coefficients. For Nsp and Sty arrays, there were 49 and 14 probes commonly selected between SNPs and DNA methylation. Since these numbers are less than those achieved by the proposed method, 68 (for Nsp array), 81 and 50 (for Sty array), Pearon correlation coefficients have less ability to select genes commonly between SNPs and DNA methylation than the proposed strategy. Since one might wonder if not the correlation with three integers, 1, 2 and 3, but the rank itself is considered, correlation based method could outperform the proposed strategy. Thus, we replace Pearson correlation coefficients with Spearman correlation coefficients and repeated the same procedure. This results in 39 and 18 probes commonly selected between SNPs and DNA methylation for Nsp and Sty arrays, respectively. Thus, again, correlation coefficients methods could not outperform the proposed strategy. One might still wonder that correlation coefficient methods are too simple to compete with our proposed strategy. Thus, we next tried more advanced method, partial least squares (PLS) based feature selection. PLS is a kind of liner regression, but regression analysis was not performed using variables, but performed using principal component. Since our strategy based upon PCA did work pretty well, PLS is expected to also

Table 1 Enriched terms when selected genes are uploaded to g:profiler

Terms	ID	Adjusted P-value
GO MF		
Fibroblast growth factor receptor binding	GO:0005104	7.331×10^{-3}
GO BP		
Cell adhesion	GO:0007155	1.723×10^{-2}
Biological adhesion	GO:0022610	1.859×10^{-2}
Morphogenesis	GO:0003007	2.986×10^{-2}
KEGG		
Pathways in cancer	KEGG:05200	2.293×10^{-3}

work well. PLS is mathematically formulated as

$$\max_{w_i} \sum_j \left(\sum_i w_i x_{ij} \right) y_j, \sum_i w_i^2 = 1 \tag{2}$$

and if limited number of is was selected, it can also work as feature selection. Here we employed implementation where categorical regression is possible [11]. Even using this advanced methods, the number of commonly selected probes are 7 and 13 for Nsp and Sty array, respectively. These numbers are even less than those achieved by much simpler correlation based methods. The reason why it is so difficult is rather obvious. The number of features (probes) is as many as 10^5 whereas the number of features selected is as small as 10^2; two set of selected features using two independent data set must be largely ovelapped. In this regard, our strategy is definitely the best one.

Now we would like to discuss the biological respects of selected probes. First of all, most of SNPs are not located in exon but located in intron, downstream and upstream regions. This suggests that most of selected SNPs associated with aberrant methylation do not affect amino acid sequence but the regulation of genes. Then we have uploaded 155 genes that have selected genes in exon, intron, downstream and upstream (see Additional file 2 [3]) to various enrichment servers. When they are uploaded to g:profiler [9], we got Table 1. It is obvious that the selected terms are related to cancers. Thus, it supports the conclusion that our integrated analysis is successful. When 155 genes are uploaded to Enrichr [4], huge number of transcription factors (TFs) are detected (Table 2). This suggests that SNPs might affect TF binding to genes.

Table 2 TFs identified in "ChEA 2016" category in Enrichr

AHR, AR, ARNT, CDX2, CJUN, CTBP1, CTBP2, CTNNB1, E2F1, EBF1, EP300, ERA, ESR1, EZH2, FOXA2, FOXM1, GBX2, JUN, LMO2, MITF, NANOG, NFE2L2, NR3C1, NRF2, OLIG2, P300, P53, PAX3-FKHR, PIAS1, PPAR, PPARD, RNF2, RUNX2, SMAD2, SMAD3, SMAD4, SMARCA4, SMARCD1, SMRT, SOX2, SOX9, STAT3, TAL1, TBX3, TCF3/E2A, TCF4, TCFCP2L1, TOP2B, TP53, VDR, WT1, YAP1, ZFP57, ZNF217,

3 Integrated Analysis of microRNA, mRNA and Metabolome

In the previous section, we demonstrated that our strategy could integrate SNPs and DNA methylation effectively. In this section, we applied our strategy to other combinations of omics data set: mRNA expression, miRNA expression and metabolome [6]. In this data set, there are 10 Intrahepatic cholangiocarcinoma (ICC) patients and 6 hepatocellular carcinoma (HCC) patients, from which both tumors and respective surrounding non-tumor tissues were taken. Thus, in total there are 32 samples. On the other hand, the number of features consider are 10^2 to 10^4; $x_{ij}^{\mathrm{mRNA}} \in \mathbb{R}^{58717\times 32}$, $x_{kj}^{\mathrm{miRNA}} \in \mathbb{R}^{60180\times 32}$, and $x_{mj}^{\mathrm{metabolome}} \in \mathbb{R}^{583\times 32}$. Thus, it is definitely *large p small n* problem. $\sum_i x_{ij}^{\mathrm{mRNA}} = \sum_k x_{kj}^{\mathrm{miRNA}} = 0$ and $\sum_i \left(x_{ij}^{\mathrm{mRNA}}\right)^2 = 58717$, $\sum_k \left(x_{kj}^{\mathrm{miRNA}}\right)^2 = 60180$ whereas $x_{mj}^{\mathrm{metabolome}}$ were not scaled since metabolome is expected to measure not relative values but absolute values. PC scores attributed to mRNA, miRNA and metabolome were $u_{\ell i}^{\mathrm{mRNA}} \in \mathbb{R}^{58717\times 32}$, $u_{\ell k}^{\mathrm{miRNA}} \in \mathbb{R}^{58717\times 32}$, $u_{\ell m}^{\mathrm{metabolome}} \in \mathbb{R}^{583\times 32}$, computed by PCA. PC loading attributed to mRNA, miRNA and metabolome were $v_{\ell j}^{\mathrm{mRNA}}$, $v_{\ell j}^{\mathrm{miRNA}}$, $v_{\ell j}^{\mathrm{metabolome}} \in \mathbb{R}^{32\times 32}$, also computed by PCA.

At first, we investigated the PC loading in order to identify which PC scores should be use for selecting mRNA, miRNA and metabomolome. Since the purpose of our analysis is integrated analysis of three omics data set, mRNA, miRNA and metabolome, we tried to find which pairs of PC loading, $v_{\ell j}^{\mathrm{mRNA}}$, $v_{\ell j}^{\mathrm{miRNA}}$, $v_{\ell j}^{\mathrm{metabolome}}$, are well coincident with each other. In order that, we computed absolute values of pairwise PC loading between 96 PC loading composed of 32 $v_{\ell j}^{\mathrm{mRNA}}$, 32 $v_{\ell j}^{\mathrm{miRNA}}$ and 32 $v_{\ell j}^{\mathrm{metabolome}}$ and generated hierarchical clustering using s Unweighted Pair Group Method using arithmetic Average (UPGMA, Fig. 9). Occasionally, since v_{1j}^{mRNA}, v_{2j}^{mRNA}, v_{1j}^{miRNA}, v_{2j}^{miRNA}, $v_{3j}^{\mathrm{metabolome}}$ are clustered together (red rectangular in Fig. 9) with relatively larger absolute correlation coefficient, we decided to employ these five PC loading.

Although these five PC loading are well correlated, we are not sure if they are distinct between four classes, HCC, ICC, and two representative normal tissues surrounding them. In order to address this point, we draw boxplots of these PC lading (Fig. 10) and compute P-values using categorical regression. It is obvious that they are distinct between, at least, some of four categorical classes. Thus our strategy

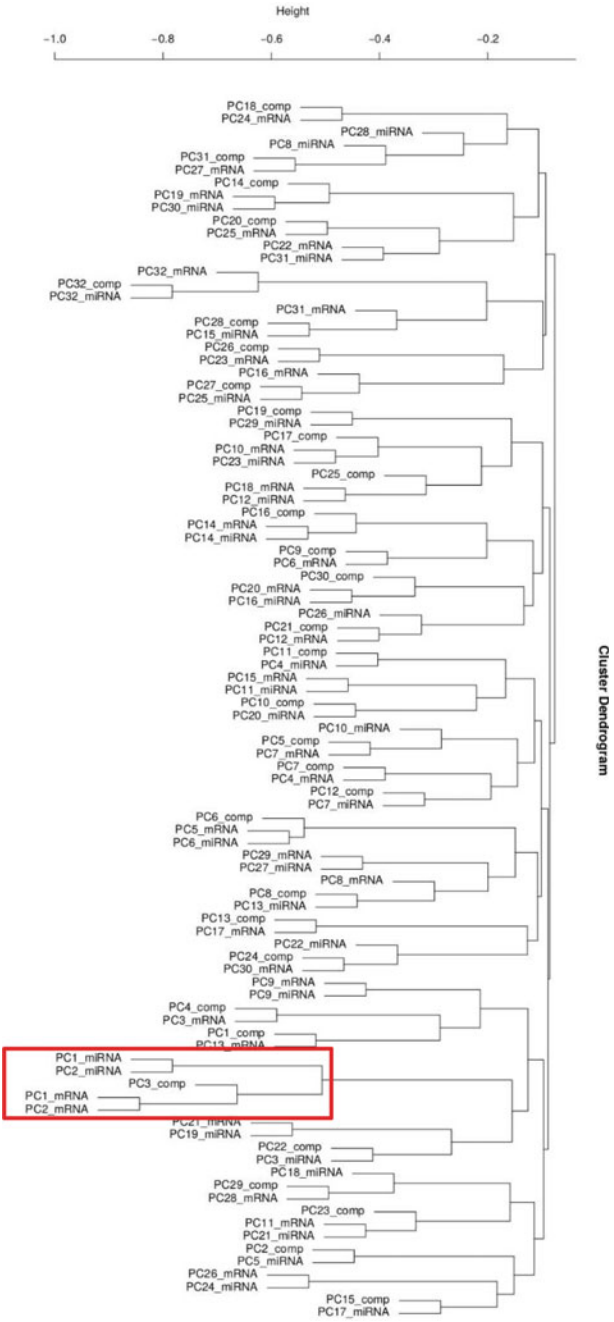

Fig. 9 Hierarchical clustering of PC loading, $v_{\ell j}^{\mathrm{mRNA}}$, $v_{\ell j}^{\mathrm{miRNA}}$, $v_{\ell j}^{\mathrm{metabolome}}$ using UPGMA. Regions surrounded by red open rectangular are employed as those coincident with one another

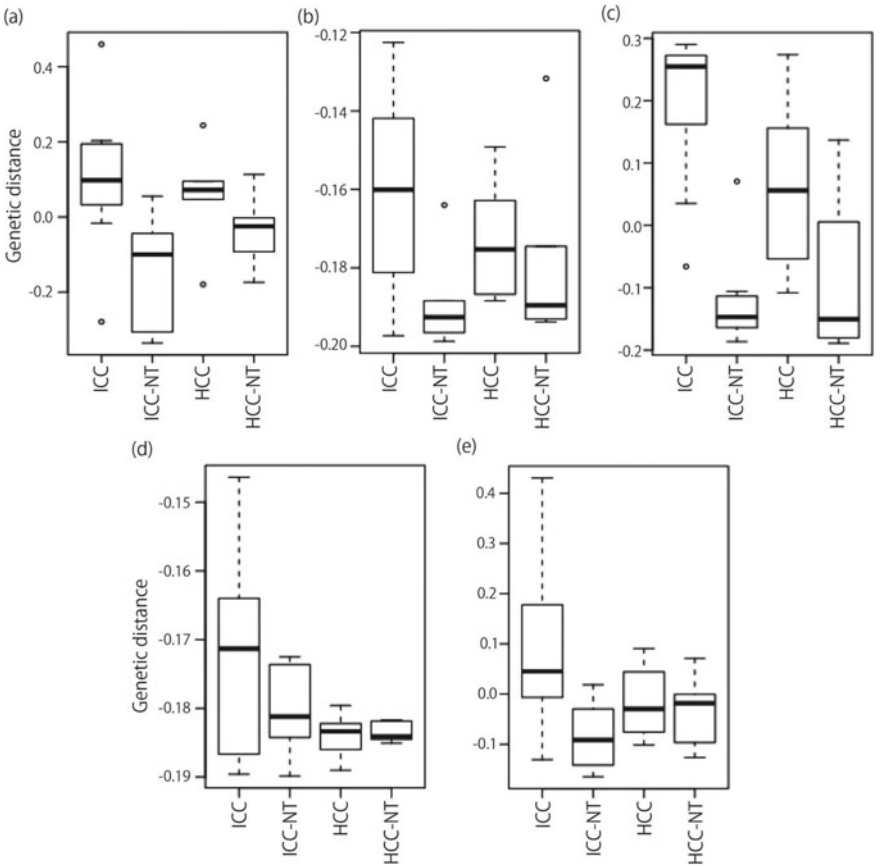

Fig. 10 Boxplots of PC loading. **a** $v_{3j}^{\text{metabolome}}$, $P = 8.68 \times 10^{-3}$ **b** v_{1j}^{mRNA}, $P = 1.69 \times 10^{-2}$ **c** v_{2j}^{mRNA}, $P = 3.98 \times 10^{-6}$ **d** v_{1j}^{miRNA}, $P = 4.74 \times 10^{-2}$ **e** v_{2j}^{miRNA}, $P = 9.42 \times 10^{-3}$. P-values are computed by categorical regression

could successfully identify PC loading that are mutually coincident with one another as well as distinct between some of four classes.

Now we can select mRNAs, miRNAs and compounds using PC scores, u_{1i}^{mRNA}, u_{2i}^{mRNA}, u_{1k}^{miRNA}, u_{2k}^{miRNA}, and $u_{3m}^{\text{metabolome}}$, that correspond to the selected PC loading, v_{1j}^{mRNA}, v_{2j}^{mRNA}, v_{1j}^{miRNA}, v_{2j}^{miRNA}, and $v_{3j}^{\text{metabolome}}$ (Fig. 11). Fifty three compounds, m, with $|u_{3m}| > 0.1$, 96 mRNAs, i, with $u_{1i} < -50$, and 286 miRNAs, k, with $\sqrt{u_{1i}^2 + u_{2i}^2} > 10$ were selected as outliers. Since for miRNAs multiple probes are attributed to one miRNA, 286 miRNA probes correspond to only 17 miRNAs.

One possible evaluation is to check if the selected mRNAs, miRNAs and metabolome can discriminate four classes correctly. In order that, we have done as follows. PC loading attributed to js, $v_{\ell j}^{\text{mRNA}}$, $v_{\ell j}^{\text{miRNA}}$, $v_{\ell j}^{\text{metabolome}}$, was recomputed

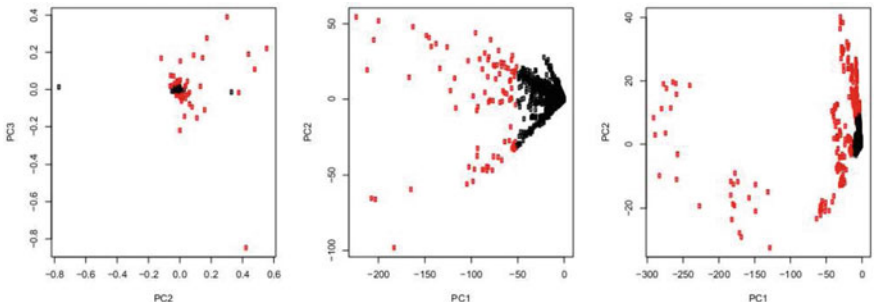

Fig. 11 **a** Horizontal axis $u_{2m}^{\text{metabolome}}$, vertical axis $u_{3m}^{\text{metabolome}}$ **b** horizontal axis u_{1i}^{mRNA}, vertical axis u_{2i}^{mRNA} **c** horizontal axis u_{1k}^{miRNA}, vertical axis u_{2k}^{miRNA}. Open red circles are selected **a** compounds, **b** mRNAs and **c** miRNAs

Table 3 Confusion matrix between true sample labels and predicted sample labels when LDA was applied to re-computed PC loading, $v_{\ell j}^{\text{metabolome}}$, using selected 53 metablome

		True		
Predicted		CNTL	HCC	ICC
	CNTL	14	0	2
	HCC	0	5	0
	ICC	2	1	8

Table 4 Confusion matrix between true sample labels and predicted sample labels when LDA was applied to re-computed PC loading, $v_{\ell j}^{\text{miRNA}}$, using selected 17 miRNAs

		True		
Predicted		CNTL	HCC	ICC
	CNTL	13	1	1
	HCC	2	4	1
	ICC	1	1	8

with using only selected 96 mRNAs, 17 miRNAs, and 53 compounds. Then optimal number of PC loading was used to discriminate four classes using liner discriminant analysis (LDA) with leave one out cross validation. For metabolome, the first 17 $v_{\ell j}^{\text{metabolome}}$s were optimal (Table 3, accuracy = 0.84). For miRNA, the first 6 $v_{\ell j}^{\text{metabolome}}$s were optimal (Table 4, accuracy = 0.78). Unfortunately, we could not well discriminate 32 samples using selected 96 mRNAs. Anyway, our strategy could successfully select limited number of miRNA can compounds that can discriminate three classes more ore less successfully.

4 Integrated Analysis of mRNA and miRNA of Kidney Cancer Using TD Bases Unsupervised FE

In the previous sections, individual omics data was separately analysed. Among the obtained PC loading and PC scores, limited number of them were selected based upon the coincidence between them. These strategy could work well, but yet anther strategy is possible; i.e., omics data was integrated not after before analysis. Tensor decomposition (TD) enables us to perform this strategy [12]. TD is an extension of matrix factorization (MF) applied to matrices toward tensors. In MF, a matrix was represented as a product two low ranked matrices. Similarly, in TD, tensors are represented as a product of lower ranked matrices and/or tensors.

Tensors are the extension of matrices that have only rows and columns so as to have more number of (i.e., more than two) rows and columns. Using matrices, we can represent the relationship between two properties, e.g., gene versus samples (patients), tensor enables us to relationship between more numbers of propertied, e.g., genes versus samples versus tissues. In this case, a tensor, $x_{ijk} \in \mathbb{R}^{N \times M \times K}$ can represent gene expression of ith gene of jth person's kth tissue. In this sense, tensor is more fitted to deal with multiomics data than matrix that can deal with only a combination of two properties, e.g., genes and samples.

On the other hand, since the number of distinct omics data is also distinct; the number of mRNA is 10^4, that of $miRNA$ is 10^3, whereas that of methylation site is as many as 10^5 to 10^6, tensor as is is not very fitted to include this kind of heterogeneous number of features. In this case, we can generate a tensor from multiomics data set, each of which is associated with the different number of features. In the recent study [7], we integrated mRNAs and miRNAs using tensors. The procedure is as follows.

- Suppose that $x_{ij} \in \mathbb{R}^{N \times M}$ represents expression of ith mRNA of jth sample, and $x_{kj} \in \mathbb{R}^{K \times M}$ represents expression of kth miRNA of jth sample.
- Define tensor $x_{ijk} = x_{ij} x_{kj} \in \mathbb{R}^{N \times M \times K}$
- Compute a tensor $x_{ik} = \sum_{j=1}^{M} \in \mathbb{R}^{M \times K}$ in order to reduce the computer memory required.
- Apply singular value decomposition SVD to x_{ik} and get $x_{ik} = \sum_{\ell=1}^{\min(N,K)} u_{\ell i} \lambda_\ell v_{\ell k}$
- Regenerate missing singular value vectors attributed to jth sample as $u_{\ell j} = \sum_i x_{ij} u_{\ell i}$ and $v_{\ell j} = \sum_k x_{kj} v_{\ell k}$.
- Identify ℓ that satisfies the following requirements,
  - $u_{\ell j}$ are highly distinct between healthy controls and patients.
  - $v_{\ell j}$ are highly distinct between healthy controls and patients.
  - $u_{\ell j}$ and $v_{\ell j}$ are highly correlated.
- Attribute P-values to ith mRNA with assuming $u_{\ell i}$ obeys Gaussian distribution (the null hypothesis) using χ^2 distribution

$$P_i = P_{\chi^2}\left[> \left(\frac{u_{\ell i}}{\sigma_\ell}\right)^2 \right] \tag{3}$$

where $P_{\chi^2}[> x]$ is the cumulative χ^2 distribution where the argument is larger than x and σ_ℓ is the standard deviation. Also attribute P-values to kth miRNA with assuming $u_{\ell k}$ obeys Gaussian distribution (the null hypothesis) using χ^2 distribution

$$P_k = P_{\chi^2}\left[> \left(\frac{v_{\ell k}}{\sigma_\ell}\right)^2 \right] \tag{4}$$

- Correct P-values using BH criterion [12] with considering multiple comparison corrections.
- Select mRNAs and miRNAs associated with adjusted P-values less than 0.01.

Applying the above strategy, TD based unsupervised FE, to mRNA and miRNA expression of kidney cancer retrieved from TCGA data base [7] composed of 253 kidney tumors and 71 normal kidney tissues ($M = 324$), we selected 72 mRNAs and 11 miRNAs. The process was as follows. At first, we noticed that u_{2j} and v_{2j} satisfied these above requirements (Fig. 12); not only u_{2j} and v_{2j} are highly correlated (Pearson's correlation coefficient is 0.905, $P = 1.63 \times 10^{-121}$), but also they are highly distinct between tumors and normal tissues ($P = 7.10 \times 10^{-39}$ for u_{2j} and $P = 2.13 \times 10^{-71}$ for v_{2j}, computed by t test). Then u_{2i} and v_{2k} were used to attribute P_i to the ith mRNA using Eq. (3) and P_k to the kth miRNA using Eq. (4), respectively. Then 72 mRNAs and 11 miRNAs associated with adjusted P-values less than 0.01 were selected.

In order to see if the results obtained by this analysis is robust, i.e., independent of data set used, we applied the same procedure to yet another data set: mRNAs and miRNA expression of 17 kidney tumors and 17 normal kidney retrieved from GEO data set. Applying the same procedure to GEO data set, again we found that not only u_{2j} and v_{2j} are highly correlated (Pearson's correlation coefficient is 0.931, $P = 1.58 \times 10^{-15}$) but also they are distinct between kidney tumor and normal tissue ($P = 6.74 \times 10^{-22}$ for u_{2j} and $P = 2.54 \times 10^{-18}$ for v_{2j}, computed by t test) Since the outcomes when TD bases unsupervised FE was applied to two independent data sets are vary similar with each other, we can conclude that the success of our strategy is expected to be independent of used data set.

In order to confirm the superiority of TD based unsupervised FE toward other conventional supervised methods, we applied three conventional supervised methods, t test, SAM [13] and limma [10] to TCGA and GEO date sets. 13,895 mRNAs and 399 miRNAs for TCGA and 12,152 genes and 78 miRNAs for GEO were associated with adjusted P-values less than 0.01 by t test. On the other hand, when we applied SAM to TCGA and GEO data sets, 14,485 mRNAs and 441 miRNAs for TCGA and 16,336 mRNAs and 108 miRNAs for GEO were associated with adjusted P-values less than 0.01. Finally, limma attributed adjusted P-values less than 0.01 to 18,225 genes and 662 miRNAs for TCGA and 28,524 genes and 319 miRNAs for GEO. In contrast to the expectation, convectional statistical tests selected too many genes in

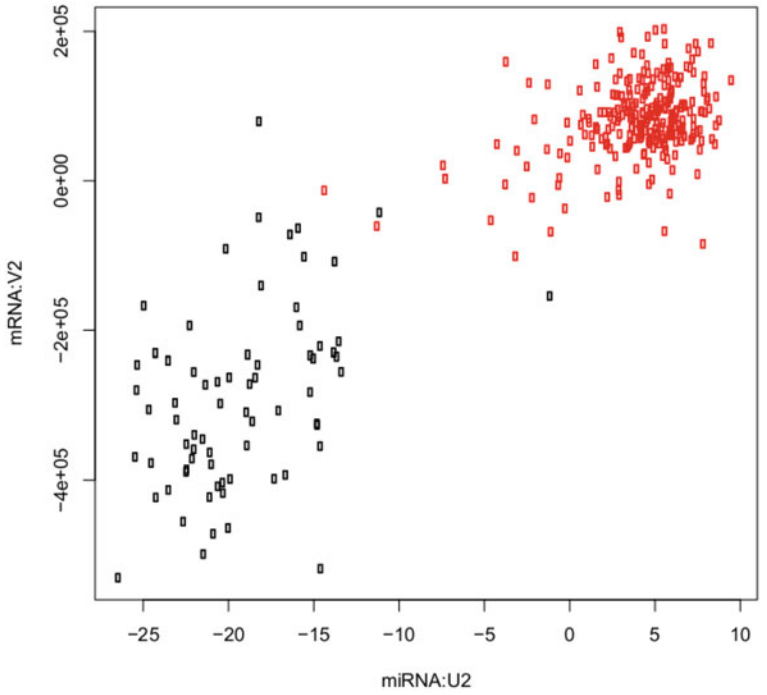

Fig. 12 Scatter plot between u_{2j} and v_{2j}. Black (red) open circle corresponds to normal (tumor) tissue

spite of that extreme *large p small n* problems. This can be understood as follows. Since tumor formation can affect wide range of mRNAs and miRNAs, there are huge number of mRNAs and miRNAs whose expression is distinct between tumor and normal tissues. Obviously, there are too many mRNAs and miRNAs if compared with total number of mRNAs ($\sim 10^2$) and that of miRNAs ($\sim 10^3$). Usually, in these cases, additional criterion, e.g., fold change between two classes, in this case tumor and normal tissue, is taken into consideration. For example, those associated with fold changes larger than 2 or less than 1/2. Empirically, this combinatorial strategy of fold change and significant P-values are known to work well. In spite of that, there are no reasoning why it works well. In contrast to these conventional statistical supervised methods that require additional criterion, e.g., fold change, in order to screen manageable limited number of genes, our strategy, TD based unsupervised FE, could identify reasonable limited number of mRNAs and miRNAs, without supports of any additional methods. In this sense, TD based unsupervised FE is a superior method to other conventional statistical supervised methods, e.g., t test, SAM and limma.

Acknowledgements The contents of this chapter were supported by KAKENHI, 23300357, 26120528, 20H04848, 20K12067, 19H05270 and 17K00417.

References

1. Batra, R., Malhotra, G.K., Singh, S., Are, C.: Managing squamous cell esophageal cancer. Surg. Clin. N. Am. **99**(3), 529–541 (2019). https://doi.org/10.1016/j.suc.2019.02.006
2. Johnstone, I.M., Titterington, D.M.: Statistical challenges of high-dimensional data. Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci. **367**(1906), 4237–4253 (2009). https://doi.org/10.1098/rsta.2009.0159
3. Kinoshita, R., Iwadate, M., Umeyama, H., Taguchi, Y.h.: Genes associated with genotype-specific DNA methylation in squamous cell carcinoma as candidate drug targets. BMC Syst. Biol. **8**(Suppl 1), S4 (2014). https://doi.org/10.1186/1752-0509-8-s1-s4
4. Kuleshov, M.V., Jones, M.R., Rouillard, A.D., Fernandez, N.F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S.L., Jagodnik, K.M., Lachmann, A., McDermott, M.G., Monteiro, C.D., Gundersen, G.W., Ma'ayan, A.: Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucl. Acids Res. **44**(W1), W90–W97 (2016). https://doi.org/10.1093/nar/gkw377
5. Montgomery, E., et al.: Oesophageal cancer. In: Stewart, B., Wild, C. (eds.) World Cancer Report 2014, Chap. 5.3, pp. 374–382. World Health Organization (2014)
6. Murakami, Y., Kubo, S., Tamori, A., Itami, S., Kawamura, E., Iwaisako, K., Ikeda, K., Kawada, N., Ochiya, T., Taguchi, Y.h.: Comprehensive analysis of transcriptome and metabolome analysis in intrahepatic cholangiocarcinoma and hepatocellular carcinoma. Sci. Rep. **5**(1) (2015). https://doi.org/10.1038/srep16294
7. Ng, K.L., Taguchi, Y.H.: Identification of miRNA signatures for kidney renal clear cell carcinoma using the tensor-decomposition method. Sci. Rep. **10**(1) (2020). https://doi.org/10.1038/s41598-020-71997-6
8. Oka, D., Yamashita, S., Tomioka, T., Nakanishi, Y., Kato, H., Kaminishi, M., Ushijima, T.: The presence of aberrant DNA methylation in noncancerous esophageal mucosae in association with smoking history. Cancer **115**(15), 3412–3426 (2009). https://doi.org/10.1002/cncr.24394. https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/cncr.24394
9. Raudvere, U., Kolberg, L., Kuzmin, I., Arak, T., Adler, P., Peterson, H., Vilo, J.: g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). Nucl. Acids Res. **47**(W1), W191–W198 (2019)
10. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., Smyth, G.K.: Limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucl. Acids Res. **43**(7), e47–e47 (2015). https://doi.org/10.1093/nar/gkv007
11. Student, S., Fujarewicz, K.: Stable feature selection and classification algorithms for multiclass microarray data. Biol. Direct **7**(1), 33 (2012). https://doi.org/10.1186/1745-6150-7-33
12. Taguchi, Y.h.: Unsupervised Feature Extraction Applied to Bioinformatics, A PCA Based and TD Based Approach. Springer International (2020). https://doi.org/10.1007/978-3-030-22456-1. https://app.dimensions.ai/details/publication/pub.1120509454
13. Tusher, V.G., Tibshirani, R., Chu, G.: Significance analysis of microarrays applied to the ionizing radiation response. Proc. Natl. Acad. Sci. **98**(9), 5116–5121 (2001). https://doi.org/10.1073/pnas.091062498

Machine Learning for Protein Engineering

Andrew D. Marques

Abstract This chapter is intended to outline methods and applications for machine learning (ML) in the context of protein engineering. The content of this chapter is geared toward the biologist's perspective of ML with an emphasis on experimental design for optimized data collection. After a brief introduction, the subsequent sections of this chapter will be dedicated to following a schema to design and implement ML for problems involving protein engineering. The steps below offer a guide to this schema and the organization of this chapter in the context of protein engineering.

I. Formulate a question that ML tools can answer
II. Design an experiment to collect data
III. Curate the dataset
IV. Choose and train a model
V. Interpret results

Keywords Protein engineering · Machine learning · Protein library · Linear regression · Random forest · Support vector machine · Artificial neural network · One-hot representation · Physicochemical representation

1 Applying Machine Learning in Protein Engineering

Protein engineering is the process of modifying proteins to improve the biological function or properties of the enzyme, receptors, or structural molecules. The first examples of engineering a protein of known function were performed in 1982 with TyrRs and Beta-lactamase. For TyrRs, this aminoacyl tRNA synthetase was engineered to determine that two amino acid residue positions in the protein contribute to more than 300,000-fold of the protein's catalytic activity [13]. By 2002, many of the current methods for protein engineering were developed, such as family shuffling and

A. D. Marques (✉)
Perelman School of Medicine, University of Pennsylvania, Philadelphia, USA
e-mail: amarques@pennmedicine.upenn.edu

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_2

surface scanning [4]. In the twenty years since 2002, one major change has occurred: a substantial decrease in the cost of sequencing. In 2002 the cost of sequencing was $3,898.64 per megabase, but in 2020 we have observed the rate of decrease in the cost of sequencing exceed Moore's law to the extent that one megabase may be sequenced at a cost of $0.008 [19]. Now that sequencing information is inexpensive, the question becomes, "What can we do to utilize this mass of new, low-cost data to its fullest potential?" ML offers an answer to this question. Big data, like the next generation sequencing (NGS) data previously described, has presented challenges in many related fields like pharmaceutical sciences, biomedical engineering, and biological sciences, all of which are handling the mass of data by using ML [14, 21]. Several forerunners in the protein engineering field are already paving the path for future work. Moreover, recent developments in ML-guided paradigms are helping to make ML protein engineering increasingly accessible with protein spaces containing millions of distinct sequences [3]. In particular, current resources like the *Handbook of Deep Learning Applications* have addressed challenges with computation time and complex reasoning in a manner that is understandable and practical [2]. In the coming years, large protein datasets will be expanded and the cost of computer processing will plummet, converging on a solution using ML.

1.1 Formulate a Question

In the first stage of applying ML for protein engineering, one must formulate a question that ML may answer. It is important to note that the algorithms are excellent at identifying complex patterns that nontraditional methods may fail to capture, however, not all questions are ideally suited for ML [22]. For example, if you are designing an algorithm to engineer ideal T-cell receptors (TCRs) given any number of antigens, you may run into issues with small sampling of this large space. Also, if the tools are not present to measure the features of the antigens, then the algorithm may not be able to capture the patterns associating sequences of TCRs with antigens. An important question that remains is determining how the library of peptides will be barcoded to assign them to the library of TCRs. Some advancements in peptide barcoding can help solve this, but it is likely that for the time being, the potential libraries' sizes will be limited to at most thousands of variants [5, 18]. If the patterns associated with the question are relatively simple, then asking a question where only thousands of data points are collected may be sufficient to answer the question, but for highly complex spaces, then the order of 10^5 or greater may have to be sampled for these patterns to be learned by the algorithms [27]. For the case of adeno-associated virus capsid libraries, it is possible to generate high complexity pools of variants and measure the space because methods exist to develop libraries where the genotype of the capsid is packaged within its respective phenotype of the virus-like particle. For this example, NGS can be utilized to measure the proteins present by sequencing the capsid genes rather than debarcoding barcoded peptides.

1.2 Design an Experiment

Once a question is formulated that ML tools may be used to answer, the subsequent step is to design and execute the experiments required to gather the data for training and validating the model. For this stage of the schema, it is important to consider that the model will be only as good as the protein data that it is used to train on. The experiment should be performed in a manner that tests for as few variables as possible resulting in fewer confounding factors. If multi-stepped questions must be asked, consider designing an experiment where data is collected in a stepwise fashion and the models are trained at each of these steps, followed by combining the stepwise model into an ML pipeline.

The complexity of the data is an important aspect to consider. One strength of ML is to offer solutions to complex problems, however, complex problems must be presented with as many of the variables as possible that may result in the outcome of the protein's function. For example, if a complex protein library presents mutations to positions outside the window that is captured by sequencing, those mutations that are not captured may result in confounding variables causing noise in the data and a reduction in the predictive capabilities of the algorithm. For this reason, the mutational space of the proteins should be limited to the availability of sequencing tools. Moreover, the designed experiment will likely involve a selective pressure, designed to answer the question at hand, followed by sampling of the protein space after the pressure. This provides the model with essential information to draw patterns for proteins that do or do not have the features required for answering the question. In many cases, it can be especially useful to design an experiment where sampling occurs both immediately before and immediately after the selection event which may then be used to normalize a result. For instance, protein variants with especially low presence before selection that show average presence after selection should be scored higher than protein variants with especially high presence before selection that also show average presence after selection.

1.3 Curate the Dataset

Curating the protein dataset is one of the most important steps after the data collection has completed. This step is often specific for the project at hand, but a few tips should be kept in mind for all cases. As a case in point, if the intended engineered space for potential protein variants is defined before selection, as it often is, then it may be prudent to remove NGS sequences gathered that do not represent the original intended variants. This is important for several reasons.

First, it may not be known if these unintended proteins are a result of sequencing error, or peptide synthesis. Sequencing errors are analogous to reading a book with poor fidelity resulting in poor reading comprehension, whereas peptide synthesis error, while maintaining true fidelity to the written text and therefore adequate

"reading comprehension," can result in the loss of intended meaning and pose a risk to the model's interpretation of the data. Sequencing-related errors can arise in the PCR steps for extracting the gene of interest before or after selection, in the PCR steps for barcoding, or there can simply be errors while sequencing. A predetermined quality score used as a cutoff from the NGS reads will likely help avoid problems associated with the latter cause [17]. Phred scores are the most common method for determining the quality of NGS reads and a Phred score threshold of 30 or greater will result in an inferred base call accuracy of >99.9% [6]. Equation 1 outlines how Phred scores are calculated where Q is the sequencing quality score, and e is the estimated probability of an incorrect base call.

$$Q = -10 * \log_{10}(e) \tag{1}$$

Second, even if the errors arose from peptide synthesis, because of their unintended nature they would likely be disproportionately underrepresented in the proportions of reads generated resulting in poor or misleading interpretation by the ML models. Consequently, it is advised to remove all reads that do not conform to the originally designed space of protein variants. In addition to removing sequences that do not match the intended protein space, curating techniques to minimize the memory required by the model can help to increase the size of the training/validation groups as well as speed up run times. Especially after removing reads that do not match the intended mutation space, it may then be useful to remove the amino acid positions which remain constant through all the variants, if there are any in the designed protein space.

1.4 Pick and Train a Model

Determining which ML approach is ideal to model a given dataset can be difficult because no one model will fit all data types optimally [24]. Consequently, the model employed must be examined on a question-by-question basis. For this reason, it may be useful to compare several models after data collection to determine which model performs best for fitting the given protein dataset. In the three previous steps where a question is posed, an experiment planned and executed, and data curated, the models intended to be used should be kept in mind. In the following paragraphs we will discuss the fundamental differences of several ML algorithms currently used in protein engineering and present different representations for the data to be framed when training the algorithms. Finally, this section will conclude with tips for choosing the ML models for your question.

1.4.1 ML Algorithms

Several ML algorithms are becoming especially popular in applications for protein engineering. In this section of the chapter, we will define and discuss the following 4 algorithms: linear regression, random forests, support vector machines (SVMs), and artificial neural networks (ANN).

Linear Regression

Linear regression is the simplest model that may be used for protein engineering. Conceptually, a simple formula is created to determine the best-fit line through the training examples given. Often for proteins, this is performed when each residue has undergone a linear transformation [7]. Alternatively, if peptides contain strings of residues that remain the same, blocks of the residue sequence may be linearly transformed [15]. Linear regression is recommended to be used as a baseline model before other more powerful models are used [27].

Random Forest

Random forests are ML models based on decision trees. Specifically, a random forest uses a set number of decision trees that are randomly generated, followed by an averaging of the results from each of the trees to produce a final classification. The easiest way to use random forests in the context of proteins is to have the branches be the position/residue combination at that position, and for the leaves to be the output in the form of classification. For example, if a protein dataset assessing the enzymatic function is used to train a random forest, one branch may be arginine in position 1, another branch may be lysine in position 1, and another branch may be tyrosine in position 2. Another decision tree may select several different position/residue combinations. If the majority of the decision trees provide a consensus that the enzyme has increased enzymatic function, then that would be the output of the random forest classification. Enzyme thermostability is one example of a protein feature that has used random forests for protein engineering [11, 16, 25].

SVM

An SVM identifies similarities in the training data to implicitly project the features, creating a map of the features in a way that may predict unknown inputs. Critical to this approach are the different kernels that are responsible for making projections from the input features to a high-dimensional space for this mapping to occur. A couple examples of previous publications using SVMs for protein engineering include enantioselectivity [28], protein localization [23], and protein folding [26].

ANN

An ANN can be broken into three components: (1) the input layer that would represent the encoded protein, (2) the hidden layer(s) that can have stacks of layers connected by linear or non-linear activation functions, and (3) the output layer that communicates the end-point classification for the proteins. Non-linear activation functions have been the backbone to many modern ANN models. Several examples of ANNs used

in protein engineering have been used to predict protein–protein interactions [10] and three-dimensional structures [9].

1.4.2 ML Representation

Representing protein data to the model is critical to how the model learns. Conceptually, this can be thought of as the "language" that will be spoken to the model. For example, in a mathematics classroom, a teacher would not likely teach a lesson by using roman numerals nor spelling out each number, rather, Hindu-Arabic numerals are the preferred representation because each base ten digit received a dedicated character to read, simplifying addition and making long division understandable. Conversely, teaching a literature history course would be exceedingly difficult if Morse code was used over the students' native language.

Just as some classrooms require specific languages to effectively be taught, ML using protein-derived data may best train the model in one of several different representations. In few instances would a categorical approach to representing a protein be ideal. One of these instances would be a random forest model where categorical data is easily presented. In this case, "A" for Alanine, "C" for cysteine, "D" for aspartic acid, etc. is commonly used.

One-hot Representation

However, many models are not optimized to work with categorical data. For these models, a one-hot representation is the next simplest option. In the one-hot representation, each possible residue for each position receives a binary variable where a "1" indicates that a particular residue was present at the given position in the protein sequence. For example, most proteins are composed of 1 of 20 common amino acid residues, so a peptide containing n number of alanines may be represented as a "1" followed by nineteen "0" repeated n number of times to represent the entire peptide. See Fig. 1 for another example.

Ordinal Representation

Notably, using a numerical representation using the values 1–20 where 1 indicates alanine, 2 indicates cysteine, 3 indicates aspartic acid etc. results in ordinal values being prescribed to the residues that can cause problems as the model is being trained. This approach is not recommended. If this representation is used, it is advisable to encode the residues in an ordinal manner that conveys biological information. For example, residues may be ordered based on their physical size where 1 may indicate the largest residue (tryptophan), 2 indicates the second largest residue (tyrosine), 3 indicates the third largest residue (phenylalanine) and so on. This type of protein representation begins to take on the role of representing the physical properties of each residue, which will be discussed next.

Fig. 1 One-hot representation of peptide sequences

Physicochemical Representation

Using physical properties is one method of representing protein data with increased information. With more information about the residues, the model is more likely to have a greater accuracy learning the datasets. Marques et al. [17] describe a representation using physical and chemical properties of each residue termed the physicochemical classification, as depicted in Table 1. This data is derived from data gathered by Pommie et al. [20] that includes volume, hydropathy, polarity, charge, hydroxyl presence and several other properties. There are a variety of alternative approaches using ProFET or AAIndex to represent protein data in this manner.

Table 1 Amino acid physicochemical classification

Amino Acid Physicochemical Classification												
Residue	Volume	Hydropathy	Polarity	Hydrogen Donor	Hydrogen Acceptor	Positive Charge	Negative Charge	Aliphatic	Aromatic	Sulfur	Hydroxyl	Amide
A	0.17	0.70	0	0	0	0	0	1	0	0	0	0
C	0.29	0.78	0	0	0	0	0	0	0	1	0	0
D	0.30	0.11	1	0	1	0	1	0	0	0	0	0
E	0.47	0.11	1	0	1	0	1	0	0	0	0	0
F	0.77	0.81	0	0	0	0	0	0	1	0	0	0
G	0.00	0.46	0	0	0	0	0	1	0	0	0	0
H	0.56	0.14	1	1	1	1	0	0	0	0	0	0
I	0.64	1.00	0	0	0	0	0	1	0	0	0	0
K	0.65	0.07	1	1	0	1	0	0	0	0	0	0
L	0.64	0.92	0	0	0	0	0	1	0	0	0	0
M	0.61	0.71	0	0	0	0	0	0	0	1	0	0
N	0.32	0.11	1	1	1	0	0	0	0	0	0	1
P	0.31	0.32	0	0	0	0	0	1	0	0	0	0
Q	0.50	0.11	1	1	1	0	0	0	0	0	0	1
R	0.68	0.00	1	1	0	1	0	0	0	0	0	0
S	0.17	0.41	1	1	1	0	0	0	0	0	1	0
T	0.33	0.42	1	1	1	0	0	0	0	0	1	0
V	0.48	0.97	0	0	0	0	0	1	0	0	0	0
W	1.00	0.40	0	1	0	0	0	0	1	0	0	0
Y	0.80	0.36	1	1	1	0	0	0	1	0	0	0

Table 2 ML models summary

ML models summary			
Model	Potential accuracy for complex datasets	Categorical input?	Training size
Linear regression	Low	Yes	>10 samples per variable
Random forest	High	Yes	<1E4 samples
SVM	High	No[a]	<1E4 samples
ANN	High	No[a]	>50 samples per weight

[a] Categorical data may not be used unless it is numerically encoded

If complex relations with the data are not expected, then a simple linear regression model may be employed. Relatively small sample sizes may need to be collected to utilize linear regression with a recommended minimum of 10 samples per variable [8]. In the context of protein engineering, this might translate to having at least 50 training samples if 5 amino acid residues are mutated. Linear regression models are not likely to capture complex interactions in protein engineering schemas, therefore, a random forest is the next simplest option. One prominent benefit of constructing random forests is the readability of the model, however, high accuracy models capturing complex relationships can quickly become exceedingly large and unwieldy. Experiments with tens of thousands of mutational variants may be best designed with this model in mind. Moreover, the ability for this model to retain categorical inputs further aids in the interpretability of this model's functioning. Additionally, for similarly sized complexity in protein engineering schemas, SVMs may be used. However, because categorical data cannot be represented by most SVM kernels, there is more opacity with the internal workings of this model. One-hot representations or representations based on the physical properties of the residues are two of the best options for this approach. Finally, for most of the cases that an SVM may be employed, an ANN may alternatively be applied with more than 50 samples collected per weight [1]. Furthermore, Marques et al. [17] provide the python scripts to convert NGS reads to one-hot and physicochemical representation, as well as templates for SVM and ANN modelling.

As we have seen above, understanding your dataset can help direct you to models that may work best, but thinking about the models that will be used at the start of the experiment can be critical. Table 2 outlines the different ML models paired with the types of input data that they perform best with.

1.5 Interpret the Results

Correctly interpreting the output of the tuned models to your protein engineering questions is perhaps just as critical as choosing and training the model itself. At this point, the cumulative nature of the previous four steps determines the consequential

ease of interpreting the output results. Either careful deliberation for experimental designs and model training can yield straightforward results, or hasty decisions can cause difficulties and often convoluted interpretations.

Interpretability is the ability to understand and conceptualize the meaning of something. In the context of ML, this is understanding the significance of the output and to what extent the findings of the readout may extend. With highly complex protein sets, it is especially important to not extrapolate beyond what the data represents. If for example, a specific backbone is used in the protein and mutations are made to specific variable regions of the backbone, then the algorithm's ability to make outcome predictions based on the mutations of the variable regions would be limited in implications to the backbone with few exceptions. In most cases there is a tradeoff between higher interpretability and generalizability. For models that function in more simple manners, such as simple linear regression or decision trees, then the ability for the user to understand the working of the model comes at a cost to the model's ability to capture complex relations. Moreover, as many nonlinear models can capture more complex data, they are now able to have increased generalizability for output within the context of the input data, but at the cost of the user's ability to understand how the model learns, as with random forests and ANNs.

Causation and correlation of the model's predictive capabilities is yet another important factor to consider. Fortunately for many protein engineering schemas, if you are measuring the mutations to residues and their impacts on protein structure or function, then it is highly likely that outcomes of the results are causal, so much as the data set was generated and the samples used to train the dataset are taken before and after the selective process without substantial interference from other unintended selective pressures. Existing databases that only show protein functions after selective processes may have important data that models can be trained on, but the full implications should be approached with caution. To illustrate this, a database indicating the enzymatic function of a protein given a specific sequence may be used to train algorithms, but without full knowledge of the pool of variants before and after the measurement of enzymatic function, the data set may also be representing sequences that are optimal at a given pH or temperature and the pool of variants may be different if the conditions are altered. In this example, the dataset might be measuring multiple selective pressures beyond the intended functions.

Additionally, bottlenecking can skew results especially when using large datasets. This effect in ML protein engineering is analogous to evolutionary genetics in the event of mass extinction where some phenotypes may not be advantageous in itself but may be presented in post-bottleneck populations in disproportionate representation solely due to chance in the surviving population. These bottleneck events have the potential to introduce bias into the model and can severely alter the implications of model interpretation [12]. Similarly, results from protein engineered ML models that experience any number of bottleneck effects that may pose a non-specific selection should yield results that are only cautiously used until the model's predictive power can be further verified. Biological replicates are one useful tool to substantiate a model's predictive power. To supplant this, increasing the sample sizes gathered to be used for training and testing the model may further validate the model's findings.

Finally, experimental replicates are one of the best approaches to check if the designed model fits the data in a biologically accurate manner.

Ultimately, envisioning the end product before beginning the experiment and collecting data will simplify this interpretation stage of implementing ML in protein engineering. The model will only be as good as the data that it is trained on, therefore, clean data with easily represented results will be most likely to yield simpler and more generalizable model predictions.

2 Conclusion

This chapter outlines some of the fundamental principles required to apply ML for protein engineering questions. In doing so, it includes references and tools for further reading to help direct the reader to more in-depth reviews that may be relevant to your specific research areas and topics. The applications of this field in protein engineering are still in its infancy, but with an ever-increasing number of researchers utilizing ML to elevate their research, the dawn of ML-derived proteins has already begun.

References

1. Alwosheel, A., van Cranenburgh, S., Chorus, C.G.: Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. J. Choice Model. **28**, 167–182 (2018). https://doi.org/10.1016/j.jocm.2018.07.002
2. Balas, V.E., Roy, S.S., Sharma, D., Samui, P. (eds.): Handbook of Deep Learning Applications, vol. 136. Springer, New York (2019)
3. Biswas, S., Khimulya, G., Alley, E.C., Esvelt, K.M., Church, G.M.: Low-N protein engineering with data-efficient deep learning. Nat. Methods **18**(4), 389–396 (2021)
4. Brannigan, J.A., Wilkinson, A.J.: Protein engineering 20 years on. Nat. Rev. Mol. Cell Biol. **3**, 964–970 (2002). https://doi.org/10.1038/nrm975
5. Egloff, P., Zimmermann, I., Arnold, F.M., et al.: Engineered peptide barcodes for in-depth analyses of binding protein ensembles (2018). https://doi.org/10.1101/287813
6. Ewing, B., Green, P.: Base-calling of automated sequencer traces using Phred II. Error probabilities. Genome Res. **8**, 186–194 (1998). https://doi.org/10.1101/gr.8.3.186
7. Fox, R.J., Davis, S.C., Mundorff, E.C., et al.: Improving catalytic function by ProSAR-driven enzyme evolution. Nat. Biotechnol. **25**, 338–344 (2007). https://doi.org/10.1038/nbt1286
8. Harrell, F.: Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal ... Regression, and Survival Analysis. Springer (2016)
9. Hopf, T.A., Colwell, L.J., Sheridan, R., et al.: Three-dimensional structures of membrane proteins from genomic sequencing. Cell **149**, 1607–1621 (2012). https://doi.org/10.1016/j.cell.2012.04.012
10. Hu, J., Liu, Z.: DeepMHC: Deep Convolutional Neural Networks for High-performance peptide-MHC Binding Affinity Prediction (2017). https://doi.org/10.1101/239236
11. Jia, L., Yarlagadda, R., Reed, C.C.: Structure based thermostability prediction models for protein single point mutations with machine learning tools. PLoS ONE (2015). https://doi.org/10.1371/journal.pone.0138022

12. Kadoya, S., Urayama, S., Nunoura, T., et al.: Bottleneck Size-Dependent Changes in the Genetic Diversity and Specific Growth Rate of a Rotavirus a Strain (2019). https://doi.org/10.1101/702233
13. Leatherbarrow, R.J., Fersht, A.R., Winter, G.: Transition-state stabilization in the mechanism of tyrosyl-tRNA synthetase revealed by protein engineering. Proc. Natl. Acad. Sci. **82**, 7840–7844 (1985). https://doi.org/10.1073/pnas.82.23.7840
14. Lee, K.C., Roy, S.S., Samui, P. (eds.): Data Analytics in Biomedical Engineering and Healthcare. Academic Press (2020)
15. Li, Y., Drummond, D.A., Sawayama, A.M., et al.: A diverse family of thermostable cytochrome P450s created by recombination of stabilizing fragments. Nat. Biotechnol. **25**, 1051–1056 (2007). https://doi.org/10.1038/nbt1333
16. Li, Y., Fang, J.: PROTS-RF: a robust model for predicting mutation-induced protein stability changes. PLoS ONE (2012). https://doi.org/10.1371/journal.pone.0047247
17. Marques, A.D., Kummer, M., Kondratov, O., et al.: Applying machine learning to predict viral assembly for adeno-associated virus capsid libraries. Molecular Ther. Methods Clin. Dev. **20**, 276–286 (2021). https://doi.org/10.1016/j.omtm.2020.11.017
18. Miyamoto, K., Aoki, W., Ohtani, Y., et al.: Peptide barcoding for establishment of new types of genotype–phenotype linkages. PLoS ONE (2019). https://doi.org/10.1371/journal.pone.0215993
19. NIH: DNA sequencing costs: data. In: Genome.gov (2020). https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data. Accessed 24 Feb 2021
20. Pommié, C., Levadoux, S., Sabatier, R., et al.: IMGT standardized criteria for statistical analysis of immunoglobulin V-REGION amino acid properties. J. Mol. Recognit. **17**, 17–32 (2004). https://doi.org/10.1002/jmr.647
21. Roy, S.S., Samui, P., Deo, R., Ntalampiras, S. (eds.): Big Data in Engineering Applications, vol. 44. Springer (2018)
22. Roy, S.S., Taguchi, Y.H.: Identification of genes associated with altered gene expression and m6A profiles during hypoxia using tensor decomposition based unsupervised feature extraction. Sci. Rep. **11**(1), 1–18 (2021)
23. Saladi, S.M., Javed, N., Müller, A., Clemons, W.M.: A statistical model for improved membrane protein expression using sequence-derived features. J. Biol. Chem. **293**, 4913–4927 (2018). https://doi.org/10.1074/jbc.ra117.001052
24. Samui, P., Roy, S.S., Balas, V.E. (eds.): Handbook of Neural Computation. Academic Press (2017)
25. Tian, J., Wu, N., Chu, X., Fan, Y.: Predicting changes in protein thermostability brought about by single- or multi-site mutations. BMC Bioinf. **11**, 370 (2010). https://doi.org/10.1186/1471-2105-11-370
26. Yan, K., Wen, J., Liu, J.X., Xu, Y., Liu, B.: Protein fold recognition by combining support vector machines and pairwise sequence similarity scores. IEEE/ACM Trans. Comput. Biol. Bioinf. (2020)
27. Yang, K.K., Wu, Z., Arnold, F.H.: Machine-learning-guided directed evolution for protein engineering. Nat. Methods **16**, 687–694 (2019). https://doi.org/10.1038/s41592-019-0496-6
28. Zaugg, J., Gumulya, Y., Malde, A.K., Bodén, M.: Learning epistatic interactions from sequence-activity data to predict enantioselectivity. J. Comput. Aided Mol. Des. **31**, 1085–1096 (2017). https://doi.org/10.1007/s10822-017-0090-x

Statistical Relational Learning for Genomics Applications: A State-of-the-Art Review

Marenglen Biba and Narasimha Rao Vajjhala

Abstract This paper aims to review the state-of-the-art statistical relational learning models (SRL) in genomics. SRL deals with machine learning and data mining in relational domains where observations may be missing, partially observed, and noisy. This chapter introduces a background overview of various models, including probabilistic graphical models, Bayesian networks, dependency networks, Markov networks, first-order logic, and probabilistic inductive logic programming. This chapter also discusses the various statistical relational learning approaches, including probabilistic relational models, stochastic logic programs, Bayesian logic programs, relational dependency networks, relational Markov networks, and Markov logic networks. Finally, the last part of the paper focuses on the practical application of statistical relational learning techniques in genomics. The chapter concludes with a discussion on the limitations of current methods.

Keywords Genomic · Artificial intelligence · Machine learning · Probabilistic · Bayesian · Markov · Dependency · Genetics · Bioinformatics

1 Introduction

Genomics is a convergence of many sciences, including genetics, molecular biology, biochemistry, statistics, and computer science [1]. Genomics has affected all the areas of health sciences and the completion of the Human Genome Project (HGP) in 2003 initiated significant genomic studies [2]. Artificial Intelligence (AI) has considerable potential in advancements in genomics, biomedical data analysis, and drug discovery [3]. Significant advances in genomics, including Next-Generation Sequencing (NGS) technologies, computational analysis, deep learning, and machine

M. Biba · N. R. Vajjhala (✉)
University of New York Tirana, Tirana, Albania
e-mail: narasimharao@unyt.edu.al

M. Biba
e-mail: marenglenbiba@unyt.edu.al

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_3

learning approaches, have contributed to several advancements in the areas of medicine and agriculture [4–6]. Modern supercomputers, coupled with machine learning algorithms and systems, have explored genetic data and helped rapid advancements in precision medicine [3]. The human genome projects have reported large amounts of genetic information, and the application of machine language algorithms was essential to process and analyze these large volumes of genetic data.

Statistical Relational Learning (SRL) is considered as an area of machine learning combining the features of statistical and probabilistic modeling with languages supporting structured data representations [7, 8]. In SRL, the data is represented as a graph consisting of nodes or entities and labeled edges indicating the entities' relationship [9]. SRL involves learning from relational datasets as well as solving tasks, including entity resolution, link-based clustering, and link prediction or knowledge graph completion [9–11]. SRL integrates logic-based learning approaches with probabilistic graphical models [8]. SRL involves the creation of statistical models for relational data [9]. SRL incorporates structure through logic, databases, and programming languages and also builds on ideas of statistics and probability theory to address uncertainty [7]. Statistical learning addresses the issue of integration of probabilistic reasoning with first-order logic representations and machine learning [12]. SRL provides the information needed for automatically applying powerful statistical techniques from social sciences, for instance, quasi-experimental designs [13]. There are three categories of SRL models, namely, relational graphical models, latent class models, and tensor factorization models [10]. Relational graphical models are constructed mainly through the Bayesian or Markov networks, including Markov logic networks and probabilistic relational models [10]. The approach in probabilistic relational models is a combination of Bayesian networks with an object-oriented language for describing the structure of the domain [7]. Some of the examples of latent class models include Stochastic Block Model (SBM) and the Infinite Relational Model (IRM) [7].

In the context of precision medicine, AI is broadly classified into three categories, namely, Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI), and Artificial Super Intelligence (ASI). The field of ANI is currently in the development stage and is likely to be used and applied over the next decade. The developments in the area of ANI are likely to allow deep learning algorithms to analyze the data sets, identify new correlations, draw conclusions, and support physicians [3]. Third-generation sequencing started with the Nanopore and Pacific Bioscience technologies that allowed a cheaper and faster method for genome sequencing. These technologies longer sequence reads but led to a higher number of sequencing errors than Illumina sequencing. Several applications and packages, including Clairvoyante and DeepVariant based on Artificial Neural Networks (ANN), became prominent in genomics studies [3].

Machine learning algorithms mainly focused on propositional data that mainly recorded homogeneous and statistically independent objects [11–14]. In contrast, relational data recorded the characteristics of heterogeneous objects and the relationship between these objects. The difference between machine learning and SRL approaches is that while machine learning methods focus on learning probabilistic

models from specific data and efficient learning and inference, the SRL methods focus on storing and efficiently querying uncertain data [15]. Genomic structures are typical examples of relational data [14]. Genomics data is complex and cannot be investigated by pairwise correlations [16]. Genomics data is largely unstructured and can be considered to be in the realm of big data because of its volume, velocity, and variety [17]. A collection of random variables is considered independent and identically distributed (i.i.d) if the probability distribution of each of the random variables is similar, and all the random variables are mutually independent. There is a literature gap as the most recent work in learning has focused on propositional data [18]. A number of the existing structure learning algorithms assume that data consists of a single i.i.d. sample [13, 18, 19]. This independence assumption is not valid in real-world application domains where uncertainty, non-deterministic relations, and complex relational structure characterize the data [11]. Several existing i.i.d. models, including k-means and k-nearest neighbors algorithms, cannot handle non-i.i.d. data [20]. Most of the big data is non-i.i.d., while most of the existing analytical methods are i.i.d [20]. Non-i.i.d. data refers to the mixture of couplings, including neighborhood, dependence, linkage, correlation, and causality [20]. Data is not identically and independently distributed in several domains, including genomics, metabolism, bioinformatics, fraud, terrorism analysis, and robotics [12, 13, 18, 19].

Machine learning methods' performance depends heavily on data representation, representing a critical difference between classic machine learning and deep learning. Feature learning is an essential aspect of DL. Conventional machine learning approaches use manual feature engineering to highlight the weaknesses of the current learning algorithms. In contrast, features are learned automatically with multiple levels of representation in deep learning. These semantic labels are assumed to be mutually exclusive and the feature learning methods do not capture the complexity of these semantic labels [18–20]. However, in real-world applications there several thousands of semantic categories because of which representing the semantic relations between these categories is quite complex. Deep learning makes it possible for quicker extraction of useful information for complex tasks and often provides better results than other classical machine learning approaches.

2 Background and Notation

In this section, the background theory and notation needed to understand the statistical relational learning models and techniques are included.

2.1 Probabilistic Graphical Models

The five current research areas in probabilistic machine learning include probabilistic programming, Bayesian optimization, probabilistic data compression, automated

discovery of interpretable models from data, and hierarchical modeling [21–25]. PGMs use diagrammatic representations to describe the random variables and the relationships among these variables [26]. The nodes of a PGM represent the random variables, while the links represent the nodes' probabilistic relationships [26]. PGMs provide a distinct combination of several features of the probability theory and graph theory. PGMs provide a reliable platform for dealing with uncertainty, independence, and complexity.

The main challenge for probabilistic machine learning is the model's flexibility in capturing most of the properties of the data required to complete the prediction [25]. There are two types of PGMs, namely, directed PGM and undirected PGM. Directed PGM is also referred to as Bayesian networks, while undirected PGM is also referred to as Markov random fields [26]. Bayesian networks are directed graphical models with the arrows between the links of the graphs indicate the directionality. Markov random fields are undirected graphical models with no arrows between the links and no directional significance [27]. Latent Dirichlet Allocation (LDA) is an example of directed PGM. LDA is often used as a topic model to analyze the generation of words and topics in documents [26]. A detailed treatment of directed and undirected PGM can be found in Koller and Friedman [27], Larrañaga [28], and Pernkopf et al. [29].

2.2 Bayesian Networks

Machine learning methods improve their performance at specific tasks based on observed data [25]. A machine learning model is considered as well-defined if the model can forecast unobserved data using the training on observed data. Probabilistic models use the probability theory as the framework to express all forms of uncertainty [25]. Bayesian learning refers to learning from data through the application of the probability theory [25]. Bayesian networks are a formal graphical language representing the joint probability distribution over a set of random variables [30]. Bayesian models and classification-based machine learning methods are used extensively for genomic prediction [31]. Bayesian networks are ideal for assessing cascading disruptive events offering several advantages, including combining multiple information sources, structural learning possibility, and explicit treatment of uncertainty [32]. A sample graph structure of a BN is shown in Fig. 1.

Bayesian networks have three types of nodes: root notes, leaf nodes, and intermediary nodes. Root notes do not have a parent node, lead nodes do not have child nodes, and intermediary nodes include both parent and child nodes [32]. There are two types of learning in Bayesian networks—structure learning and parameter learning. Structure learning estimates the network links, while parameter learning estimates the conditional probabilities in the network [32]. The complete joint probability distribution of BN consisting of n variables A_1, A_2, …., A_n is shown in Eq. 2.1.

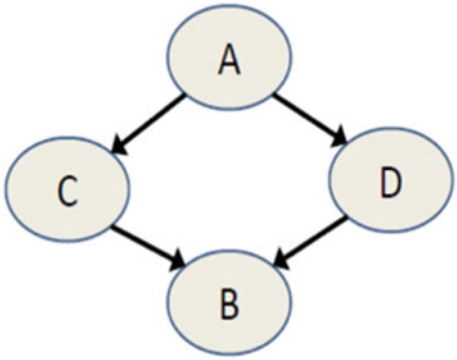

Fig. 1 Graph structure of a sample Bayesian network. Adopted from [33]

$$\mathrm{P}(A_1, A_2, \ldots, A_n) = \prod_{i=1}^{n} P(A_i | Parents(A_i)) \tag{2.1}$$

3 Statistical Learning Relational Models

SRL models deal with the combination of relational logic with probabilistic and statistical approaches to inference and learning [33–38]. In classical statistical inference problem formulations, the instances are homogeneous, independent, and identically distributed (i.i.d) [39–42]. Hence, these instances are represented as a single flat table while using traditional machine language approaches. However, in real-world scenarios, including in genomics, the data is represented by interrelated entities requiring relational approaches [43, 44]. Some of the SRL approaches used include Probabilistic Relational Models (PRMs), Stochastic Logic Programs (SLPs), Bayesian Logic Programs (BLPs), Relational Dependency Networks (RDN), Relational Markov Networks (RMNs), and Markov Logic Networks (MLNs) [44].

3.1 Probabilistic Relational Models

Probabilistic Relational Models (PRMs) model the intersection between relational and uncertainty representation through the combined use of Bayesian Networks and relational databases [44]. Relational algebra, which is the processing basis for SQL, lacks a relevance-based ranking of retrieved objects in the context of uncertainty management. Hence, many probability extensions for relational algebra have been proposed, including probabilistic relational modeling [45]. Bayesian Networks are not suitable for relational data, so PRMs perform flattening to reduce the dataset to a propositional form by fixing the feature space and mapping multiple feature values

into a single value [44]. PRMs are quite useful while modeling the intrinsic uncertainty of knowledge [45]. PRMs extended the classical graphical models, including BNs, to relational domains and were successfully evaluated in genomic data [14]. PRMs remove the assumption of i.i.d. instances that is core to conventional machine learning techniques [14].

3.2 Relational Dependency Network

Autocorrelation is a key characteristic of relational datasets indicating the statistical dependency between the same variable's values on related entities. For instance, there is autocorrelation in the functions of co-located proteins in a cell. Relational Dependency Networks (RDN) are extensions of DNs for relational data representing the cyclic dependencies required to exploit autocorrelation during collective inference [14]. RDNs offer simple methods for structure learning and parameter estimation resulting in models that are easy to interpret [14]. RDNs are an approximate model as compared to other PRMs. The approximation quality of RDNs is sufficient if the models learn from large datasets and are combined with inference techniques, for instance, the Monte Carlo technique [14].

3.3 Relational Markov Networks

Markov models are ideal for models that represent sequential processes. Markov models were evaluated and applied in several areas, including computational biology. RMMs are a generalization of Markov models allowing states of different types, with each type consisting of different variables [46–49]. Relational Markov Networks (RMNs) extends the framework of Markov networks to relational domains [50]. RMNs are a generalization of conditional random fields allowing the collective classification of a set of related entities by integrating information from the entities and the relationship between the entities [51]. A relational Markov network specifies a conditional distribution over all of the entities' labels in an instantiation given the relational structure and the content attributes. RMNs are undirected PRMs that can represent arbitrary forms of autocorrelation [14]. Undirected PRMs address two limitations of the directed models. As the Markov nets are undirected models, so the cycles are no longer a problem. Also, undirected models are well suited for discriminative training. In undirected models, we optimize the labels' conditional likelihood, given the features generally improve classification accuracy. However, even though RMN techniques can learn cyclic autocorrelation dependencies, it is quite impractical to implement because of inefficient parameter estimation [14]. RMNs are useful in machine learning in domains, such as genomics, where the state space is large and heterogeneous with only sparse data [49].

$$P(I.y|I.x, I.r) = \frac{1}{Z(I.x, I.r)} \prod_{c \in \boldsymbol{C}} \prod_{c \in \boldsymbol{C}(I)} \varphi_c(I.x_c, I.y_c) \tag{3.1}$$

where $Z(I.x, I.r)$ is the normalizing partition function as shown in Eq. 3.2:

$$Z(I.x, I.r) = \sum_{I.y'} \prod_{c \in \boldsymbol{C}} \prod_{c \in \boldsymbol{C}(I)} \varphi_c(I.x_c, I.y'_c) \tag{3.2}$$

4 SRL in Genomics—Problems and Applications

4.1 SRL Problems in Genomics

Genomics is an established field now and is expected to generate the largest amounts of data by 2025 [51–54]. The data gathering is taking a faster pace as compared to data processing and analytics. In particular, machine language algorithms are suited for genomics, as these algorithms are designed to detect patterns in data automatically. However, the machine language algorithms' performance is limited by how the data is represented and each variable is computed. Eraslan et al. [16] give the example of the classification of a tumor as malign or benign from a microscopy image. In this case, the algorithm's classification accuracy depends on the visual features, such as cell morphology and the distances between the cells. However, the accuracy of these machine learning models can be improved through deep neural networks that would allow the discovery of high complexity features, such as cell morphology, by taking the operations of previous operations as input.

The problems in SRL in genomics mainly involve the issues related to probabilistic approaches to machine learning, including data compression, optimization, decision making, scientific model discovery, and interpretation [25]. While probabilistic machine learning provides clear indications and directions on solving a problem, the key challenge lies in finding a computationally efficient solution. One of the problems with SRL methods is that several models only focus on predicting relations and ignoring the properties [55]. Some of the SRL methods only consider a subset of the relational data, for instance, only set data or graph data [55]. One of the critical limitations of SRL methods relates to the computational complexity of inference. For instance, PRMs, MLNs, and BLNs suffer from inference as they use standard complex BN inference algorithms on the graphs or use an undirected model as their ground network [56]. Another factor causing significant representation and computational difficulties for most of the SRM models is autocorrelation. For instance, PRMs and BLNs have autocorrelation shown in the model as they add a single random variable to their class model for every descriptive attribute in the dataset [56]. The computational complexity of inference is probably the most significant limitation shared between most SRL methods, followed by the size of the graph

that is proportional to the number of descriptive attributes and objects, limiting the scalability for many realistic datasets.

4.2 SRL Applications in Genomics

Machine learning methods, and in particular SRL methods have been applied to a wide range of applications within genomics and genetics. For instance, SRL methods have been used in the interpretation of large genomic datasets apart from the annotation of wide range of genomic sequence elements [30]. SRL methods also help handle missing data values, such as defective cells in a gene expression microarray. The missing data values could be values missing at random or values that, when absent, provide information about the task at hand [30]. SRL methods can explicitly model missing data by considering all potential missing values by providing an annotation assigning labels to each position across the genome and modeling the probability of observing a specific value given this label. SRL methods have been used to help breeders and researchers identify stress resistance genes. ILP was used to identify several drought resistant genes that correspond to the production of certain proteins in some of the plant species [57].

Researchers have also identified plant disease resistance genes using SRL methods. This research is likely to help improve crop production. SRL methods can also be used in Genome Wide Association Studies (GWAS) to infer genotype–phenotype associations and identify the relations between the genetic characteristics and response to specific treatments [58–62]. The Million Genomes project was launched in Europe in 2018 to collect and make available large amounts of individual genome information [63, 64]. SRL methods will play an important role in genome analysis for these large genomic data sets. SRL methods can also help in mining Electronic Health Records (EHR) to provide useful information that can be used in drug discovery and safety [65, 66]. SRL techniques can help mine structured and unstructured data in EHRs. Researchers also use SRL methods to analyze complex genomic data by applying genomic data to SRL algorithms and predicting new unlabeled data. SRL methods can help improve the existing methods and identify new methods for developing genomic and cancer data visualization tools.

5 Conclusion

SRL methods aim at learning in noisy domains described in terms of objects and relationships by combining probability with first-order logic. Machine learning methods and, in particular, SRL methods are applied to a wide range of genomics and genetics applications. New technologies for generating large genomic and proteomic data sets are emerging, with genomic data generated faster than the existing methods for analyzing these data. Annotating all the sequences in genomic and proteomic data

can be unfeasible and costly, and the use of SRL methods is essential to automatically annotate these sequences. The demand for machine learning methods capable of applying and adapting to these big data sets will increase over the next decade. SRL methods will be essential to handling these large genomic data sets and advancing genetics and genomics. We hope that the issues presented in this chapter will advance the statistical relational learning community's discussion about the next generation of statistical relational learning techniques for genomics.

References

1. Qu, Z., et al.: Using visualization to illustrate machine learning models for genomic data. In: Proceedings of the Australasian Computer Science Week Multiconference, p. Article 15. Association for Computing Machinery, Sydney, NSW, Australia (2019)
2. Khorraminezhad, L., et al.: Statistical and machine-learning analyses in nutritional genomics studies. Nutrients **12**(1), 1–20 (2020)
3. Nagarajan, N., et al.: Application of computational biology and artificial intelligence technologies in cancer precision drug discovery. Biomed. Res. Int. **2019**, 8427042 (2019)
4. Dias, R., Torkamani, A.: Artificial intelligence in clinical and genomic diagnostics. Genome Med. **11**(1), 70 (2019)
5. Esposito, S., et al.: Applications and trends of machine learning in genomics and phenomics for next-generation breeding. Plants **9**(1), 1–18 (2020)
6. Eapen, B.: Artificial intelligence in dermatology: a practical introduction to a paradigm shift. Indian Dermatol. Online J. **11**(6), 881–889 (2020)
7. Getoor, L., Mihalkova, L.: Learning statistical models from relational data. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 1195–1198 . Association for Computing Machinery, Athens (2011)
8. Biba, M., et al.: A novel structure refining algorithm for statistical-logical models. In: 2010 International Conference on Complex, Intelligent and Software Intensive Systems (2010)
9. Nickel, M., et al.: A review of relational machine learning for knowledge graphs. Proc. IEEE **104**(1), 11–33 (2016)
10. Shi, C., Lu, W., Song, R.: Determining the number of latent factors in statistical multi-relational learning. J. Mach. Learn. Res. **20**(1), 809–846 (2019)
11. Trouillon, T., et al.: Knowledge graph completion via complex tensor factorization. J. Mach. Learn. Res. **18**(1), 4735–4772 (2017)
12. Raedt, L.D., Kersting, K.: Statistical relational learning. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 916–924. Springer, Boston, MA (2010)
13. Jensen, D.D.: Beyond prediction: directions for probabilistic and relational learning. In: Inductive Logic Programming. Springer, Berlin, Heidelberg (2008)
14. Neville, J., Jensen, D.: Relational dependency networks. J. Mach. Learn. Res. **8**, 653–692 (2007)
15. Getoor, L.: Statistical relational learning: unifying AI & DB perspectives on structured probabilistic models. In: Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, p. 183. Association for Computing Machinery, Chicago, Illinois (2017)
16. Eraslan, G., et al.: Deep learning: new computational modelling techniques for genomics. Nat. Rev. Genet. **20**, 389 (2019)
17. Abdullah, T., Ahmet, A.: Genomics analyser: a big data framework for analysing genomics data. In: Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, pp. 189–197. Association for Computing Machinery, Austin, Texas (2017)

18. Lediona, N., Marenglen, B.: Statistical relational learning for collaborative filtering a state-of-the-art review. In: Vishal, B. (ed.) Collaborative Filtering Using Data Mining and Analysis, pp. 250–269. IGI Global, Hershey, PA (2017)
19. Tillman, R.E.: Structure learning with independent non-identically distributed data. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1041–1048. Association for Computing Machinery, Montreal (2009)
20. Cao, L.: Data science: challenges and directions. Commun. ACM **60**(8), 59–68 (2017)
21. Imani, M., Braga-Neto, U.M.: Control of gene regulatory networks using Bayesian inverse reinforcement learning. IEEE/ACM Trans. Comput. Biol. Bioinform. **16**(4), 1250–1261 (2019)
22. Li, M., et al.: Automated ICD-9 coding via a deep learning approach. IEEE/ACM Trans. Comput. Biol. Bioinform. **16**(4), 1193–1202 (2019)
23. Zhang, Q., Zhu, L., Huang, D.S.: High-order convolutional neural network architecture for predicting DNA-protein binding sites. IEEE/ACM Trans. Comput. Biol. Bioinform. **16**(4), 1184–1192 (2019)
24. Gevaert, O., et al.: Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. Bioinformatics **22**(14), e184–e190 (2006)
25. Ghahramani, Z.: Probabilistic machine learning and artificial intelligence. Nature **521**(7553), 452–459 (2015)
26. Wang, H., Yeung, D.-Y.: A survey on Bayesian deep learning. ACM Comput. Surv. **53**(5), Article 108 (2020)
27. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques—Adaptive Computation and Machine Learning. The MIT Press (2009)
28. Larrañaga, P.: An introduction to probabilistic graphical models. In: Larrañaga, P., Lozano, J.A. (eds) Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, pp. 27–56. Springer US, Boston, MA (2002)
29. Pernkopf, F., Peharz, R., Tschiatschek, S.: Introduction to Probabilistic Graphical Models, pp. 989–1064 (2014)
30. Libbrecht, M.W., Noble, W.S.: Machine learning applications in genetics and genomics. Nat. Rev. Genet. **6**, 321 (2015)
31. Baker, L.A., et al.: Bayesian and machine learning models for genomic prediction of anterior cruciate ligament rupture in the canine model. G3: Genes|Genomes|Genetics **10**(8), 2619–2628 (2020)
32. Ojha, R., et al.: Bayesian network modelling for supply chain risk propagation. Int. J. Prod. Res. **56**(17), 5795–5819 (2018)
33. Biba, M.: Integrating Logic and Probability: Algorithmic Improvements in Markov Logic Networks. University of Bari, Bari (2009)
34. Heckerman, D., et al.: Dependency networks for inference, collaborative filtering, and data visualization. J. Mach. Learn. Res. **1**, 49–75 (2001)
35. Taskar, B., Chatalbashev, V., Koller, D.: Learning associative Markov networks. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 102. Association for Computing Machinery, Banff, Alberta (2004)
36. Domingos, P., et al.: Unifying logical and statistical AI. In: Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, pp. 1–11. Association for Computing Machinery, New York, NY (2016)
37. Genesereth, M.R., Nilsson, N.J.: Chapter 7—Induction. In: Genesereth, M.R., Nilsson, N.J. (eds.) Logical Foundations of Artificial Intelligence, pp. 161–176. Morgan Kaufmann, San Francisco (CA) (1987)
38. Dzeroski, S.: Relational data mining. In: Data Mining and Knowledge Discovery Handbook, 2nd ed., pp. 887–911 (2010)
39. Muggleton, S.: Inductive logic programming: derivations, successes and shortcomings. SIGART Bull. **5**(1), 5–11 (1994)
40. Riguzzi, F., et al.: Editorial: statistical relational artificial intelligence. Front. Robot. AI **6**(68) (2019)

41. Dragiev, S., et al.: An Abductive-Inductive Algorithm for Probabilistic Inductive Logic Programming, pp. 20–26 (2016)
42. Riguzzi, F., Bellodi, E., Zese, R.: A history of probabilistic inductive logic programming. Front. Robot. AI **1**(6) (2014)
43. Kersting, K.: An inductive logic programming approach to statistical relational learning. AI Commun. **19**(4), 389–390 (2006)
44. Fersini, E., Messina, E., Archetti, F.: Probabilistic relational models with relational uncertainty: an early study in web page classification. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (2009)
45. Roelleke, T., et al.: Modelling retrieval models in a probabilistic relational algebra with a new operator: the relational Bayes. VLDB J. **17**(1), 5–37 (2008)
46. Tamaddoni-Nezhad, A., Muggleton, S.: Stochastic refinement. In: Proceedings of the 20th International Conference on Inductive Logic Programming, pp. 222–237. Springer, Florence (2010)
47. Turliuc, C.-R.: ProbPoly: a probabilistic inductive logic programming framework with application in model checking. In: Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering, pp. 43–50. Association for Computing Machinery, Lawrence, Kansas (2011)
48. Raghavan, S., Mooney, R., Ku, H.: Learning to "read between the lines" using Bayesian logic programs. In: ACL (2012)
49. Anderson, C.R., Domingos, P., Weld, D.S.: Relational Markov models and their application to adaptive web navigation. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 143–152. Association for Computing Machinery, Edmonton, Alberta (2002)
50. London, B., et al.: AC-Bayesian Collective Stability, pp. 585–594 (2014)
51. Bunescu, R., Mooney, R.J.: Collective information extraction with relational Markov networks. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, pp. 438–es. Association for Computational Linguistics, Barcelona (2004)
52. Qiu, X., et al.: Recognizing inference in texts with Markov logic networks. ACM Trans. Asian Lang. Inf. Process. **11**(4), Article 15 (2012)
53. Garrette, D., Erk, K., Mooney, R.: Integrating logical representations with probabilistic information using Markov logic. In: Proceedings of the Ninth International Conference on Computational Semantics, pp. 105–114. Association for Computational Linguistics, Oxford (2011)
54. Biscarini, F., Cozzi, P., Orozcoter Wengel, P.: Lessons learnt on the analysis of large sequence data in animal genomics. Anim. Genet. **49**(3), 147–158 (2018)
55. Kazemi, S., Poole, D.: RelNN: a deep neural model for relational learning. In: AAAI (2018)
56. Khosravi, H., Bina, B.: A survey on statistical relational learning. In: Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence, pp. 256–268. Springer, Ottawa (2010)
57. Sun, S., et al.: Machine learning and its applications in plant molecular studies. Brief. Funct. Genomics **19**(1), 40–48 (2020)
58. Oliveira, A.L.: Biotechnology, big data and artificial intelligence. Biotechnol. J. **14**(8), 605–613 (2019)
59. Bose, A., et al.: Deep learning for brain computer interfaces. In: Balas, V.E., et al. (eds.) Handbook of Deep Learning Applications, pp. 333–344. Springer International Publishing, Cham (2019)
60. Krogel, M.-A., Scheffer, T.: Multi-relational learning, text mining, and semi-supervised learning for functional genomics. Mach. Learn. **57**(1), 61–81 (2004)
61. Sahab, M.G., Toropov, V.V., Gandomi, A.H.: Chapter 31—Optimum design of composite concrete floors using a hybrid genetic algorithm. In: Samui, P., Sekhar, S., Balas, V.E. (eds) Handbook of Neural Computation, pp. 581–589. Academic Press (2017)
62. Jain, R., Chotani, A., Anuradha, G.: 9—Disease diagnosis using machine learning: a comparative study. In: Lee, K.C. et al. (eds.) Data Analytics in Biomedical Engineering and Healthcare, pp. 145–161. Academic Press (2021)

63. Skënduli, M.P., Biba, M., Ceci, M.: Implementing scalable machine learning algorithms for mining big data: a state-of-the-art survey. In: Roy, S.S., et al. (eds.) Big Data in Engineering Applications, pp. 65–81. Springer Singapore, Singapore (2018)
64. Roy, S.S., Taguchi, Y.H.: Identification of genes associated with altered gene expression and m6A profiles during hypoxia using tensor decomposition based unsupervised feature extraction. Sci. Rep. **11**(1), 8909 (2021)
65. Roy, S.S., et al.: A hypothetical relationship between the nuclear reprogramming factors for induced pluripotent stem (iPS) cells generation—bioinformatic and algorithmic approach. Med. Hypotheses **76**(4), 507–511 (2011)
66. Chen, Q., Li, Y., Tan, K., Qiao, Y., Pan, S., Jiang, T., Chen, Y.P.P.: Network-based methods for gene function prediction. Brief. Funct. Genomics (2021)

A Study of Gene Characteristics and Their Applications Using Deep Learning

Prajjwal Gupta, Saransh Bhachawat, Kshitij Dhyani, and B.K. Tripathy

Abstract DNA sequencing deals with figuring out the order of arrangement of the bases in the DNA. These bases are the building blocks of DNA molecules and their arrangement mostly determines the genetic information carried within a DNA segment, therefore sequencing becomes a very important aspect in the field of genomics. Now it becomes ever more important to optimize this process of sequencing and analysis and the field of deep learning has a lot to offer. Autoencoders are artificial neural networks which are trained in an unsupervised manner to obtain feature representation or dimensionality reduction. Now as clustering is difficult to perform for data with large dimensions, autoencoders can be used to reduce the dimension of data by associating each gene cluster with an autoencoder. Genetic algorithms are algorithms which are based on Darwin's law of evolution and provide a better alternative to traditional clustering algorithms which have been found to have various drawbacks when implemented for genetic data. Drug repositioning is the examination of existing drugs on new disease targets and pharmacogenomics, looking to predict the target's response to a drug. Deep learning acts as a powerful tool for repositioning drugs by allowing us to perform robust predictions and provide deep insights to drug-disease combinations. This chapter aims to provide the reader with various deep learning models and analysis algorithms which have been employed in some or the other forms for studying gene characteristics and gene development or have the potential to form the basis for ground breaking research for the same.

Keywords Gene clustering · Autoencoders · Genetic algorithms · RNN · Sequencing · Drug repositioning

P. Gupta · S. Bhachawat
School of Computer Science and Engineering, VIT, Vellore, Tamil Nadu 632014, India
e-mail: prajjwal.gupta2019@vitstudent.ac.in

S. Bhachawat
e-mail: saransh.bhachawat2019@vitstudent.ac.in

K. Dhyani · B.K. Tripathy (✉)
School of Information Technology and Engineering, VIT, Vellore 632014, India
e-mail: tripathybk@vit.ac.in

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_4

1 Introduction

The field of genomics has seen an exponential rise of novel deep learning models and genetic algorithms for the analysis, characterization sequencing of genes and also in the development of various unconventional practices such as DNA storage and drug repositioning. Gene data is complex enough that it requires high computational power for processing and analysis. To serve this requirement, there has been a need for development of newer and advanced algorithms and models to tackle this computational problem. Development of new drugs or treatments for a new disease or a variant of an existing disease requires a lot of research, testing and manual work, these efforts can be greatly reduced through drug repositioning with the help of powerful deep learning models. Almost everything which we desire to analyze in the field of genomics, requires us to sequence the DNA, therefore DNA sequencing becomes another center point and an important aspect across a plethora of domains. The primary distinction between Deep Neural Networks (DNN) and other machine learning algorithms lies in their processing of raw natural data. While other machine learning algorithms require external experts for selecting the feature of raw data DNN has its own mechanism to select the features. In DNNs representational learning method is followed. Thus a set of methods are used, which determine the representation of raw data. DNN uses several levels, each level representing a non-linear module and finally composing all of these. The pattern followed is that higher layer modules are more abstract than those in the lower layers and thus the complex functions are learnt.

DNN algorithms have provided us with methods such that problems which were considered to be unsolved by using the other AI techniques could be solved. Different domains of government, business and science could apply DNN techniques because of its capability to handle high dimensional data [1]. For example, in the case of potential drug molecule predictions using several ML techniques are now closed and it has supported in analyzing particle accelerator data.

DNN techniques are applied in numerous application areas now. It has been used in Image processing areas [2]. In [3], the techniques used for the classification of Audio signals have been presented nicely. Among the first few applications of DNN is image retrieval through feature learning. In this direction an architecture is proposed in [4]. Also, in [5] a method for such information retrieval using text based techniques is discussed. Specific parts of human society require the recognition of sign languages and some techniques in this direction are presented in [6]. A useful variant of DNN is the Recurrent Neural Networks (RNN). RNNs are used in the study of sentiment analysis. Which in turn are useful in the study of human behaviour [7]. It goes without saying that the development and advancement of DNN are largely responsible for many advanced research in AI [8]. S-LSTM-GAN, which happens to be a shared neural networks model is proposed in [9] and has been established to be a very efficient method. Generative adversarial networks (GAN) happen to be an important development in the field of DNN, which is used to develop a single image

super resolution technique in [10]. Another important variant of DNN is the Convolutional Neural Network (CNN). The concepts associated with these models and their components are discussed in detail in [11] along with their working principles. A difficult but interesting application is to estimate the age and gender from images. However, a successful attempt using Wide ResNet has been proposed in [12].

Deep learning has enabled us to look beyond just the nucleotide sequence and has helped us discover hidden patterns and behaviours within the genomic sequence. Deep Learning models and techniques such as RNN are gaining more and more traction in the field of genomics since they do not require instructions to be specified to them explicitly. This traction has invited innovators to develop novel technologies with high accuracy and applicability, the latest developments are discussed in this chapter.

2 Deep Learning Techniques for Gene Clustering

2.1 Gene Clustering

Gene expression refers to the process by which the information encoded by a gene is used to synthesize a functional gene component/product which often produce and regulate protein as an end product. These proteins in turn define the function of the cell. Thus thousands of genes expressed together determine the function of the cell. At each step in the flow of information from DNA to RNA to proteins, the genes can control and regulate the type and amount of protein to be manufactured. A gene cluster is a group of genes which have similar encoding for polypeptides or proteins and share a common generalized function. They are usually within a few thousand base pairs of each other. Clustering and analysis provides an insight to various genetic and physical interactions, pattern recognition, cross-species analysis, mutation, evolution and development of superior or healthier gene sequences.

Since genetic data has a complex and high dimensional nature with high intra-variance, clustering of such data is difficult and expensive, resulting in low quality gene clusters. Thus there is a need to encode the gene data into smaller dimensions while retaining essential and sensitive information. Thus we can employ the use of Autoencoders to find an optimal feature representation of the gene data to optimize and increase the accuracy of the clustering algorithms.

Also further, traditional methods of sampling and clustering such as K-means produces low quality clusters which are not very useful in the development of high level gene sequences. Newer genetic clustering algorithms like GenClust++ and HEMI++ solve the drawbacks of K-means while producing high quality clusters.

2.2 *Autoencoders and Novel Techniques for Gene Clustering*

Autoencoders are artificial neural networks trained in an unsupervised manner to learn efficient data encodings/ feature representation. An autoencoder essentially performs dimensionality reduction by removing the input signal "noise" [13]. It provides a nonlinear data mapping to a space with lower dimensions (Fig. 1). An autoencoder has two parts: the encoder and decoder. The encoder is the non-linear mapping that we discussed above and the decoder ideally performs accurate reconstruction of the encoded data to the input data provided to the encoder. The aim of the autoencoder is to minimize the reconstruction error.

The autoencoder has to be sensitive enough so that it can accurately build a reconstruction and insensitive enough so that it doesn't overfit and memorize the training data. By this tradeoff, the model only maintains the variations required to reconstruct the data and redundancies are reduced.

Sensitive to the inputs enough to accurately build a reconstruction.

The loss function for such a model consists of two terms: the reconstruction loss, which is the difference between the original input data and reconstructed data, and a regularization term which is used to prevent overfitting. The sensitivity tradeoff can be regulated by multiplying a constant (λ) to the regularization term whose value is set between 0 and 1.

Now for clustering encoding can be implemented in two strategies: single autoencoder based clustering, association of each cluster with an autoencoder [14].

The first strategy involves training of the autoencoder model on the entire initial population. But this won't be an effective strategy as the model would learn only the common features between the ideal clusters and discard the features native to the clusters as noise. Thus the clustering algorithm followed by this type of encoding would produce undesirable results. The second strategy is a more apt one as it involves

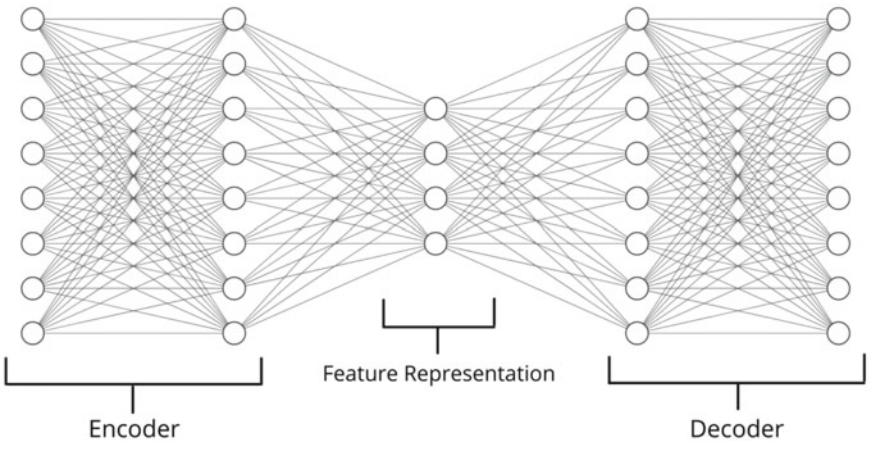

Fig. 1 Autoencoder components

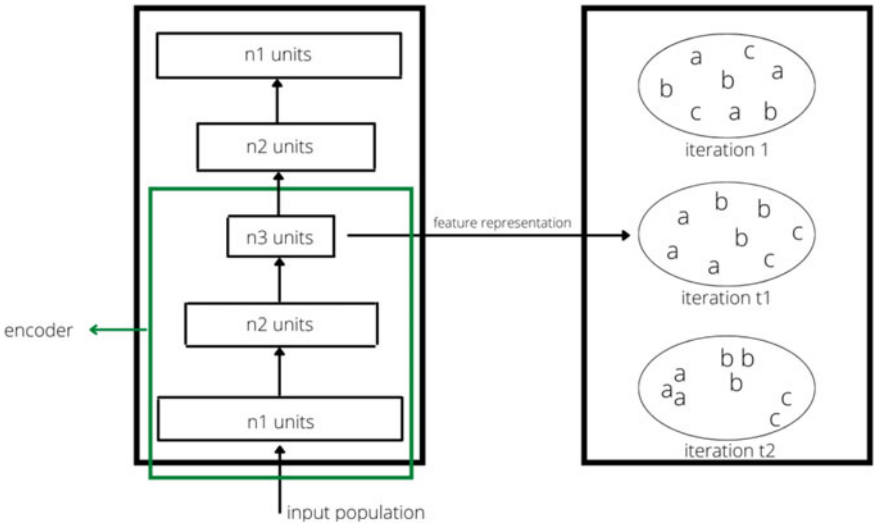

Fig. 2 Clustering using autoencoders

the association of each cluster with its own autoencoder, i.e. an autoencoder trained on the samples from the cluster itself (Fig. 2).

The algorithm for autoencoders based clustering [14] is: For dataset X, no. of clusters K (depending on the clustering algorithm, may or may not be required as a user input), hyper-parameter λ and number of iterations T perform:

Initialize cluster 0 (C0) randomly.

For iteration t (t <= T).

- Update the mapping network by minimizing the objective function [reconstruction loss + (λ * clustering cost function)] with gradient descent (or any other suitable algorithm)
- Update the cluster centers
- Partition X into K clusters and assign samples to respective clusters
- increment t.

Now onto clustering algorithms. K-means [15] and Fuzzy C-means are two of the simplest most commonly used algorithms used for clustering. However, they require the user to input the number of clusters k, but for genetic data it is difficult for a user to determine the number of clusters beforehand. Apart from that k-means heavily depends on the quality of initial seeds/samples and a bad set of initial seeds would result in the formation of bad quality clusters. K-means also has a tendency to get stuck in local minima, thus producing poor results.

To overcome these drawbacks, various genetics clustering algorithms have been proposed: AGCUK, GAGR, GenClust [16], HeMI [17], GenClust++ [18] and HeMI++ [19]. Genetic Algorithms (GA) are randomized search and optimization algorithms based on Darwin's law of evolution: Survival of the fittest. GA primarily

Table 1 Comparison of novel genetic clustering algorithms [9]

Algorithm	Complexity	Tree index	Silhouette coefficient rank[a]
K-means	O(N2)	27.41	4.26
K-Means++	O(N2)	31.01	4.40
GenClust	O(N2)	5.27	3.50
HeMI	O(N)	α	5.90
HeMI++	O(N)	0.55	1.13

[a] Silhouette coefficient rank computed on multiple datasets as illustrated in the paper

involves 5 steps: population initialization, selection, crossover, mutation and elitist operation.

GenClust algorithm produces high quality initial population. But the complexity of the algorithm is O(N2), N being the total number of records in the dataset. Also it requires the user to enter the radius of clusters for the initial population which is a difficult task for the user. HeMI also produces high quality clusters and it does not require any user input for cluster radii while keeping the complexity of initial population selection as low as O(N). GenClust++ is another algorithm for creating high quality of initial population with no user input and low complexity of O(N). It chooses the initial population probabilistically in contrast to random selection used in HeMI. HeMI picks half of the initial population from the set chromosomes obtained through K-means (k ranging from 2 to 10), and the rest of the half obtained from random k. HeMI++ is similar to HeMI but has an added advantage that it learns reasonable properties from the clusters formed and uses this information/knowledge to produce clustering solutions (Table 1).

2.3 HeMI++ Algorithm for Sensible Clusters [19]

HeMI++ algorithm selects only the best chromosomes depending on the fitness value, 50% of which come from a deterministic phase and 50% from random phase. The selection of the initial population in the deterministic phase is done by a k-means algorithm with k (number of clusters) ranging from 2 to 10. Datasets in which the number of clusters is more than 10, the range of k is set from 2 to $\sqrt{n}$ where n is the total number of records in the dataset. The HeMI++ algorithm described in the paper is as follows:

The user has to define/input the following parameters:

a. Number of streams (m)
b. Number of intervals (G)
c. Number of iteration (I).

The first step is to carry out normalization of the data points. Then carry out population initialization for each stream. Now select sensible properties for each cluster. For each iteration carry out the following operations in order: Noise-Based Selection, Crossover Operation, Twin Removal, Three steps Mutation Operation, Health Improvement Operation, Cleansing Operation, Cloning Operation and The Elitist Operation. Carry out "I" number of iterations for each stream. After finishing these operations for each stream, carry out Neighbour Information Sharing. After completing the Neighbour Information Sharing, proceed to the next interval. The steps are the same for each interval. After completion of all the intervals, perform Global Best Selection.

The components of this algorithm are:

i. **Normalization**: It refers to scaling the attributes of the data so that each attribute is treated equally irrespective of their domain size.
ii. **Number of streams**: HeMI++ uses a multiple stream approach in order to utilize a big population. The population is divided into multiple streams in which each stream contains a small number of chromosomes.
iii. **Population Initialization**: The initial population is selected in two phases: deterministic (p/2) and random (p/2) phase as discussed above. The value of p is set to 20 for HeMI++ .
iv. **Selection of sensible properties**: From the initial population, HeMI++ selects 'p' number of best chromosomes based on their DB index (goodness of fit). Each generation uses this sensible clustering solution so that chromosomes may not contradict the properties belonging to their cluster.
v. **Noise based selection**: Chromosomes from two generations are compared to select chromosomes for the next genetic operations.
vi. **Crossover Operation**: Crossover operation for chromosomes is performed in pairs. The best chromosome in the population is taken as the first chromosome and the second one is determined probabilistically. To perform the crossover operation, the chromosomes are divided into two segments, and one segment from the first chromosome is swapped with a segment from the second chromosome.
vii. **Twin Removal**: In some cases, genetic operations like mutation and crossover may create a condition wherein there are identical/twin genes in a chromosome. To handle such cases, HeMI++ performs twin removal. If there are twin genes in a chromosome and the length of the chromosome is more than two, then HeMI++ simply removes one of the genes. But if there are twin genes and the length of the chromosome is two, then HeMI++ randomly changes the attribute value of one of the genes.
viii. **Three step mutation operation**: Division, absorption and random changes are performed on chromosomes in this step.
ix. **Health improvement operation**: This operation ensures that the health of the chromosomes is consistent or improved in each generation.
x. **Cleansing operation**: This operation aims to identify the sensible and non-sensible chromosomes belonging to the population.

xi. **Cloning operation**: In this operation, the sick chromosomes found in the cleansing operation are replaced.
xii. **The elitist operation**: In this operation, the best chromosome throughout the generation is taken and passed to the next generation in order to improve the quality of population in the next generation.
xiii. **Neighbour information sharing**: It involves sharing of the best chromosomes among the neighbouring streams at a regular interval like every 10th iteration.
xiv. **Global best selection**: This component is used to find the best chromosome across all streams.

3 DNA Sequencing Using RNN

We are all essentially just carbon atoms, attached with functional groups made up of hydrogen and oxygen. This coherent existence of carbons along with other functional groups gives rise to various kinds of proteins. These proteins along with water, make up what we know as the cell. At the center of this cell lies the nucleus, inside which comfortably sits the key to evolution—the DNA. The information stored inside the DNA is the directing force of protein production, which in turn are vital for physical, mental, biological growth and development. It also carries the hereditary material in almost all organisms, and hence has crucial biological and evolutionary importance. Owing to its vital significance in multiple aspects, it becomes essential that we understand its constituents to attain a better understanding. DNA is made up of four bases, namely adenine (A), cytosine (C), guanine (G), and thymine (T). The sequence of these very bases determines the protein assembly instructions. Therefore sequencing has become the center of almost every development in the fields of biotechnology, molecular biology, forensics, etc. Attempts to sequence genomic data have been made since the mid-1900s, ranging from Sangers sequencing methods to the recent concepts of nanopore sequencing. The latest to join this arsenal of sequencing are the Deep learning techniques. The popularity of Deep Learning techniques in genomics could be accredited to the ability of the DL techniques to learn relevantly new features from the old features without being told what to do. The most accepted architecture amongst all, are the Recurrent Neural Networks, due to their special ability to support feedback connections to previous layers. Deep learning techniques have leaped past simple sequencing of the base sequence, and have dived deeper into finding the intricacies within the genomic data. Novel innovations using DL architectures can be seen emerging in this new space of innovation [20].

3.1 Applications of DNA Sequencing

DNA Sequencing has become a center point of innovation across many areas of biology since the sequence of bases in the DNA determines almost all characteristics

showcased by a majority of the organisms. As a result, it has countless applications across many domains.

3.1.1 Molecular Biology

Molecular Biology studies the composition of cellular molecules and has special interests in nucleic acids and proteins. To study these very nucleic elements we gravely require fast sequencing technologies. This helps us understand and identify phenotypes, associate certain types of gene change with diseases, also to understand the effect of medicine and drugs on specific gene types. Deep learning has become popular amongst molecular biologists since it helps them dive deeper into the biological secrets of the DNA sequence.

3.1.2 Virology

Virology has gained special popularity after the pandemic onset. Viruses enjoy benefits due to their ultra-small size, this renders them invisible to a light microscope, and dealing with them extremely difficult. This makes sequencing the only tool which gives us insights into the structure and behavior of viruses. Sequencing methodologies proved especially useful to sequence the genome of the Coronavirus. Deep learning techniques have allowed us to understand further depths from the obtained sequences.

3.1.3 Forensics

Forensics is often the turning point of many criminal cases, with faster and more robust sequencing techniques, we can attain speedy results on DNA profiling and paternity testing. This has resulted in the resolution of many pending criminal cases, and also the release of a few wrongly accused. Deep learning techniques have accelerated the speed of advancements by multiple folds.

3.1.4 Metagenomics

Metagenomics aims to identify all the different types of microbes present in a microbial ecosystem. Faster and next-generation sequencing techniques have enabled us to identify new and different microbes within a microbiome. Many Deep learning models are being proposed which perform taxonomical classification with higher accuracy than the state of the art genus identification tools. Deep Microbes is one such DLM that showed an accuracy of around 89.40% [21].

3.1.5 Medicine

DNA majorly controls the physical mental and biological characteristics portrayed by all living organisms. Therefore through DNA sequencing, we can get insights into how our body might react to certain medicinal compounds; moreover, it helps us identify genetic disease in the early stages. Additionally, we can sequence the DNA of the bacterial genomes, which might give us insights into developing more effective antibiotics and treat disease more effectively.

3.2 Latest Advances in DNA Sequences Using Deep Learning

Deep learning has identified various new vertices in DNA sequencing, which has developed a whole new level of understanding of biological processes. We have gained deeper insights into working at cellular levels and microscopic interactions.

3.2.1 Splice Junction Prediction in DNA Sequence Using Multi-Layer RNN Model

Genes are made of up DNA and hence are the driving force for protein synthesis. At times a single gene can take charge of synthesizing multiple proteins, this is due to the post-transcriptional modification. This post-transcriptional modification removes intron as shown in Fig. 4 enzymatically and this site of removal is known as a splice junction.

These splice joints are of special concern since they assist in understanding the protein that is being synthesized and its nature, function, and characteristics. These splice junctions can be predicted by sequencing the DNA and then analyzing it, and the Deep Learning technique utilizing RNN seems to perform this prediction with high accuracy [22]. A three-layered RNN model utilizing the Molecular Biology data set by Murray et al. achieved an accuracy of 99.95%.

3.2.2 Deep Learning Method for Identification of Short Viral Sequences from Metagenomes

Viruses and their high mortality effects are something we have all become familiar with after the Covid-19 epidemic. Therefore identifying the viral sequences and understanding their characteristics from those sequences is essential. An RNN based deep learning model is discussed named VinSeeker, which exhibits higher accuracy compared to other standard virological sequencing models for short sequences. The data set on which the VinSeeker was trained, contained sequences of well-known Virus genomes. To fully prove the technology the proposed mechanism was also trained and tested on a CAMI dataset and also a dataset involving the human gut

metagenome [23]. RNN based VinSeeker outperformed similar sequencing models with a greater AUROC score and precision.

3.2.3 Chiron: Translating Nanopore Raw Signal Directly into Nucleotide Sequence Using Deep Learning

Nanopore Sequencing is a rapidly growing and advancing technology. It possesses the capabilities to not only sequence long DNA/RNA fragments at high speeds but also to generate entire genome assemblies, spot modified base sequences in a DNA, and recognize and classify microbes in a metagenome. The technology at the base root employs a bed of membrane wells which each contains a nanopore. Nanopores have an active electrical current running through them. These nanopores are further augmented with a few tethers which allow the DNA strands to be linked with the nanopore opening. Once linked, the nucleic acid-containing DNA strands are run through the nanopore, this causes the calculated disruptions within the electric current running through the nanopore, and the resulting electric signal obtained is then decoded to get the DNA sequence. This decoding process is where the Chiron steps in, it is a hybrid neural network that coherently integrates an RNN and a CNN along with a CTC decoder (Connectionist Temporal Classification) as shown in Fig. 5.

This was trained and tested on multiple genomes namely Lambda virus, *E. coli*, and human, and showed to perform better than the other market standards [24].

3.2.4 Biren: Predicting Enhancers with a Deep-Learning-Based Model Using the DNA Sequence Alone

Transcription is a very vital process, as it is responsible for the expression of the genes and controls the extent of development of many characteristics in different types of cells. This transcription process is amplified when enhancer sequences attach themselves to specific proteins. These enhancer sequences have high importance since they majorly control and regulate the gene expression process. Therefore predicting and identifying these enhancer sequences can be very beneficial and could give us insight into many challenges being faced by the bioinformatics industry. However, accurate enhancer prediction methods are absent.

A recent solution being proposed utilizes a hybrid of Convolution neural networks and a modified Recurrent Neural Network (RNN). Regular RNN's support only forward states progression. The proposed solution employs a concept of the bidirectional flow of input information. Bidirectionality allowed the network to extract features from the DNA sequence more efficiently. This was the Neural Network can process input data in both positive and negative time direction. This modified RNN model is called the Bidirectional recurrent neural network (BRNN) and the proposed technology is being called BiRen. BiRen essentially aims to directly identify enhancer sequences from a DNA sequence. This is achieved by deploying hybrid architecture of CNN and BRNN. The CNN half employs the DeepSEA model which

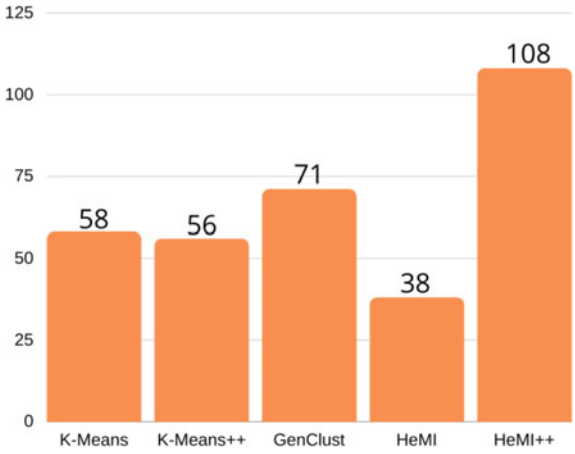

Fig. 3 Performance scores of clustering algorithms

encodes the original DNA Sequence into label vectors. The BRNN half predicts the probability of enhancer sequences after learning from the features obtained from the label vectors. Both the CNN and BRNN work in harmony and utilize the strengths of each other to showcase excellent potential. BiRen showed better performance than other such enhancer prediction models with an AUC of 0.956 [25].

Another proposed model based on a Bidirectional Gated Recurrent Unit network with k-mer embedding called KEGRU showed promise in identifying the transcriptional factors binding site. The KEGRU model divides the DNA sequence into k-mer sequences. Each k-mer sequence is treated as a word and fed to the word2vec algorithm for being vectorized (Fig. 6).

This data is then used by the deep Bidirectional Gated Recurrent Unit for the learning of features. Once the features are learned, the classification into TF and non-TF binding sites is performed as shown in Fig. 3. KEGRU model possesses an advantage over other models crediting to its robustness [26].

3.2.5 A Deep Learning Model for Predicting NGS Sequencing Depth from DNA Sequence

Next-generation sequencing (NGS) techniques are vastly based around parallelism's distributive nature. This has resulted in most NGS protocols being centered on the breaking of genome sequences into shorter fragments. These shorter genome fragments are then read individually and later computationally overlapped to make meaningful sequences. This results in each nucleotide being sequenced multiple times over and over. This is where the concept of Sequencing Depth comes into the picture, the number of iteration of reading, a nucleotide goes under is known as the Sequencing Depth. Sequencing depths have a great impact on sequencing cost and also on the accuracy of the models being used. Hence understanding the sequencing depth can sometimes be very essential for a project.

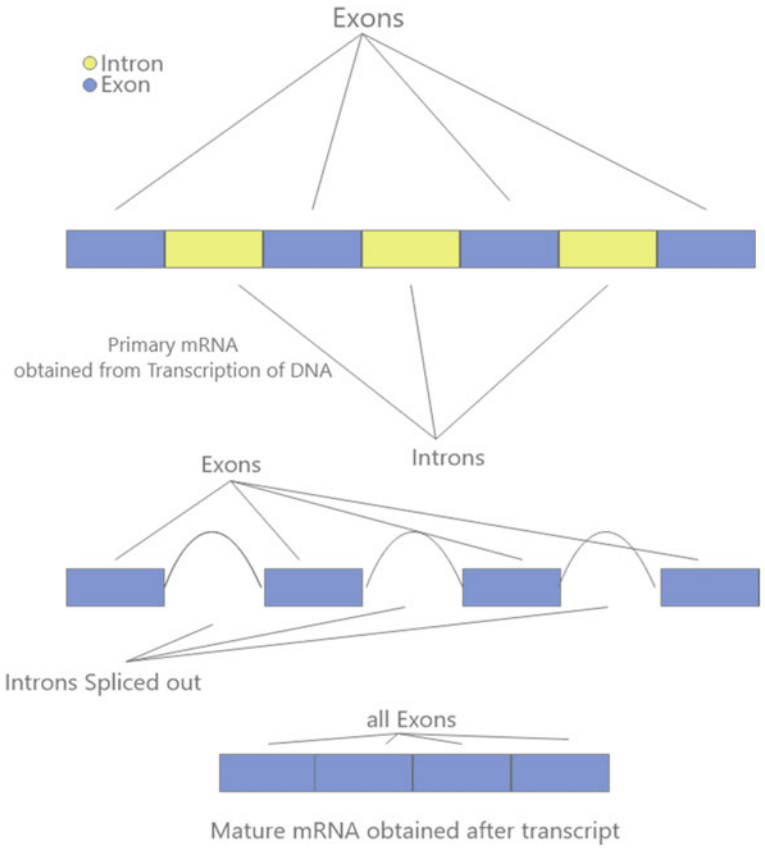

Fig. 4 Splicing of genes

To solve this challenge, a deep learning model is being conceptualized which takes the DNA sequence and probability of parity as input and predicts with accuracy the sequencing depth of the different NGS panels. It is built upon a recurrent neural network to facilitate the capture of both short-range and long-range interaction using gated recurrent units (GRUs) within the DNA. The proposed deep learning model (DLM) was tested on 2 different NGS panels with an accuracy of 99% and 93% [27].

3.2.6 Deep Recurrent Neural Network for Protein Function Prediction from Sequence

DNA stores the instructions to produce various kinds of proteins and these proteins, in turn, help in carrying out various functional processes. However, understanding the exact functions of these numerous proteins has been a long-standing challenge.

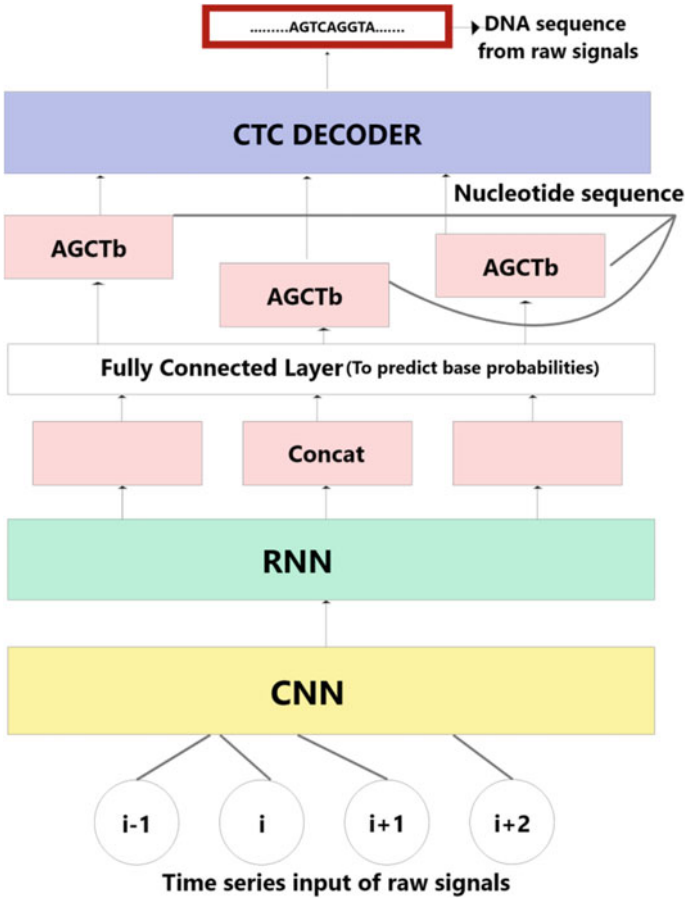

Fig. 5 Chiron- an RNN and CNN hybrid deep learning model architecture

A deep learning model based on recurrent neural networks appears to solve this by converting it into a problem of classification, where classes represent the protein function. The RNN model is powered by a long short term memory (LSTM) unit, and trained on a UniProt dataset [28]. The proposed model seems to show several advantages over similar tools and has several potential benefits.

3.3 Future Scope

The extent and the speed at which we can sequence the DNA plays a crucial role across multiple disciplines. Therefore it has been attracting innovators and research to develop ever faster sequencing technologies and techniques. The future is very

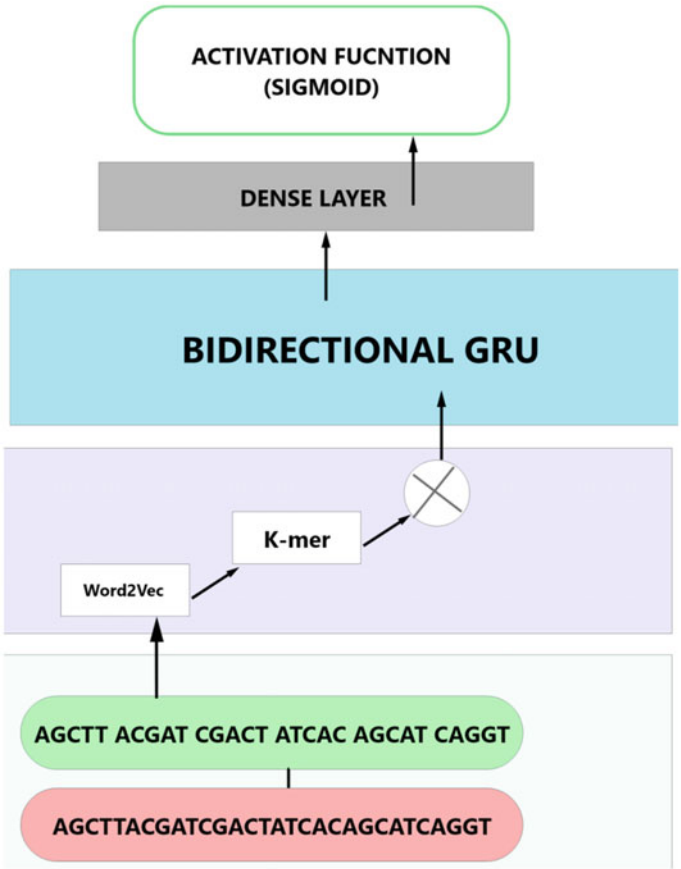

Fig. 6 Deep learning model architecture of KEGRU for predicting enhancers

bright as new discoveries are being uncovered in the field of biology that might just be the next big break for DNA sequencing [29].

3.3.1 DNA Storage

Every day millions of bytes of data are being stored and processed, at this rate, there is an inevitable data explosion just around the corner. There is not enough metal on earth to sustain the upcoming data needs. However, there are a few technologies that could help us contain this data explosion, one such being DNA storage. We know that DNA already stores information and instructions about protein synthesis proteins. Therefore it has become a very real possibility to store data in the DNA as well. DNA has a much greater storage density and can store quantities of magnitude much greater than any present technology on the planet [30]. Storing the data is

just the first half of the story, we need fast sequencing techniques to be able to read data at high speeds, since without high-speed reading capabilities, storing large data in DNA is like throwing it down an inaccessible well. The increasing popularity of DNA sequencing and the in-pour of funds in the Bioinformatics industry could result in the entrance of DNA based storage devices in the market very shortly.

3.3.2 Exploration of Genome Diversity

Countless genomes are waiting to be sequenced and we have just barely scratched the surface. With concepts such as Metagenomics picking up attention, we are regularly gaining insights into microbial diversity. Countless biomes possess rich genomic diversity and gaining a large scale understanding of these different genomic sequences can be beneficial in understanding why the DNA functions the way it functions. This will also give information about protein structure determination and an overall better understanding of the environment we see around ourselves.

3.3.3 Developmental Biology

We are already capable of cultivating and nurturing many plants and animals in labs under delicate and mindful conditions. However, we lack knowledge of the actual development process on a cellular level. Some events and processes such as the gene expressions can be observed and analyzed in a more broad and general way by editing the genes. This editing requires sequencing techniques and through editing we can observe the effects of each edit and develop a better understanding. Many deep learning solutions have emerged in the past couple of years which bring promising possibilities to the table [31].

3.3.4 Extensive Monitoring of Nucleic Acids

Technologies such as nanopore sequencers allow us to sequence data at high speeds for a specific genome. A disseminated, extensive, well-coordinated network of nanopore sequencers could result in a real-time monitoring system of nuclide acids. This could result in innovative applications such as real-time air and water quality testing, food tasting, and human body testing.

4 Deep Learning for Repositioning of Drug and Pharmacogenomics

Time and again new diseases and disorders are discovered all over the globe. The HIV, Covid-19 pandemic and Cushing syndrome are some examples of diseases which have affected numerous people all over the world. We are constantly in the process of developing cures for these diseases but the process of finding and developing a new cure is often very time consuming and requires very large amounts of funds, therefore making us unable to provide quick cures for these diseases and also leads to side-lining and neglecting of the rare diseases. To overcome these issues drug manufacturers often use an alternative method called drug repositioning or repurposing. Drug repositioning is the examination of existing drugs on new disease targets and pharmacogenomics potentially looking to predict the target's response to a drug, potentially saving many lives and time by reducing the cost and time to produce a new drug. Drug repositioning has been a booming field of research in the past year as the need for faster drug production echoed throughout the planet. Repositioning of drugs is also significantly cheaper than building your own drug from the scratch and the safeties of the repositioned drugs are pre-determined from previous preclinical tests. Viagra is a famous example of a repositioned drug where although its initial use was meant for curing heart related diseases, now it is widely used to cure erectile dysfunction. Another example of this is the "Thalidomide" medicine which was highly detrimental when it was released initially, was successfully repositioned and transformed into an effective cancer drug therapy [32].

During drug repositioning features of drugs are taken along with tests taken across on patients diagnosed with a disease. This is then passed into a Deep Learning model which uses various algorithms such as autoencoders, variational autoencoders, deep walking etc. These algorithms provide us with the architecture to determine and compare similarities between the drug features, drug-disease, drug-side-effect, drug-target, etc. It is through these similarities that the model predicts potential drugs for the target diseases. A large abundance of such data, increases the likelihood of predicting the drug. With Deep Learning models becoming extremely powerful with predictive power, drug repositioning is becoming more and more feasible by the day.

4.1 Applications of Drug Repositioning

Drug repositioning has been cited as one of the best alternatives to manufacturing new drugs. It reduces cost and is a very time efficient process. It also provides a much higher safety than manufacturing a new drug. Drug repositioning is being used to battle various diseases which require immediate solutions or are very costly and rare to develop a new drug. Cancer and Covid-19 are some of the diseases where drug repositioning is being used for a cure [33].

4.2 Novel Deep Learning Methods for Drug Repositioning

As more and more companies are getting attracted towards drug repositioning there has been a significant rise in the research for more powerful deep learning models. These models incorporate a number of deep learning algorithms together.

Some of these algorithms are [34]:

(i) **Logistic regression and kernel regression**: Logistic regression model is used to estimate a binary dependent variable and predict the probability of a certain class. A Kernel regression on the other hand is used to estimate the expectation of a random variable based on certain conditions.
(ii) **Random forest**: A random forest is an ensemble method for prediction, classification, estimation etc. It works by constructing multiple decision trees and giving the output of the class which is the mode or mean of the prediction outputs of individual trees.
(iii) **Autoencoder**: An autoencoder is an artificial network used to obtain a feature representation of high dimensional data (as discussed in Sect. 3.2).
(iv) **Variational autoencoders**: A variational autoencoder is used to generate the probability of the latent vector in a confined space to represent the high dimensional data.
(v) **Support Vector machine**: Support vector machine algorithm is used to find an optimal hyper plane to classify different data points.

A Deep learning model for repositioning of drugs needs to follow certain steps (Fig. 1) in order to identify new scope for existing drugs. The following five steps fulfil this requirement [35]:

(i) **Representing drug features**: In the first step i.e. the drug feature representation we take as input data of various drug features such as the chemical structure and use neural networks such as autoencoders to perform PCA and dimensionality reduction and obtain an optimal feature representation.
(ii) **Transforming disease features**: In this step we aim to identify disease-disease similarities. First we convert the categorical data to a form that can be understood by the model using the one hot encoder technique. After the conversion of the data we determine the disease-disease similarities using PCA.
(iii) **Using drug features to construct the drug-drug similarity matrices**: We use the extracted drug features from the previous steps to compare the properties/features of different drugs and construct a similarity matrix.
(iv) **Using drug features to construct the disease-disease similarity matrices**: We use the extracted disease features from the previous steps to compare the properties/pharmacogenomics of different diseases and construct a similarity matrix.

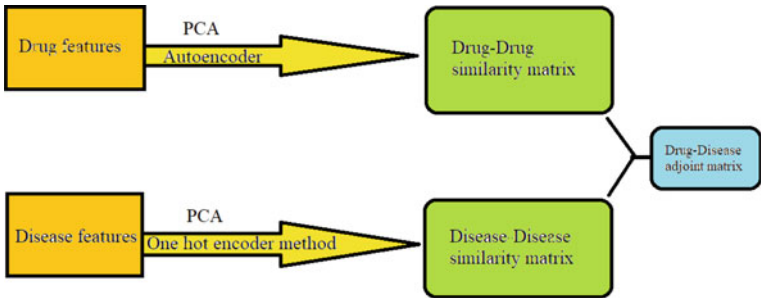

Fig. 7 Steps followed by a data learning model

(v) **Using drug-drug similarity and disease-disease similarity to construct drug disease association matrices**: We use the information from the existing drug and disease matrices to form an association matrix where we once again compare the similarities between the drugs and diseases to form drug, disease pairs (Fig. 7).

Let us now discuss a novel deep learning model based on the methods mentioned above [36]:

The deepDR neural network architecture has been developed to facilitate repositioning of drugs by integrating various networks such as the drug-drug, drug-disease, drug-side-effects etc. The architecture uses autoencoders to determine the high-level features of the drugs from the network. Then the low-dimensional representations of these features are encoded and decoded through a variational autoencoder. Thus it finally approves drugs for some shortlisted candidates which were not originally approved for these drugs.

5 Future Scopes

As discussed above drug repositioning is a widely supported alternative to novel drug manufacturing. Having already seen successful results such as thalidomide, Viagra, etc., drug repositioning has become an accepted method at various drug research institutions. Drug repositioning also saw a significant boost in research in the year 2020 due to the covid-19 pandemic. It is seen as a good method to find a cure for covid-19 due to the urgent requirement of a vaccine. It has also been used for finding cures for various other diseases including different types of cancers. Drug repositioning not only helps manufacturers save money in building a new drug, it also recycles the existing drug in the market. During the current circumstances drug repositioning can safely be called the next big step in the drug manufacturing industry and so we can also expect better and more advanced deep learning models to be built to assist with these requirements.

6 Conclusions

In this chapter we saw how gene analysis on different levels can form the basis for extraction of characteristics on higher levels. We talked about clustering the genes based on existing sequences and gene expressions and how genes taken from these clusters can be sequenced in a specific manner to find applications in various fields. Such a method can be employed in developing mutations, predicting successful cross breeding sequences, development of new drugs or repositioning the drugs to treat some other diseases. Further refinement of the approaches or development of new ones can significantly help the medical world in the development of newer treatment practices and drugs in a lesser amount of time. Some deep learning models like a RNN can also pave the way to the commercial scaling of DNA storage which could be the next ground breaking industry in the upcoming years.

References

1. Biswas, R., Vasan, A., Roy, S.S.: Dilated deep neural network for segmentation of retinal blood vessels in fundus images. Iran. J. Sci. Technol. Trans. Electr. Eng. **44**(1), 505–518 (2020)
2. Adate, A., Tripathy, B.K.: Deep learning techniques for image processing. In: Machine Learning for Big Data Analysis Berlin, pp. 69–90. De Gruyter, Boston (2018)
3. Bose, A., Tripathy, B.K.: Deep learning for audio signal classification. In: Deep Learning Research and Applications, pp. 105–136. De Gruyter Publications (2020)
4. Garg, N., Nikhitha, P., Tripathy, B.K.: Image retrieval using latent feature learning by deep architecture. In: Proceedings of IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–4 (2014)
5. Singhania, U., Tripathy, B.K.: Text-based image retrieval using deep learning. In: Encyclopedia of Information Science and Technology, 5th edn., pp. 87–97. IGI Global, USA (2020)
6. Prakash, V., Tripathy, B.K.: Recent advancements in automatic sign language recognition (SLR). In: Computational Intelligence for Human Action Recognition, pp. 1–24. CRC Press (2020)
7. Baktha, K., Tripathy, B.K.: Investigation of recurrent neural networks in the field of sentiment analysis. In: Proceedings of IEEE International Conference on Communication and Signal Processing, pp. 2047–2050 (2017)
8. Adate, A., Tripathy, B.K., Arya, D., Shaha, A.: Impact of deep neural learning on artificial intelligence research. Deep Learn. Res. Appl. De Gruyter Publications **7**, 69–84 (2020)
9. Adate, A., Tripathy, B.K.: S-lstm-gan: Shared recurrent neural networks with adversarial training. In: Proceedings of the 2nd International Conference on Data Engineering and Communication Technology, pp. 107–115. Springer, Singapore (2019)
10. Adate, A., Tripathy, B.K.: Understanding single image super resolution techniques with generative adversarial networks. Adv. Intell. Syst. Comput. Springer, Singapore **816**, 833–840 (2019)
11. Maheshwari, K., Shaha, A., Arya, D., Rajasekaran, R., Tripathy, B.K.: Convolutional neural networks: a bottom-up approach. Deep Learn. Res. Appl. **7**, 21–50 (2019)
12. Debgupta, R., Chaudhuri, B.B., Tripathy, B.K.: A wide Resnet-based approach for age and gender estimation in face images. In: Proceedings of International Conference on Innovative Computing and Communications, pp. 517–530. Springer, Singapore (2020)
13. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P. A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning, pp. 1096–1103 (2008)

14. Song, C., Liu, F., Huang, Y., Wang, L., Tan, T.: Auto-encoder based data clustering. In: Iberoamerican congress on pattern recognition, pp. 117–124. Springer, Berlin, Heidelberg (2013)
15. Krishna, K., Murty, M.N.: Genetic K-means algorithm. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **29**(3), 433–439 (1999)
16. Rahman, M.A., Islam, M.Z.: A hybrid clustering technique combining a novel genetic algorithm with K-Means. Knowl.-Based Syst. **71**, 345–365 (2014)
17. Beg, A.H., Islam, M.Z., Estivill-Castro, V.: Genetic algorithm with healthy population and multiple streams sharing information for clustering. Knowl.-Based Syst. **114**, 61–78 (2016)
18. Islam, M.Z., Estivill-Castro, V., Rahman, M.A., Bossomaier, T.: Combining K-Means and a genetic algorithm through a novel arrangement of genetic operators for high quality clustering. Expert Syst. Appl. **91**, 402–417 (2018)
19. Beg, A.H., Islam, M.Z., Estivill-Castro, V.: HeMI++: a genetic algorithm based clustering technique for sensible clusters. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2020)
20. Dutta, P., Patra, A.P., Saha, S.: DeePROG: deep attention-based model for diseased gene prognosis by fusing multi-omics data. IEEE/ACM Trans. Comput. Biol. Bioinform (2021)
21. Liang, Q., Bible, P.W., Liu, Y., Zou, B., Wei, L.: DeepMicrobes: taxonomic classification for metagenomics with deep learning. NAR Genom. Bioinform. **2**(1), lqaa009 (2020)
22. Sarkar, R., Chatterjee, C.C., Das, S., Mondal, D.: Splice junction prediction in DNA sequence using multilayered RNN model. In: International Conference on Emerging Trends in Engineering, pp. 39–47. Springer, Cham (2019)
23. Liu, F., Miao, Y., Liu, Y., Hou, T.: RNN-VirSeeker: a deep learning method for identification of short viral sequences from metagenomes. IEEE/ACM Trans. Comput. Biol. Bioinform. (2020)
24. Teng, H., Cao, M.D., Hall, M.B., Duarte, T., Wang, S., Coin, L.J.: Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. GigaScience **7**(5), giy037 (2018)
25. Yang, B., Liu, F., Ren, C., Ouyang, Z., Xie, Z., Bo, X., Shu, W.: BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. Bioinformatics **33**(13), 1930–1936 (2017)
26. Shen, Z., Bao, W., Huang, D.S.: Recurrent neural network for predicting transcription factor binding sites. Sci. Rep. **8**(1), 1–10 (2018)
27. Zhang, J., Yordanov, B., Gaunt, A., Wang, M., Dai, P., Chen, Y.J., Zhang, D.: A deep learning model for predicting NGS sequencing depth from DNA sequence (2020)
28. Liu, X.: Deep recurrent neural network for protein function prediction from sequence (2017). arXiv preprint arXiv:1701.08318
29. Shendure, J., Balasubramanian, S., Church, G.M., Gilbert, W., Rogers, J., Schloss, J.A., Waterston, R.H.: Publisher Correction: DNA sequencing at 40: past, present and future. Nature **568**(7752), E11–E11 (2019)
30. Koch, J., Gantenbein, S., Masania, K., Stark, W.J., Erlich, Y., Grass, R.N.: A DNA-of-things storage architecture to create materials with embedded memory. Nat. biotechnol. **38**(1), 39–43 (2020)
31. Tasaki, S., Gaiteri, C., Mostafavi, S., Wang, Y.: Deep learning decodes the principles of differential gene expression. Nat. Mach. Intell. **2**(7), 376–386 (2020)
32. Low, Z.Y., Farouk, I.A., Lal, S.K.: Drug repositioning: new approaches and future prospects for life-debilitating diseases and the COVID-19 pandemic outbreak. Viruses **12**(9), 1058 (2020). https://doi.org/10.3390/v12091058
33. Xue, H., Li, J., Xie, H., Wang, Y.: Review of drug repositioning approaches and resources. Int. J. Biol. Sci. **14**(10), 1232 (2018)
34. Luo, H., Li, M., Yang, M., Wu, F.X., Li, Y., Wang, J.: Biomedical data and computational models for drug repositioning: a comprehensive review. Brief. Bioinform. (2020)
35. Moridi, M., Ghadirinia, M., Sharifi-Zarchi, A., Zare-Mirakabad, F.: The assessment of efficient representation of drug features using deep learning for drug repositioning. BMC Bioinform. **20**(1), 1–11 (2019)

36. Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., Cheng, F.: deepDR: a network-based deep learning approach to in silico drug repositioning. Bioinformatics **35**(24), 5191–5198 (2019)

Computational Biology in the Lens of CNN

Pranjal Bhardwaj, Thejineaswar Guhan, and B.K. Tripathy

Abstract Throughout this chapter the objective is to bring deep learning techniques and algorithms, specifically CNN, which bring about ease for a researcher with respect to time and resources. The concepts are explained as an overview to implant an intuition of the techniques which can be further elaborated with the mathematics in the references. Computational biology involves the examination of how proteins interact with each other through the simulation of protein folding, motion, and interaction. Current computational biology research can be divided into a number of broad areas, mainly based on the type of experimental data that is analyzed or modeled. Deep learning and in particular, Convolutional Neural Networks (CNNs) has brought about a revolution for the analysis of gene expression images. This technique solves some of the setbacks faced by traditional machine learning approaches while advances in technology have enabled the capture of gene sequence images, while in some cases non-image data captured can be converted to an image for analysis.

Keywords Genomics · Computational biology · Biological imaging · Gene expressions · Autoencoders · Convolutional neural networks (CNN)

P. Bhardwaj
School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Vellore, Tamil Nadu 632014, India
e-mail: pranjal.bhardwaj2019@vitstudent.ac.in

T. Guhan · B.K. Tripathy (✉)
School of Information Technology and Engineering, Vellore Institute of Technology (VIT), Vellore, Tamil Nadu 632014, India
e-mail: tripathybk@vit.ac.in

T. Guhan
e-mail: thejineaswar.guhan2019@vitstudent.ac.in

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_5

1 Introduction

The characteristic of processing natural data in their raw form distinguishes Deep Neural Networks (DNN) from other machine learning approaches. For the representation of raw data internally ML algorithms other than DNN require external experts, whereas DNN develops its own feature vector. In representational learning, a set of methods are used, which are capable of discovering representation of raw data of its own. Representational learning is followed by DNNs so that raw data are represented automatically for classification or detection. Simple non-linear modules are composed using several levels of representation in DNN. The modules in succeeding layers are more abstract than their predecessors. This is how a DNN learns complex functions.

Several problems which were previously unsolved through the other AI techniques could be solved by using DNN algorithms. Its capability to handle high-dimensional data helped in applying it in different domains of government, business, and science. As a consequence, other ML techniques in predicting potential drug molecules were superseded and it helped in analyzing particle accelerator data.

There are numerous applications of DNN in various areas. Image processing is one of these areas [1]. Audio signal classification is one of the object classification areas where DNNs are useful tools. In [2] the approaches in literature dealing with audio signal classification are discussed. Using feature learning for image retrieval is one of the first few applications of Deep Learning. In [3] an architecture has been proposed in this direction. Proposals developed so far in the direction to recognize sign languages, which are useful in specific parts of human society, have been discussed in [4]. A text-based approach for information retrieval is discussed in [5]. Recurrent Neural Networks (RNN) is used in sentiment analysis, which is a useful ingredient of the study of human behavior as proposed in [6]. AI has been enriched by the advancement of DNN techniques [7]. A shared neural networks model called S-LSTM-GAN is proposed in [8]. A single image super-resolution technique using Generative adversarial networks (GAN) in [9]. Convolutional Neural Network (CNN), is a useful variant of DNN. The concepts and components associated with it, are presented in [10] in a lucid manner along with their working principles. A useful task although a difficult one is to estimate Age and gender n images. In an attempt to realize this, a Wide ResNet-Based approach is proposed in [11].

With advances in deep learning, practitioners began to use it across genomics, and eventually, it took the subject by storm. Usage of various algorithms, architectures, and techniques has changed the way experiments are conducted. These forms of experiments have not only made the process rapid but accessible to many thanks to the availability of online data repositories which contain various experimental results. So, the question arises why Deep learning? Deep learning solves the problem faced by traditional machine learning models, feature engineering. This process is automated in the case of deep learning and in CNN these features can be easily identified. The forms of data are in various forms: from sequenced DNA to images which are then used to train the models. Throughout this chapter, the spotlight is shed on various

modeling and preprocessing techniques used to predict various problem statements. These models are then made clear with the usage of their application. Alongside, tools used for these processes are discussed to give a practical sense to the reader.

2 Deep Learning for Computational Biology

2.1 What is Computational Biology?

Computational biology is a branch of biology which involves the applications of mathematics, statistics, and computer science fundamentals to the efficient understanding of genomes and predictive modeling. It mainly involves deriving results from how proteins interact with one another through the simulation of protein folding and their interactions. Research in the field of computational biology can be divided into several broad domains, primarily depending on the type of data being analyzed or modeled. The three main applications of deep learning in biology are disease prevention, the creation of biological predictive models, and specific treatments. All the related fields of research are analysis of protein structure and function prediction [12], gene and protein sequence, population genomics, regulatory and metabolic networks [13], biomedical image analysis [14], evolutionary genomics, proteomics [15], gene-disease associations and development, the spread of viruses, comparative genomics, taxonomic trees, population genetics, and systems biology.

2.2 Applications of Deep Learning

Technological progressions in the field of deep learning have led to groundbreaking innovations in genomics. However, the applications of deep learning in computational biology are much more than these two. Deep learning can help computational biologists to take advantage of very large data sets to find a hidden structure inside them. We will now discuss the applications of deep learning in this domain and how biological insights are derived using different methods.

The ability to obtain accurate predictive models without making strong hypotheses about the underlying biological mechanisms makes it one of the most popular choices amongst biologists. The majority of such applications can be approached along a simple Machine Learning pipeline. These steps are as follows.

2.2.1 Data Gathering and Cleaning

In order to collect data, the most popular choice is NCBI Computational Biology Branch online data bank, amongst the upcoming researchers. But we must understand

the data sets available are not specific and incorporates all the features related to a genome. We cannot use the entire dataset in our ML model as that will contain useful as well as unnecessary data. So, if we'll feed noise into our model then the output is not going to be as efficient as we expected. In such cases, data cleaning plays the most important role.

2.2.2 Data Preprocessing

It is the process of transforming or encoding the data so that it is easy for the model to parse it. Since a dataset with well-defined features and labels result in better performance of the model, more effort should be put into collecting the data and performing labeling, cleaning, and normalization techniques [16]. We need to make sure that the training data is enough for our model to avoid overfitting. Therefore, a general rule to follow is that the number of training samples should be at least as large as the number of model parameters [17].

In the case of image analysis, we will be coming across huge datasets, but we need to manually label them otherwise it would be difficult for us to train the Neural Nets. To maintain nonlinearity, we add images that are augmented by scaling, rotating, cropping images into the training dataset [18]. An alternative approach is to use a pre-trained network on large data for the image processing (e.g. GoogleNet [19], AlexNet [18], VGG [20], and ResNet [21]) and set the appropriate parameters.

Data Normalization can help fasten the training process with the help of appropriate choices. Categorical characteristics found in the DNA sequence must be categorized. They can be represented as binary vectors using one-hot encoding.

Example: A = [1 0 0 0], G = [0 1 0 0], C = [0 0 1 0], and T = [0 0 0 1]. Thus, DNA nucleotides can be represented as a long sequence of these strings.

In a CNN, the four bits of each encoded base are commonly considered analogously to color channels of an image to preserve the entity of a nucleotide [17].

2.2.3 Train and Test Set Split

Our Deep learning models need to be split into the trained, test, and validation sets on the dataset to avoid overfitting and assure that our model will generalize to the unseen datasets. Our training data set is used to learn from the data set with different selected parameters, which is then evaluated on the validation set. Most divisions in the datasets are 60% for training, 10% for validation, and the remainder for testing. If we are dealing with a small dataset, we may use techniques such as k-fold cross-validation or bootstrapping in certain cases [22].

2.2.4 Model Building

After we have the data prepared, it's time for us to choose which model architecture we need to use. The default architecture is a feed-forward neural network with fully connected hidden layers, starting and an endpoint [17]. If we are dealing with high-dimensional data, then convolutional neural networks are appropriate for those processes. In the case of long sequences like the DNA or protein sequence, or in the case of variable lengths (long-range dependencies), then a Recurrent Neural Network would be the most appropriate [23]. Most models are built by combining different architecture. Most deep learning frames offer different modules for a different architecture.

2.2.5 Training the Model

Decide appropriate algorithms for our model so that it can be trained. Then the produced results are compared with the expected results. Our main goal for the model training is to find the selective parameters which are going to be optimizing the fit function. The most commonly used objective functions are cross-entropy for classification and MSE for regression [17]. The stochastic gradient is used a lot for training the deep learning models.

2.2.6 Model Evaluation

Evaluate our predictive model from specific measurements which may be leveraged later to improve the accuracy of our model. It is recommended to monitor the accuracy of the training and validation set, not just the accuracy of the training. Here, we can observe that if our model is overfitted, the accuracy of the validation set will decrease in comparison with the performance of the instruction [16].

2.3 Novel Applications in Convolutional Neural Networks

- **DeepBind**: The DeepBind approach leverages the convolutional neural network to predict the sequential specificities of DNA and RNA binding proteins [24]. It contributes to the development of a variety of biological models and the identification of disease mutations. This deep learning model is by far the most scalable and unified than the existing research. This unified computational approach to pattern recognition has made it possible to recover functional single nucleotide variants (SNVs) and sequential specificity variations in diseases [25]. This 1-Dimensional CNN model is trained in a way that it can work with raw DNA/RNA as input

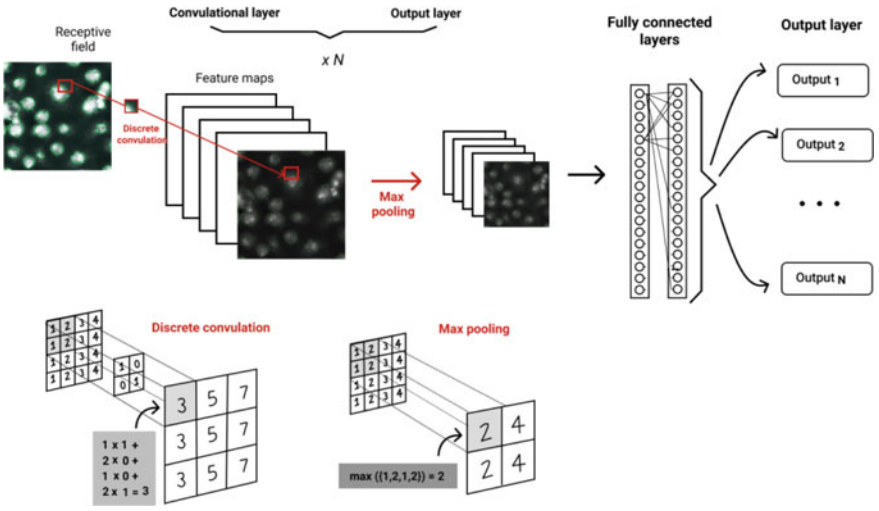

Fig. 1 Convolutional neural network for biological computations

samples. Using the perceptron algorithm [26], the neurons used in the convolutional layer analyze the pattern and combinations similar to conventional positional weight matrices. The learning function from deeper layers passes information to the convolutional layer about which motifs (Fig.1). The model then decides which motifs are more important. The relevant motifs can later be visualized in the form of heatmaps.

- **Predict mutation effects for in silico**: We can train our DNA/RNA sequence on the deep neural networks which can help predict the mutation effects in silico. This research allowed us to assess the effects of the sequence modifications. These methods are based on QTL mapping which can shed light on hidden information about rare SNV [24]. It is an effective approach for finding information about rare single nucleotide variations (SNVs). To better analyze these predictions associated with changes, we use mutation maps. Mutation mapping analysis can be performed by creating a pipeline to the raw RNA sequence giving us a matrix view of the input DNA/RNA sequence. The researchers were able to identify SNV using a deep neural network with predicted scores for both wild and mutant sequences [24]. MMAPPR (mutation mapping analysis pipeline for pooled RNA-seq) can even handle highly noisy data and calculates allelic frequency by Euclidean distance to identify the region of mutations [27]. We talked about how convolutional neural networks can help us in a number of ways in the area of regulatory genomics. Another major application of CNN architecture is to predict chromatin marks from a raw DNA sequence. The study suggests that input sequence is a major determinant of model performance, where the convolutional layers are connected with main windows to capture the features at different genomic length scales [28].

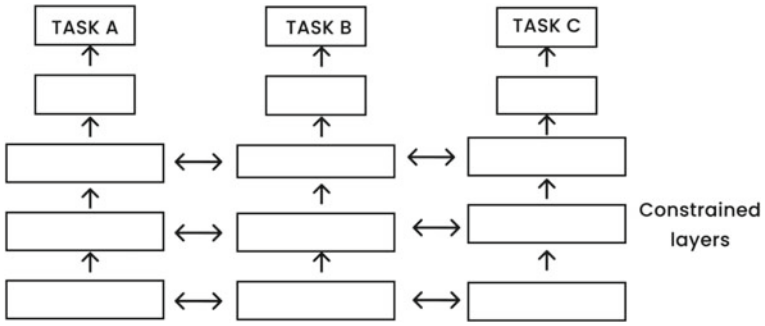

Fig. 2 The basic architecture of multi-task neural network

- **Multitask Neural Networks (Fig. 2)**: A similar but unique approach consisted of using a neural architecture with multiple output variables to simultaneously predict different chromatin states. The network tends to learn from the shared features between the outputs, thus allowing the model to scale the performance [21]. This drastically reduces the model training cost as compared to the traditional learning methods of neural networks. This is a huge application in the field of drug discovery, where hundreds of active compounds should be predicted, but in this architecture, there is a continuous increase in accuracy with the number of tasks.
- **Basset**: It's a wonderful open-source framework that is used to predict hypersensitivity across several types of cells to quantify the effect of SNVs on chromatin accessibility [29]. This model is trained on various genomic sites DNase-Seq and shows much greater accuracy than previous methods. It is a very powerful tool for the calculation and interpretation of the non-coding genome. Similar architecture to this was used to predict DNA methylation states in single-cell bisulfite sequencing. This CNN-based approach detects DNA sequence pattern information to account for the methylation context [30]. Subsequently, researchers applied different techniques to reduce noise genome-wide chromatin immunoprecipitation followed by sequencing data to get more accurate results about the chromatin marks.

Currently, CNNs are most widely used in the field of computational biology from fixed DNA sequences. An improved version of this architecture is Recurrent Neural networks (RNN) (Fig. 3) are better for sequential data [31]. RNNs have a lot of applications in regulatory genomics because you can vary the length in the model [32]. This enables us to capture long-range interactions in the sequence and outputs [33]. CNNs are much easier and scalable to train thus preferred over RNNs for the following reasons:

– They're not good enough in the training method as compared to CNN. The training process is complex and time-consuming. The problem arises when the various activation functions available like Sigmoid, Tanh, ReLU are applied in sequential training, the weights make the training process difficult.

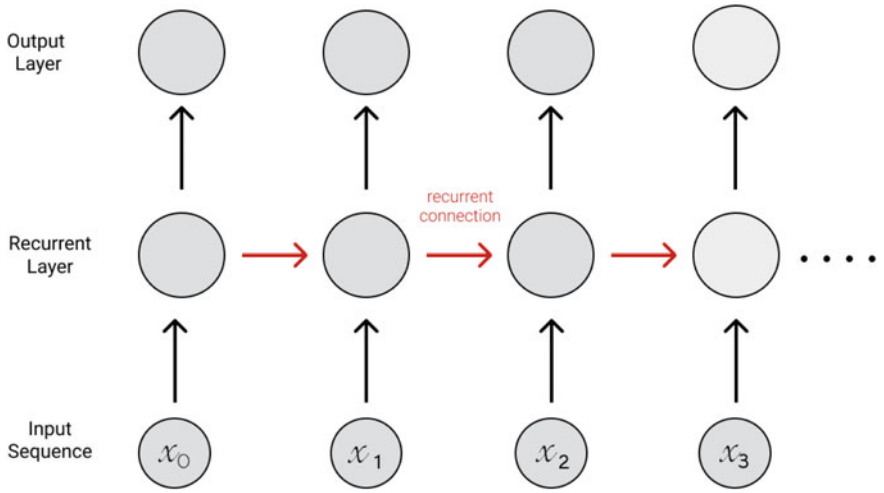

Fig. 3 The basic structure of recurrent neural network (RNN)

– It's hard to deal with long DNA sequences while using RNN architecture, especially using the Tanh and ReLU activations, which is another reason for the introduction of the Multi-Task Neural Network.
– There is another problem with the vanishing gradient descent. It is very difficult to catch long-haul conditions caused by the multiplicative angle that can be dramatically decreasing or expanding.

- **Deep belief Network (DBN):** Deep belief is a generative graphical model which is composed of multiple layers of Restricted Boltzmann Machines (RBM) or autoencoders that are stacked on top of each other. Each hidden layer in the network serves as a visible layer for the next coming layer, while the lowest visible layer is called a training set. The Deep belief network uses the unsupervised greedy approach to train the network layer-wise. The unsupervised greedy approach helps in initializing the weights. After the network is trained we can adopt different methods like Backpropagation to fine-tune the hyperparameters. We can also utilize the wake-sleep algorithm for the same. The hyperparameter tuning from the backpropagation algorithm may encounter some problems like:

  – The data inputs for the model should be properly labeled.
  – The learning rate for backpropagation and the wake-sleep algorithm is very low.

Researchers have suggested that DBN has better discriminative results than deep learning and holds promise in medical imaging diagnosis. Recent applications based on DBN include the classification of schizophrenic patients according to brain MRI. DBN is used for conducting quantitative structure–activity relationships (QSAR study) in drug design based on high-throughput screening. We can conclude from

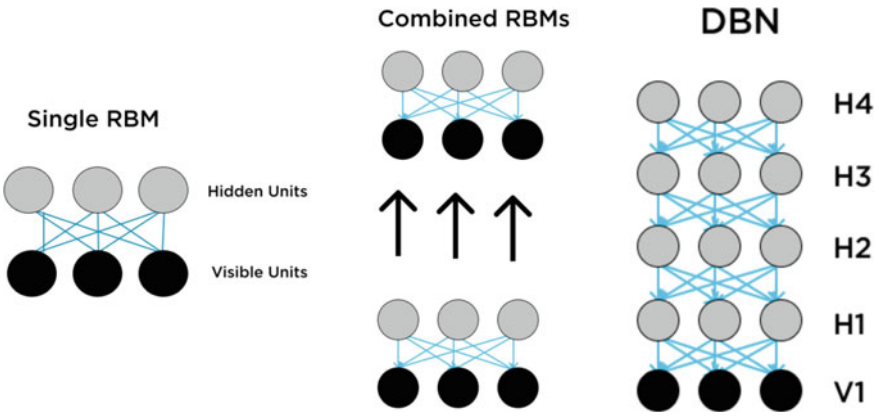

Fig. 4 Illustrative network structures of RBM and DBN

the results that optimization in weights or parameter initialization highly improves the deep neural network. The improvements in the weights help us to get better predictions (Fig. 4).

2.4 Deep Learning for Biological Imaging

Deep neural networks were the most important breakthrough in the field of analyzing medical and biological images. These deep neural networks are trained on huge datasets for e.g. Millions of images to successfully detect and categorize differences in the images [34]. These trained models can give a greater accuracy than a normal human who specializes in the field. Some of the most important searches associated with image analysis are performed for object detection models, image enhancement and retrieval, and semantic segmentation. The common term used for these applications in the domain of disease processing is Computer-aided diagnosis (CAD) [35]. Modeling of diseases using neural networks such as breast cancer, malaria, pneumonia, tuberculosis, Sars-Cov-2, and Alzheimer's disease has made it easier to classify these diseases.

The most common architectural network used for image classification is the convolutional neural network. CNN works better due to its great scalability and integration with computer vision and they support GPU parallelization. Convolutions refer to how the given input image is acquired, and calculate the matching strength for each position [36]. To compute the maximum amount of model matching in small patches, we use pooling (max-pooling) to determine the same thing. CNN has facilitated the classification process as it can easily detect slight differences in inputs. For example, CNNs can help classify an X-ray with an abnormal tissue or cell when it's trained on a large dataset. CNN greatly facilitates the extraction of relevant data from input

images and the automation of the classification process. Some of the examples of best researches are Example applications of CNNs in CAD include the classification of lung diseases based on computed tomography images, classification of breast cancer based on mammography images, the classification of tuberculosis based on X-ray images, and the detection of hemorrhages in color fundus images.

Several researchers also suggested a CNN-based architecture with a hybrid approach [37]. An algorithm can be used to encode the model parameters, thereby facilitating the LV segmentation process from cardiac MRI [35]. CNNs may be used to detect LVs, while autoencoders are used to analyze the image shape. A new approach introduced a design in which a wireless capsule endoscopy classification system was based on a hybrid CNN with Extreme Learning Machine (ELM).

Early approaches in deep learning uses for medical and biological imaging were more focused on pixel-level scale tasks, where the model was scaled using the additional models which were built over the output of the network. We can see an example of the same where a convolutional neural network was applied to predict and classify abnormal development in elegans embryo images. During this research, the model was formed on a CNN with a 40×40-pixel patch in order to classify the central pixel present at the cell wall, cytoplasm, membrane, and outside membrane [38]. The model predictions were then passed into a model for further analysis. These CNNs have completely outperformed all other standard models, for example, Markov random fields and conditional random fields in raw analysis tasks which are employed for the retrieval of images [20]. Image recovery has enabled us to produce higher-quality images from distorted images. The addition of layers to the CNN clears the noise in the input that is transmitted to the model. This process is followed by the addition of 4–5 convolutional layers and pooling layers, then two connected layers. This model exceeded all other submissions in the mitosis detection challenge at the 2012 International Profiling Recognition Conference.

Research suggests a deep CNN architecture of up to 22 layers that are used with a limited dataset of training [39]. Although CNNs have proven to be truly effective in comparison with other deep neural networks, there are some drawbacks and limitations underlying this [40]. CNN is designed for two-dimensional images, while the segmentation issue we face in the MR and CT process is primarily 3D. We can approach this problem by creating isometric images by interpolating the dataset images, but this will produce several blurred images. Another solution is to form the CNN on orthogonal patches to get the different views (axial, sagittal, and coronal views) [30]. This approach will drastically reduce the time complexity and make the algorithm feasible.

Deep learning architecture also benefits low-level image processing processes such as image segmentation. The research proposed a multifaceted 3D image learning approach based on DBN. It is not necessary to store collector space locally, making it a new approach. This deep learning approach is superior to any other low contrast model [41]. In [42] a pipeline has been suggested to perform object detection and segmentation as part of automated volumetric image processing.

For accurate segmentation of retinal vessels in Diabetic Retinopathy (DR) detection, this novel research is based on a Deep Convolutional Neural Network (DCNN)

named D-net [43]. This architecture solves the problem of inefficient information capturing that arises when we use traditional methods based on DCNN due to their small receptive fields. This problem leads to segmented retina vessels with more noise and significantly low classification accuracy. This research focuses on dilation convolution that is used in the network to obtain larger receptive fields. Results suggest that D-Net reduced the loss of feature information and made segmentation of tiny thin vessels possible.

3 CNN Model to Analyze Gene Expressions Images

3.1 Gene Expressions

Gene expressions are the process of converting genes or instructions in the DNA to a usable object. These objects can be proteins or RNA if the instruction is non-protein coding. Typically, the analysis of these expressions is done to check the functionality of a particular gene. For example, a gene can be responsible for a particular function in the body and assuming the motive of the analysis is to find the gene(s) responsible for a particular functionality. The functionality of a gene is known through alteration of the gene expression. The gene expressions are altered which as a result leads to a different protein build. The functionality to be tested is firstly conducted on untouched gene expression and the results are recorded. These results are then compared with the product built by the altered gene expression. If the expected functionality doesn't occur with the altered sequence it is concluded that the changed part of the expression is responsible for the functionality.

3.2 Convolution Neural Networks (CNN)

CNN is a type of neural network which usually works on 2-Dimensional data and is particularly applied to image data. Image data tends to be a 2-Dimensional array with multiple color channels, prominently Red, Green, and Blue. The concept for forward propagation used is convolution wherein the image is passed through a filter and it convolves or basically moves throughout the image to capture features of the image. These features are then updated during back-propagation. The size of the filter, the movement factor or stride and the amount of padding applied to the image can all be defined by the practitioner.

3.3 Why Deep Learning on Gene Expression Images

The boom of DNA microarray has taken molecular level research to greater heights and with this data being easily accessible on the internet and the need to solve new problems in a cost-effective and time-constrained manner deep learning brings about accurate predictions. Traditional machine learning approaches caused the problems of feature engineering and feature selection and deep learning models capture features during the training period therefore making it relatively easy to work with. The rise of computational power makes deep learning accessible to naïve users as well, therefore, providing a large community to develop solutions. Most importantly the need to do testing in particular conditions for all situations makes it redundant, all one needs is a computer and data to work on and the process of modeling can be done from remote.

3.4 Data Source and Preprocessing

Throughout this subchapter the limelight would be mainly on cancer-related research, the reason being the prominence of the study on the subject. Most of the data used in studies are found on The Cancer Genomic Atlas or commonly known as TCGA. TCGA is the repository of 10,340 samples for 33 cancer-type genomic data and 713 samples for 23 normal tissues. The motive of this study is to detect the cancer type based on the gene expression sample.

Problems faced and the Preprocessing methods applied:

- The data has a high range for normalized read count for each gene and this is troublesome in the process of modeling hindering the output of the model. This phenomenon occurs primarily due to fact that the weights adjusted by the model tends to be high and at times leading to exploding gradients.
- The data is scaled using log transform which scales the value in the range 0 and 1 making the learning process avoid the problem described above.
- CNNs as specified before work best with 2-Dimensional data and the current data is 1-Dimensional Data.
- The images are converted to 2D images by taking the square root of the number of data points present, the ceil of the value obtained is kept in variable and converted to a matrix. Some values go null and these values are padded with 0.
- Adding this zero hurts part of the scaling done earlier and hence the images are normalized keeping the range of values uniform.

The data attained is RNA Sequenced data or commonly known as RNA-Seq. The process starts by converting RNA into cDNA. This cDNA is passed through multiple processes at the end it becomes fragmented and read. This data is then processed again and can be read by researchers who make their inferences. These processes are

under constant development to gain the best possible data and some semi-automated tools use PCA or Principal Component Analysis to get at this data.

3.5 Models [44]

The networks used throughout this section are shallow and this is done to avoid overfitting. Deeper models do not favor biological data in particular and this is due to the volume of data present. Deep learning models normally tend to need millions of data points and training of perhaps hundreds of thousands or even millions of parameters. When the existence of parameters exceeds the data volume, overfitting will be a big concern as predictions on unseen data would not meet up with expectations. The activation function used through fully connected layers is ReLU.

3.5.1 CNN with 1-Dimensional Input

As the name suggests this model takes a 1-Dimensional input therefore the preprocessing step where the vector was converted to the image would not be needed here. The network used here is a simple 1-Dimensional CNN layer. The output from this layer is passed to a max-pooling layer which reduces the size of the kernel by half. This output is then passed onto a fully connected layer which has around 128 nodes which eventually passes the result to a softmax layer with 33 cells. Remember that in the data section it was discussed that the dataset contains 33 classes of cancer. As the filter passes through one dimension which is along with the gene expression, the features act as a good pattern recognizer. Typically, 2-Dimensional CNN models capture edges and eventually patterns however in this case patterns found are directly on the expression thus can give outputs that are much more accurate. Predicting cancer types would not only be computationally inexpensive but accurate as well.

3.5.2 CNN with 2-Dimensional Input

This model takes the input of the image formed during the preprocessing phase. Because of the nature of the input data, the model can store filters that will convolve the image during the period of forward propagation. These filters typically contain edges and lines in the first phase of the network and by the end of the network, the filter takes the shape of recognizable shapes and patterns. Similar to a normal CNN architecture, inputs are passed to a 2-Dimensional CNN which is then passed to a Max pooling layer reducing the size of the kernel. This output is then passed to a fully connected layer where it is flattened to convert the dimensions to 1 and finally a softmax layer which produces the resulting probability for all classes. One thing to note is that the Linear layers, softmax, and fully connected remain the

same for all models discussed in this section. One thing to note is the number of convolution layers. Although the decision to take fewer layers prevents overfitting could potentially harm the process of capturing filters. Some of these filters may take shape when training is done through many epochs which can lead to overfitting as well. However, the prediction would still be accurate considering the volume of data.

3.5.3 2-D Input with 1-D CNN

This architecture takes aspects from the models discussed above. 2-D inputs are taken in the input layer however, rather than passing it to a 2-Dimensional Convolution Layer the input is passed to 2 1-Dimensional Layers. To emphasize, one of the CNN takes care of convolution operations across rows and the other takes care of the column. Both the layers have separate Pooling layers, where both correspond to the particular orientation of the CNN layer. The pooled output is passed to common Linear Layers where the outputs are obtained. This particular architecture captures the most possible features as it keeps the features for the 2 axes separately hence giving an output that is to identify the cancer type much more accurately.

3.5.4 Layered CNN with 2-D Input

So far, all the models discussed are very shallow, this model is relatively deeper than the model discussed before. This model follows the pattern of CNN Layer followed by max-pooling and batch-normalization. In total as the title specifies the Network has 3 CNN layers where each of the layers has a different number of filters: the first layer contains 64 filters; the second layer contains 128 and the third one contains 256 filters. Before the output from the convolution layer can be flattened it is passed through a dropout layer with a dropout rate of 25%. Because the final layer of CNN has 256 filters the number of channels increases as well. As a result, the first Fully Connected Layer has 36,864 nodes followed by 1024 and 512. In the end, a softmax layer with 33 nodes completes the network by outputting the probability of the class. This architecture captures a feature gradually starting from edges to recognizable features making it more appealing for researchers. Potentially, researchers could look through the filters and make inferences regarding the process of identification.

3.5.5 Inferences from the Models

The predictions from the 1D-CNN model, in particular, were then assessed using a gene-effect score using the saliency map. The results from the map were interpreted and the main inference was that the model did not have enough confidence in predicting cancer types with a smaller number of marker genes (Marker genes are the genes that indicate a particular change) as a result and more genomic profiles

such as methylation would be required in order to differentiate cancer types having the same place of origin.

The 1-D CNN model predicted the training dataset with an accuracy of 0.9971 while the test accuracy of 0.9567 was attained. While the 2-D CNN model had a training accuracy of 0.9981 and test accuracy of 0.9957, the 2-D input with 1-D CNN scored a training accuracy of 0.996 and test accuracy of 0.9582. Finally, the 3 Layered CNN with 2-D input attained a training accuracy of 0.9689 and test accuracy of 0.9419.

Some of the key expressions include:

- The accuracy of the 1-D CNN was really high considering that it did not have the requirement to do much of the preprocessing and reshaping. Also, the number of parameters in the 2-D CNN is approximately 7 times more than the one in the 1-D CNN which is mighty considering the difference in the accuracy to be within 0.001. Fewer parameters translate to better performance as the training process in terms of time and computational power reduces substantially.
- The model 2-D input with 1-D CNN has performed the best on the test set and part of the reason could be the use of separate convolution layers for both row-wise and column helping it to capture more features and in prediction-based subjects the greater the number of features the better predictions are made by the model. However, the process of training becomes rigorous due to computation and time with respect to the 1-D CNN model.

3.6 Stack Autoencoders Along with CNN [45]

The discussion on this topic occurs on different datasets. While the research was conducted on tumor-based data, the motive of this study is to predict tumor types using gene expressions (Fig. 5).

Stacked Autoencoders is a multi-layer neural network that works on replicating the input as closely as possible and it has been made to use gene expression data to predict tumor type classification. The process of prediction firstly goes about by reducing the dimensionality of data using Principal Component Analysis (PCA). Due to the scarcity of tumor-related data similar unlabeled data are passed through the network for feature engineering. Secondly, other tumor gene expression data from the same platform are used as unlabeled data for feature learning since the number of samples for the specific tumor is really small. Thirdly, the weights of the features learned in the second step are tuned using the specific labeled data. The gene data is labeled and classified. This was the approach that was used early on however this study makes a change in this approach to avoid the problem of using the tumor data which is similar but not the data which is required for predicting for the particular region.

In this model, 1-D CNN is used as the main model for prediction. Additionally, encoders are used as a classifier and denoiser where corrupted data is converted to

cleaned and usable data. The whole process will be explained in the form of a pipeline fashion to make the flow and the concepts concise.

First, we consider the input layer. The input layer takes in 1-Dimensional gene vectors similar to the 1-D CNN model discussed before. This input layer passes the input to an unsupervised feature selection method called Infinite Feature Selection or shortly known as Inf-FS. Feature selection plays a crucial role in modeling as one feature may best represent the data in one set of features while may play a dormant role in another. What makes Inf-FS special is that it considers all subsets of features possible and passes it across to the next module in the pipeline eventually leading to the model. As some features are stronger when accompanied by some selected features, this feature selection algorithm makes sure the practitioner has the best set of features to work on.

Secondly, the data is normalized using the following strategy:

$$\hat{X} = (X - mean(X))\frac{std(\hat{X})}{std(X)} + mean(\hat{X})$$

Mean(X) is the mean of the data with respect to the row, $std(X)$ and $std(\widehat{X})$ are the standard deviations of the expected data or normalized data and standard deviation of data with respect to the row. Initially $std(\widehat{X})$ and $mean(\widehat{X})$ are 1 and 0 respectively.

Thirdly, we consider the expansion of the existing data. This step is similar to the concept of data augmentation however the process of expanding, in this case, is complex. In the first set of models, a lot of data was eliminated with the eliminated data not passing some of the metrics used. By augmenting data, the volume of data increases which reduces the chances of overfitting.

The method used to noise the data is known as Sample Expansion. In this method, genes are randomly selected from the data and the number of genes selected is less than or equal to the number of genes in the expression. These randomly chosen genes are blacked out or are set to 0. If the corruption of genes is not repetitive, the process of setting zero randomly is carried out for a total of $floor(\frac{m}{a})$ times. Here m corresponds to the number of genes in the dataset and a corresponds to a random number less than m. For each iteration, the modification made is stored so for every sample the floor value dirty samples are attained. Which is if the value after the floor operation is 5, one gets 5 dirty samples along with the original sample.

For N samples, this equation is multiplied by N so that the volume of data increases drastically. These new data are passed to an autoencoder which acts as a denoiser. This autoencoder has an input layer, hidden layer, and output layer. The hidden layer learns the weights and these weights act as features for the reconstruction of the expression.

Fourthly, now that the data has been augmented this data is passed to the 1-D CNN and the stacked encoder. As seen throughout this subchapter 1-D CNN captures features globally with ease. This 1-D CNN will be a deeper model than the ones seen before and this is due to the reason for more data. Because the data is 1-D the filters

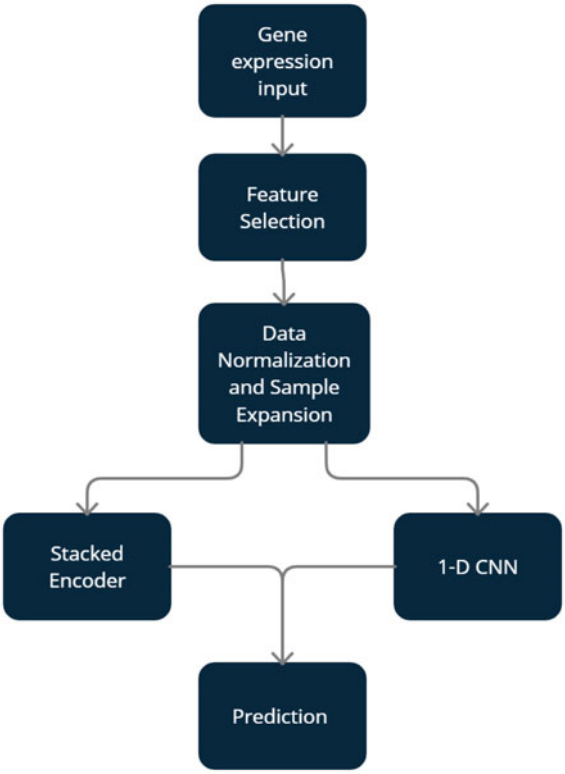

Fig. 5 Gene expression prediction model

used are 1-Dimensional as well. The structure of the network is as it goes, one input layer followed by 2 blocks of a Convolution layer followed by a Max pooling layer. The Max-Pooling Layer leads to a fully connected layer which then leads to the output layer. This prediction conveys the gene responsible for the cause of the tumor in a case.

3.7 Deep CNN with SVM

Typically for 2-dimensional image-based data, deep convolutional neural networks or DCNN have provided state-of-the-art results for image classification tasks. This research [46] focuses on using the Deep CNN model as a feature extractor from the data. Now, these features are used by a classification model like Support Vector Machine. The filter maps are passed through a flatten layer which is then passed to the SVM model. The SVM model was experimented with both RBF and Linear based kernels to learn the features.

For the dataset, the authors use 15 MGE datasets present in Array Express datasets. The feature array for training is conveyed using the microarray expression feature. Adam is used as an optimizer and cross-entropy as the choice of loss. On the basis of performance, it was concluded that Tanh activation provided the best result with respect to gene expression-based data.

For comparison, the authors used the following models: DCNN-SVM (Linear and RBF Kernel), DCNN, DCNN—Random Forest, SVM alone, and Random Forest. Results concluded that the DCNN-SVM hybrid model with RBF kernel had superior performance in most datasets.

3.8 Future Scope

Although gene expression is a topic that has been researched extensively, the main utilization of the topic in CNN is restricted to predicting the gene responsible for a particular tumor or cancer. The scope of CNN is yet to be exploited and this is possible when more data samples are available which enable researchers to use many complex architectures to predict problems that have had little human intervention with the use of transfer learning.

4 Conclusion

To conclude, this chapter portrays the applications of Deep Learning on various aspects of genomic study. The chapter covers various concepts of Computational Biology with respect to Deep Learning and discusses various applications of CNN-based architectures on gene expressions. The chapter aims at helping researchers design a suitable deep learning approach to extract knowledge from biological data, thus how to decipher and characterize data. Although the main focus of the chapter is CNN, the chapter also covers other standalone and CNN-ensemble architectures to provide the reader with reasoning as to "why CNN?". With the skyrocketing acceptance of this technology, this chapter is expected to provide a base intuition for practitioners and researchers.

References

1. Adate, A., Tripathy, B.K.: Deep learning techniques for image processing. In: Machine Learning for Big Data Analysis Berlin, pp. 69–90. De Gruyter, Boston (2018)
2. Bose, A., Tripathy, B.K.: Deep learning for audio signal classification. In: Deep Learning Research and Applications, pp. 105–136. De Gruyter Publications (2020)

3. Garg, N., Nikhitha, P., Tripathy, B.K.: Image retrieval using latent feature learning by deep architecture. In: Proceedings of IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–4. (2014)
4. Prakash, V., Tripathy, B.K.: Recent advancements in automatic sign language recognition (SLR). In: Computational Intelligence for Human Action Recognition, pp. 1–24. CRC Press (2020)
5. Singhania, U., Tripathy, B.K.: Text-based image retrieval using deep learning. In: Encyclopedia of Information Science and Technology, 5th edn., pp. 87–97. IGI Global, USA (2020)
6. Baktha, K., Tripathy, B.K.: Investigation of recurrent neural networks in the field of sentiment analysis. In: Proceedings of IEEE International Conference on Communication and Signal Processing, pp. 2047–2050. (2017)
7. Adate, A., Tripathy, B.K., Arya, D., Shaha, A.: Impact of deep neural learning on artificial intelligence research. In: Deep Learning Research and Applications, vol. 7, pp. 69–84. De Gruyter Publications (2020)
8. Adate, A., Tripathy, B.K.: S-lstm-gan: shared recurrent neural networks with adversarial training. In: Proceedings of the 2nd International Conference on Data Engineering and Communication Technology, pp. 107–115. Springer, Singapore (2019)
9. Adate, A., Tripathy, B.K. Understanding single image super resolution techniques with generative adversarial networks. In: Advances in Intelligent Systems and Computing, vol. 816, pp. 833–840. Springer, Singapore (2019)
10. Maheshwari, K., Shaha, A., Arya, D., Rajasekaran, R., Tripathy, B.K.: Convolutional neural networks: a bottom-up approach. Deep Learn. Res. Appl. **7**, 21–50 (2019)
11. Debgupta, R., Chaudhuri, B.B., Tripathy, B.K. A wide ResNet-based approach for age and gender estimation in face images. In: Proceedings of International Conference on Innovative Computing and Communications, pp. 517–530. Springer, Singapore (2020)
12. Libbrecht, M.W., Noble, W.S.: Machine learning applications in genetics and genomics. Nat. Rev. Genet. **16**, 321–332 (2015)
13. Kell, D.B.: Metabolomics, machine learning and modeling: towards an understanding of the language of cells. Biochem. Soc. Trans. **33**, 520–524 (2005)
14. Fritscher, K., Raudaschl, P., Zaffino, P., Spadea, M.F., Sharp, G.C., Schubert, R.: Deep neural networks for fast segmentation of 3D medical images. In: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 158–165. (2016)
15. Swan, A.L., Mobasheri, A., Allaway, D., Liddell, S., Bacardit, J.: Application of machine learning to proteomics data: classification and biomarker identification in postgenomics biology. OMICS **17**(12), 595–610 (2013)
16. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: Montavon, G., Orr, G., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade, pp. 437–478. Springer, Berlin Heidelberg (2012)
17. Angermueller, C., Pärnamaa, T., Parts, L., Stegle, O.: Deep learning for computational biology. Mol. Syst. Biol. **12**(7), 878 (2016). https://doi.org/10.15252/msb.20156651
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012)
19. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.2818–2826 (2016)
20. Li, S.Z.: Markov random field modeling in image analysis. Springer Science & Business Media, Berlin, Heidelberg (2009)
21. Dahl, G.E., Jaitly, N., Salakhutdinov, R.: Multi-task neural networks for QSAR predictions (2014). arXiv:1406.1231
22. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. Math. Intell. **27**, 83–85 (2005)
23. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3156–3164 (2015)

24. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J.: Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat. Biotechnol. **33**, 831–838 (2015)
25. Sønderby, S.K., Winther, O.: Protein secondary structure prediction with long short term memory networks (2014). arXiv:1412.7828
26. Stormo, G.D., Schneider, T.D., Gold, L., Ehrenfeucht, A.: Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli. Nucleic Acids Res. **10**(9), 2997–3011 (1982)
27. Hill, J.T., Demarest, B.L., Bisgrove, B.W., Gorsi, B., Su, Y.C., Yost, H.J.: MMAPPR: mutation mapping analysis pipeline for pooled RNA-seq. Genome Res. **23**(4), 687–697 (2013). https://doi.org/10.1101/gr.146936.112
28. Zhou, J., Troyanskaya, O.G.: Predicting effects of noncoding variants with deep learning-based sequence model. Nat. Methods **12**(10), 931–934 (2015)
29. Kelley, D.R., Snoek, J., Rinn, J.L.: Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. **26**(7), 990–999 (2016). https://doi.org/10.1101/gr.200535.115
30. Angermueller, C., Lee, H., Reik, W., Stegle, O.: Accurate prediction of single-cell DNA methylation states using deep learning. Genome Biol. **18**(1), 1–13 (2016). https://doi.org/10.1101/055715
31. Lipton, Z.C., Berkowitz, J., Elkan, C.: A critical review of recurrent neural networks for sequence learning (2015). arXiv:1506.00019
32. Agathocleous, M., Christodoulou, G., Promponas, V., Christodoulou, C., Vassiliades, V., Antoniou, A.: Protein secondary structure prediction with bidirectional recurrent neural nets: can weight updating for each residue enhance performance? In: Papadopoulos, H., Andreou, A.S., Bramer, M. (eds.) Artificial Intelligence Applications and Innovations, vol. 339, pp. 128–137. Springer, Berlin Heidelberg (2010)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv:1409.1556
34. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). arXiv:1512.03385
35. Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., Yang, G.Z.: Deep learning for health informatics. IEEE J. Biomed. Health Inform. **21**(1), 4–21 (2017). https://doi.org/10.1109/JBHI.2016.2636665
36. Greenspan, H., van Ginneken, B., Summers, R.M.: Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. IEEE Trans. Med. Imaging **35**(5), 1153–1159 (2016)
37. Avendi, M., Kheradvar, A., Jafarkhani, H.: A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI. Med. Image Anal. **30**, 108–119 (2016)
38. Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., Barbano, P.E.: Toward automatic phenotyping of developing embryos from videos. IEEE Trans. Image Process. **14**, 1360–1371 (2005)
39. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: CNN architectures dataset characteristics and transfer learning. IEEE Trans. Med. Imaging **35**(5), 1285–1298 (2016)
40. Havaei, M., Guizard, N., Larochelle, H., Jodoin, P.: Deep learning trends for focal brain pathology segmentation in MRI. In: Machine Learning for Health Informatics, pp. 125–148. Springer, Cham (2016). abs/1607.052582016
41. Mansoor, A., Cerrolaza, J.J., Idrees, R., Biggs, E., Alsharid, M.A., Avery, R.A., Linguraru, M.G.: Deep learning guided partitioned shape model for anterior visual pathway segmentation. IEEE Trans. Med. Imaging **35**(8), 1856–1865 (2016)
42. Ghesu, F.C., Krubasik, E., Georgescu, B., Singh, V., Zheng, Y., Hornegger, J., Comaniciu, D.: Marginal space deep learning: efficient architecture for volumetric image parsing. IEEE Trans. Med. Imaging **35**(5), 1217–1228 (2016)

43. Biswas, R., Vasan, A., Roy, S.S.: Dilated Deep Neural network for segmentation of retinal blood vessels in fundus images. Iran. J. Sci. Technol. Trans. Electr. Eng. 1–14 (2019)
44. Mostavi, M., Chiu, Y.C., Huang, Y., Chen, Y.: Convolutional neural network models for cancer type prediction based on gene expression. BMC Med. Genom. **13**, 1–13 (2020)
45. Liu, J., Wang, X., Cheng, Y., Zhang, L.: Tumor gene expression data classification via sample expansion-based deep learning. Oncotarget **8**(65), 109646 (2017)
46. Huynh, P.H., Nguyen, V.H., Do, T.N.: Novel hybrid DCNN–SVM model for classifying RNA-sequencing gene expression data. J. Inf. Telecommun. **3**(4), 533–547 (2019)

Leukaemia Classification Using Machine Learning and Genomics

Vinamra Khoria, Amit Kumar, and Sanjiban Shekhar Roy

Abstract The field of genomics is vast and innovation is happening at a rapid pace today. With the availability of lots of medical data and extensive research, the tools at our disposal are sharper than ever. One such tool that has quite a lot of untapped potential is Machine Learning. Machine Learning is the field of computer science that gives computers the ability to understand data and make decisions based on that understanding, in quite a similar way as we humans do. Machine learning has proven to be the next big thing in almost all industries today including medicine. The use of Machine Learning in the field of genomics however, is yet to reach its true momentum. With the help of machine learning, patterns in genetic data can be found that were unknown to us earlier and these patterns can be very useful in making conclusions about diseases and disorders that are inherently genetic in nature. In this work, we have classified the patients based on the two cancer classes, namely acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL).

Keywords Leukemia classification · KNN · Machine Learning · PCA

1 Introduction

Cancer treatment has been one of the most active areas of medical research for many decades now. One of the main challenges that cancer treatment poses today is targeting tumor specific therapy. This is essential to maximize the efficiency of treatment and to reduce the toxicity of treatment at the same time. Accurate cancer classification is thus central to advancing treatment today. Till date, classification

V. Khoria · A. Kumar · S. S. Roy (✉)
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: sanjibansroy@ieee.org

V. Khoria
e-mail: vinamra.khoria2019@vitstudent.ac.in

A. Kumar
e-mail: amit.kumar2019@vitstudent.ac.in

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_6

has mostly been done by observing the morphological appearance of tumors, but this approach is quite naive as different classes can often look similar, but react very differently to therapy. This calls for the need of a new approach for classifying cancer. This is where genomics and machine learning come into the scenario. Gene expression data using DNA microarrays has been suggested to be able to provide a tool for classifying cancer. This is what we have set out to do in this chapter. We will be using gene expression data to build a class predictor taking acute leukemias as our test case. Leukemia has mainly two classes—acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Leukemia classification is a lengthy process with steps involved such as—interpreting the tumor's morphology, histochemistry, immunophenotyping, and cytogenetic analysis. All of these steps have to be carried out in separate, highly specialized laboratories. And although the classification is mostly accurate, errors still happen. In this chapter, we shall be using machine learning on gene expression data to try to build a classifier that correctly classifies ALL from AML.

2 Background

During the initial state of the project, a literature survey was done to get an understanding of KNN and data preprocessing [1–5]. Various papers were reviewed to identify the performance factor to be analyzed. In the beginning, various different classification models were considered [6–13]. But KNN (K-Nearest Neighbors) has done the most needful job. The literature review helped in the understanding of the methods for measuring and comparing performances of different models. It also helped to understand the data preprocessing, different data exploration and better understanding of the problem statement. The deciding factor for the project has been accuracy of the model which as we will see is best achieved by KNN algorithm.

3 Proposed Model

Recently researchers have shown interest in developing machine learning algorithms for various disease detection [14–27].In below we have described the detail of K-NN model and shown the proposed algorithm as well.

3.1 K-Nearest Neighbors

K Nearest Neighbors or KNN is an algorithm which can be used both for classification and regression purposes, however it is more widely used for classification problems [28–30]. To explain how KNN works, let us take the example of our classification

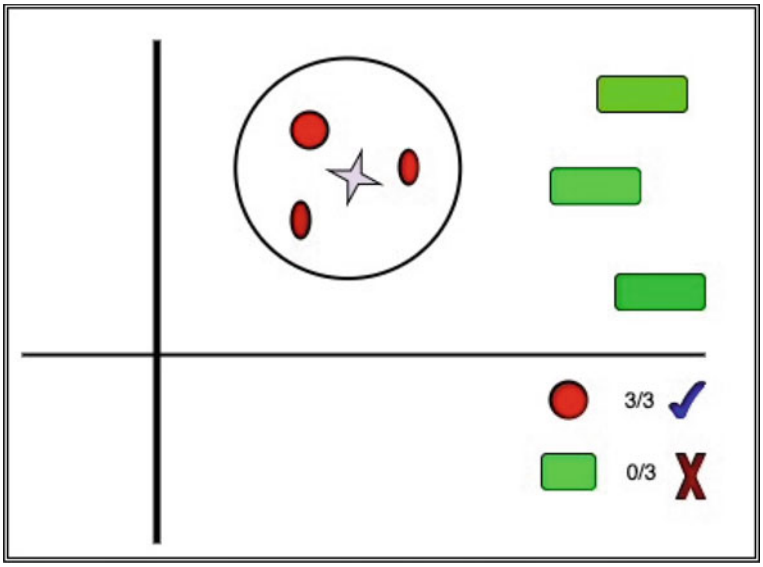

Fig. 1 KNN explained

problem. There are two classes in our example, one is represented by red circles and the other is by green squares. We have a point whose class is unknown which is represented by the blue star. To classify the unknown point correctly, the proposed KNN classifier takes a look at it's—'k' nearest neighbors. Let k be 3 here. The 3 nearest points to our blue star are the red circles. Thus we can classify the point which belongs to the class that is represented by the red circles. The more points of the same class that are included in the K nearest neighbors of the point we are analyzing, the more confidently we can classify our point (Fig. 1).

3.2 Algorithm

Let there be n number of training data samples and a be some random unknown point z.

1. *Initialize an array arr[] which stores training samples.*
2. *For i = 0 to n: Calculate Euclidean distance d(arr[i], z).*
3. *Find set St of K smallest distance obtained.*
4. *Return the majority label among St.*

Distance formula used is the Euclidean distance formula.

If there are (x_1, y_1) and (x_2, y_2) two distinct points, distance between them will be,

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

One of the advantages of K-NN is that it maintains a strong consistency. The Bayes error related to kNN can be represented as below [38].

$$R^* \leq R_{kNN} \leq R^*\left(2 - \frac{MR^*}{M-1}\right) \tag{2}$$

R^* is Bayes Error Rate, the rate of erron of kNN is R_{kNN} and the number classes are presented as M. If M equals two RB limits to zero, this limit reduces to "not more than twice the Bayesian error rate".

4 Experimental Results

4.1 Dataset

The dataset that we have used in the chapter comes from a proof-of-concept study published in 1999 by Golub et al. [31]. The data set is available in this link (https://www.kaggle.com/crawford/gene-expression). This data set represents gene expression monitoring via DNA microarrays. And it can be used for classification of new cases of cancer. Therefore, it can provide a general approach for identifying new cancer classes and assigning tumors to known classes. This data set is used for classification of leukemia patients and it contains two cancer classes namely acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The data is broadly divided into **three sets**- the class labeling for all the patients, the training data which consists of 38 samples and the testing data which consists of 34 independent samples. We will use the 38 training samples to train our proposed machine learning algorithm and the rest of the 34 samples to test its performance. Now let us look at the data more closely to understand what knowledge exactly will our classifier be learning and for that we have carried out data exploration, pre-processing; a comprehensive Principal Component Analysis (PCA) technique on the data set.

4.2 Data Exploration

First we look at the class labeling of our 72 patients which will be referred to by our classifier during the learning process (Table 1).

As we can see, a single row of the data contains a serial number, a patient ID, and the type of cancer the patient is suffering from. This is just a small part of the total 72 samples present in the data.

Table 1 Classification data of leukemia patients into two categories—ALL and AML

Serial number	Patient id	Type of cancer
0	1	ALL
1	2	ALL
2	3	AML
3	4	AML
4	5	ALL

Secondly, we look at the training data which contains the gene expression information for 38 patients from patient ID 1 to 38 (Table 2).

Each row represents a gene and it's expression in all the 38 patients. The first column contains the gene description. The second column contains a Gene Accession Number which is basically a unique identifier for the gene. The successive columns represent the patients with their information regarding this particular gene. Each patient is represented by two columns. The first column for a patient contains values of expression for the particular gene. For example—patient 1 has a value of 214 for the first gene—AFFX-BioB-5_at and patient 2 has a value of -139 for the same gene. The second column for each patient is the call column. The call columns are a decision on whether that gene is present in the sample in the preceding column. With patient 1 in the train set again, AFFX-BioB-5_at (index 0) is Absent, hum_alu_at (index 18) is Present, and D29642_at (index 341) is Marginal, which means it's too close to call. There are a total of 7129 rows which means that 7129 genes have been taken into consideration for all the patients. Our classifier will try to find patterns in this gene expression data by cross referencing the patients to the labelling data which we saw in the earlier section. These genes may be further feature engineered as we'll see ahead.

4.3 Data Preprocessing

The dataset does not have any null/empty values and hence no filling was required. The transpose of both the training and test data was done so that the rows represented a patient and the columns the gene expression values. In this way, all gene expressions could be used as features/components in our analysis. But since the number of genes we have is more than seven thousands, it would be a bad idea to train our model treating each gene as a separate feature. Hence, we will be applying Principal Component Analysis in the next step of our experiment. Since the data consists of a lot of Biological Science and scientific terms which can be confusing sometimes, also it has nothing to do with training the model, Biological names will be dropped for convenience. Also data enthusiasts reading this can get confused with confusing transcript numbers. So data will be converted into just two columns and the training and further analysis will be done.

Table 2 Training dataset containing gene expression data for 38 patients

	Gene Description	Gene Accession Number	1	Call	2	call.1	3	call.2	4	call.3	5	call.4	6	call.5	7	call.6
0	AFFX-BioB-5_at (endogenous control)	AFFX-BioB-5_at	−214	A	−139	A	76	A	−135	A	−106	A	−138	A	−72	A
1	AFFX-BioB-M_at (endogenous control)	AFFX-BioB-M_at	−153	A	−73	A	–49	A	−114	A	−125	A	−85	A	−144	A
2	AFFX-BioB_3_at (endogenous control)	AFFX-BioB3_at	−58	A	−1	A	307	A	−265	A	−76	A	215	A	238	A
3	AFFX-BioC_5_at (endogenus control)	AFFX-BioC-5_at	88	A	283	A	309	A	12	A	168	A	71	A	55	A

4.4 Principal Component Analysis (PCA)

We have used Principal Component Analysis (PCA) for emphasizing variation and identifying healthy patterns in our dataset [32]. PCA is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. In our case, the number of features/genes being more than seven thousand makes it really unfeasible to train a model taking into consideration all these features. To deal with this problem, we will use PCA to pack most of our features into tight components which will retain most of the information of our dataset and also discard the features which don't show much correlation. How PCA works is that the most important features are packed in the components at the beginning and the importance of features start decreasing as we move to the second, third and so on components. The new components may not be in a form with any actual meaning/representation but they help in the training and visualisation part of the process. In our case, we will compress our initial 7129 components into 30 Principal Components with the help of PCA and then use KNN to finally build our model and train on these 30 components. To show a glimpse of what PCA actually does, we applied PCA to our data and reduced it to 3 components (Fig. 2).

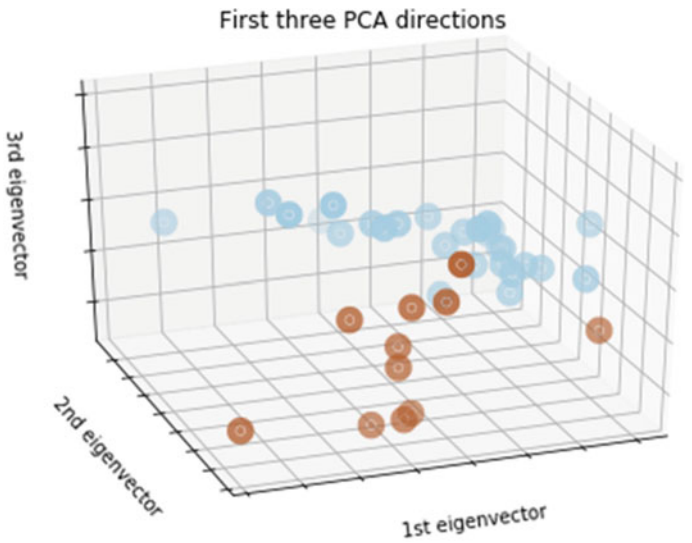

Fig. 2 A three dimension breakdown of our dataset with 7129 dimensionality using PCA

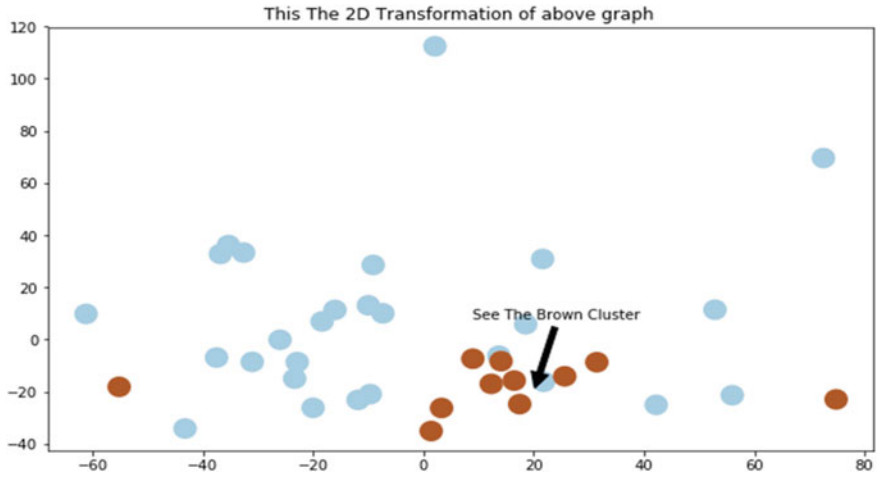

Fig. 3 2D representation of the clusters

As we can see, the two clusters—the brown and the blue clusters representing our two cancer classes (ALL and AML) can be very clearly observed when the dimensionality is reduced using PCA (Fig. 3).

4.5 Model Building

After we reduce our dataset to 30 principal components, the data can be visualised like this (Fig. 4).

It is clear from the above histogram that KNN would be a useful approach in our experimentation. The reduction to 3 components did a better job at proving why we used PCA. If you refer to the images again, you will see that the similar types of cancer tend to cluster together in the 3-dimensional space. And since reducing 7129 initial components to 3 Principal Components would lead to a lot of data loss, we set the number of final principal components as 30. The reason is, 30 components retain about 95% of the information from our original dataset and these many features can be easily learned by our algorithm. Hence the Principal Component Analysis with the standardisation of our data using a standard scaler completes our data preprocessing portion and we can finally move on to our model training section (Fig. 5).

4.6 Model Training

After fitting our 38 training samples to our KNN classifier, we tested the model on the 34 test samples and the following confusion matrix was obtained (Fig. 6).

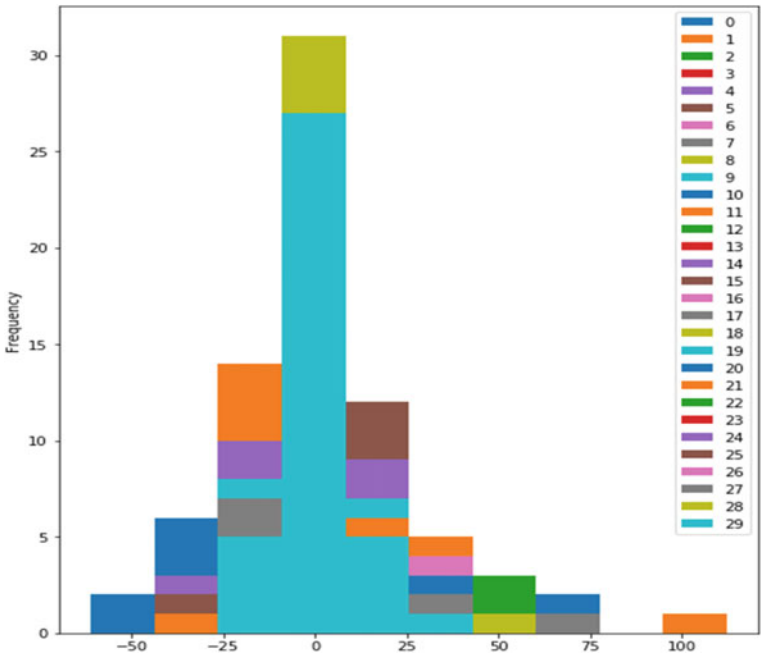

Fig. 4 Histogram representing the 30 principal components after PCA analysis on data

Let us break down the above matrix to get insights on our model.

5 Model Analysis

Let us analyze the metrics of our model by breaking down the above confusion matrix. The ALL and AML leukemia classes are represented by the 0 and 1 index on the confusion matrix respectively. As we can see, out of the 20 ALL samples, 16 were identified correctly by our model which is an 80 percent accuracy when it comes to classifying ALL samples. The rest of the 4 ALL samples were identified incorrectly as AML samples. When it comes to the AML classification, we see that the results are a little less satisfactory. Out of the 14 AML samples, 13 were classified as ALL samples which means that the model has a lot of trouble differentiating AML from ALL than it has in differentiating ALL from AML (Fig. 7).

There could be several reasons for this,

- The training data has more samples corresponding to the ALL class
- May have lead to the model overfitting towards the ALL class
- Poor correlation among principal components
- AML genes not showing any prominent patterns

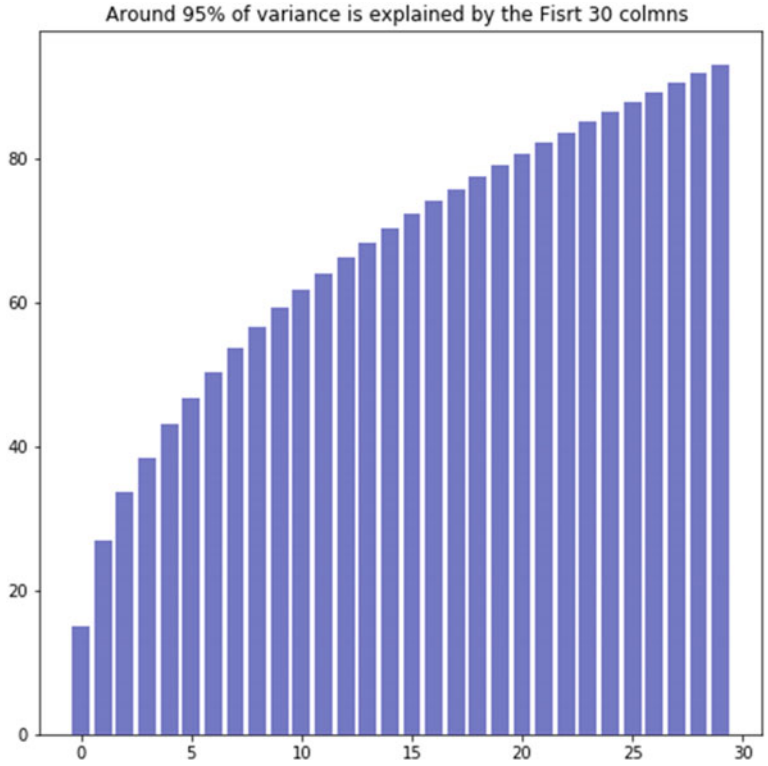

Fig. 5 Only 5% data is lost after reducing 7129 components to 30 components

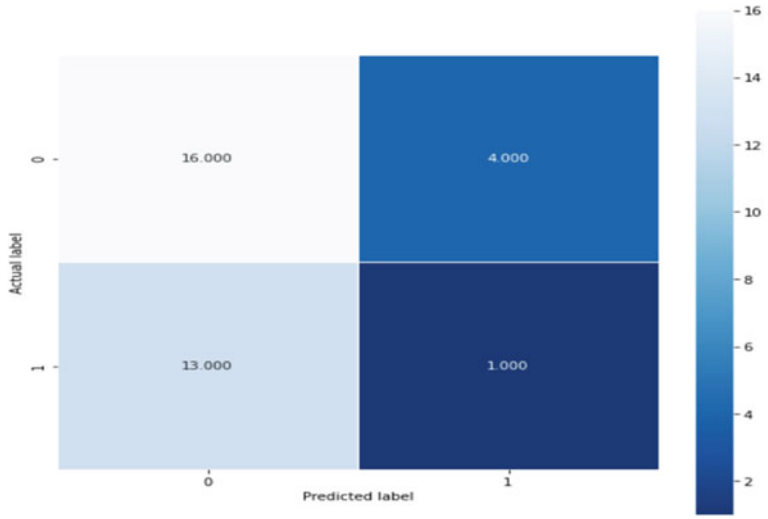

Fig. 6 Confusion matrix of test results

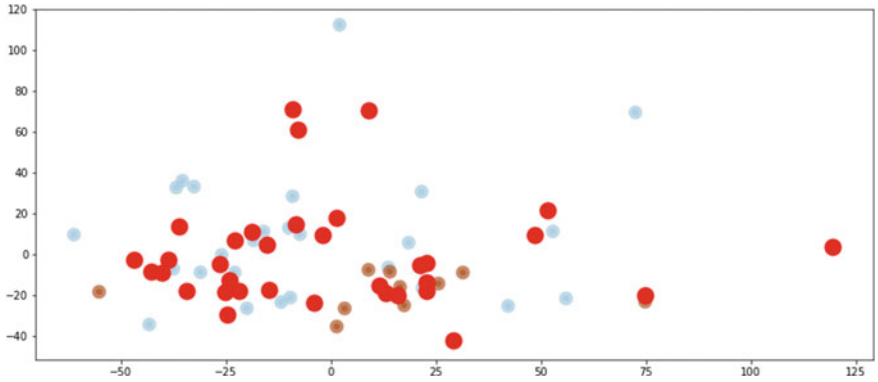

Fig. 7 Plotting the predicted test points vs their actual labels

- Insufficient training data (34 samples only).

Here the red points are plotted on the training set to show which points are falsely represented in which category. It can also be thought of as representing the points which were not on the diagonal of the above plotted confusion matrix.

6 Conclusion

The objective of our work was to find a solution to accurately classify the two acute leukemia types—ALL and AML. We started with a literature survey of other works in this field and decided to build a model of our own. The first step was the data collection part of the process. After the data has been collected, we started with pre-processing which is common to all machine learning systems. The distinctive step of our preprocessing was Principal Component Analysis which allowed us to compress a lot of information into relatively less numerous variables and thus eliminate redundant data. After that, we built a machine learning system using KNN and trained our model on that data repetitively. The result was a model which was able to classify the two types of cancer with some level of accuracy. Although we were partially successful in our efforts, the work can be expanded in future by making available more data. To improve the results and accuracy, data can be trained with different model such as deep learning; as deep learning might improve the accuracy. This proposed system if implemented as a large scale software, can at one point, even be used by professionals in the medical field to actually classify leukemia instead of doing it manually and that is the power of machine learning that we set out to harness in this proposed work.

References

1. Lee, S.I., Celik, S., Logsdon, B.A., Lundberg, S.M., Martins, T.J., Oehler, V.G., ... Becker, P.S.: A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. Nat. Commun. 9(1), 1–13 (2018)
2. Salah, H.T., Muhsen, I.N., Salama, M.E., Owaidah, T., Hashmi, S.K.: Machine learning applications in the diagnosis of leukemia: current trends and future directions. Int. J. Lab. Hematol. **41**(6), 717–725 (2019)
3. Abbas, N., Mohamad, D.: Automatic color nuclei segmentation of leukocytes for acute leukemia. Res. J. Appl. Sci. Eng. Technol. **7**(14), 2987–2993 (2014)
4. Abbas, N., Mohamad, D., Abdullah, A.H., Saba, T., Al-Rodhaan, M., Al-Dhelaan, A.: Nuclei segmentation leukocytes in blood smear digital images. Pak. J. Pharm. Sci. **28**(5), 1801–1806 (2015)
5. Jagadev, P., Virani, H.G.: Detection of leukemia and its types using image processing and machine learning. In: 2017 International Conference on Trends in Electronics and Informatics (ICEI), pp. 522–526. IEEE (2017)
6. Fahad, H.M., Ghani Khan, M.U., Saba, T., Rehman, A., Iqbal, S.: Microscopic abnormality classification of cardiac murmurs using ANFIS and HMM. Microsc. Res. Tech. **81**(5), 449–457 (2018). https://doi.org/10.1002/jemt.22998
7. Goutam, D., Sailaja, S.: Classification of acute myelogenous leukemia in blood microscopic images using supervised classifiers. Int. J. Eng. Res. Technol. (IJERT) **4**(1), 569–574 (2015)
8. Mitra, S., Roy, S.S., Srinivasan, K.: Classifying CT scan images based on contrast material and age of a person: ConvNets approach. In: Data Analytics in Biomedical Engineering and Healthcare, pp. 105–118. Academic Press (2020)
9. Wadhwa, A., Roy, S.S.: Driver drowsiness detection using heart rate and behavior methods: a study. Data Anal Biomed Eng Healthcare 163–177 (2020)
10. Deo, R., Samui, P., Roy, S.S. (eds.).: Predictive Modelling for Energy Management and Power Systems Engineering. Elsevier (2020)
11. Roy, S.S., Rodrigues, N., Taguchi, Y.: Incremental dilations using CNN for brain tumor classification. Appl. Sci. **10**(14), 4915 (2020)
12. Warnat-Herresthal, S., Perrakis, K., Taschler, B., Becker, M., Baßler, K., Beyer, M., ... Schultze, J.L.: Scalable prediction of acute myeloid leukemia using high-dimensional machine learning and blood transcriptomics. Iscience **23**(1), 100780 (2020)
13. Fathi, E., Rezaee, M.J., Tavakkoli-Moghaddam, R., Alizadeh, A., Montazer, A.: Design of an integrated model for diagnosis and classification of pediatric acute leukemia using machine learning. Proc. Inst. Mech. Eng. Part [H]: J. Eng. Med. **234**(10), 1051–1069 (2020)
14. Roy, S.S., Paraschiv, N., Popa, M., Lile, R., Naktode, I.: Prediction of air-pollutant concentrations using hybrid model of regression and genetic algorithm. J. Intell. Fuzzy Syst. (Preprint), 1–11 (2020)
15. Roy, S.S., Chopra, R., Lee, K.C., Spampinato, C., Mohammadi-ivatlood, B.: Random forest, gradient boosted machines and deep neural network for stock price forecasting: a comparative analysis on South Korean companies. Int. J. Ad Hoc Ubiquitous Comput. **33**(1), 62–71 (2020)
16. Roy, S.S., Sikaria, R., Susan, A.: A deep learning based CNN approach on MRI for Alzheimer's disease detection. Intell. Dec. Technol. (Preprint), 1–11 (2019)
17. Biswas, R., Vasan, A., Roy, S.S.: Dilated deep neural network for segmentation of retinal blood vessels in fundus images. Iranian J. Sci. Technol. Trans. Electr. Eng. **44**(1), 505–518 (2020)
18. Roy, S.S., Samui, P., Nagtode, I., Jain, H., Shivaramakrishnan, V., Mohammadi-Ivatloo, B.: Forecasting heating and cooling loads of buildings: a comparative performance analysis. J. Ambient. Intell. Humaniz. Comput. **11**(3), 1253–1264 (2020)
19. Balas, V.E., Roy, S.S., Sharma, D., Samui, P. (eds.): Handbook of Deep Learning Applications, vol. 136. Springer, New York (2019)
20. Bose, A., Roy, S.S., Balas, V.E., Samui, P.: Deep learning for brain computer interfaces. In: Handbook of Deep Learning Applications, pp. 333–344. Springer, Cham (2019)

21. Kim, D., Sekhar Roy, S., Länsivaara, T., Deo, R., Samui, P. (eds.): Handbook of research on predictive modeling and optimization methods in science and engineering. IGI Global (2018)
22. Coombes, C.E., Abrams, Z.B., Li, S., Abruzzo, L.V., Coombes, K.R.: Unsupervised machine learning and prognostic factors of survival in chronic lymphocytic leukemia. J. Am. Med. Inform. Assoc. **27**(7), 1019–1027 (2020)
23. Ayyappan, V., Chang, A., Zhang, C., Paidi, S.K., Bordett, R., Liang, T., ... Pandey, R.: Identification and staging of B-cell acute lymphoblastic leukemia using quantitative phase imaging and machine learning. ACS Sensors **5**(10), 3281–3289 (2020)
24. Roy, S.S., Samui, P., Deo, R., Ntalampiras, S. (eds.): Big Data in Engineering Applications, vol. 44. Springer (2018)
25. Bandhu, A., Roy, S.S.: Classifying multi-category images using deep learning: a convolutional neural network model. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 915–919. IEEE (2017)
26. Roy, S.S., Roy, R., Balas, V.E.: Estimating heating load in buildings using multivariate adaptive regression splines, extreme learning machine, a hybrid model of MARS and ELM. Renew. Sustain. Energy Rev. **82**, 4256–4268 (2018)
27. Roy, S.S., Mallik, A., Gulati, R., Obaidat, M.S., Krishna, P.V.: A deep learning based artificial neural network approach for intrusion detection. In: The International Conference on Mathematics and Computing, pp. 44–53. Springer, Singapore (2017)
28. Sharma, K.K., Seal, A.: Spectral embedded generalized mean based k-nearest neighbors clustering with s-distance. Expert Syst. Appl. **169**, 114326 (2021)
29. Dong, Y., Ma, X., Fu, T.: Electrical load forecasting: a deep learning approach based on K-nearest neighbors. Appl. Soft Comput. **99**, 106900 (2021)
30. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
31. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., ... Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286**(5439), 531–537 (1999)
32. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. **2**(1–3), 37–52 (1987)
33. Mahmoudi, M.R., Heydari, M.H., Qasem, S.N., Mosavi, A., Band, S.S.: Principal component analysis to study the relations between the spread rates of COVID-19 in high risks countries. Alex. Eng. J. **60**(1), 457–464 (2021)
34. Roy, S.S., Viswanatham, V.M., Krishna, P.V.: An ensemble design approach based on bagging technique for filtering email spam. Int. J. Auton. Adapt. Commun. Syst. **10**(3), 247–260 (2017)
35. Roy, S.S., Biba, M., Kumar, R., Kumar, R., Samui, P.: A new SVM method for recognizing polarity of sentiments in twitter. In: Handbook of Research on Soft Computing and Nature-Inspired Algorithms, pp. 281–291. IGI Global (2017)
36. Roy, S.S., Sinha, A., Roy, R., Barna, C., Samui, P.: Spam email detection using deep support vector machine, support vector machine and artificial neural network. In: International Workshop Soft Computing Applications, pp. 162–174. Springer, Cham (2016)
37. Roy, S.S., Pratyush, C., Barna, C.: Predicting ozone layer concentration using multivariate adaptive regression splines, random forest and classification and regression tree. In: International Workshop Soft Computing Applications, pp. 140–152. Springer, Cham (2016)
38. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)

In Silico Drug Discovery Using Tensor Decomposition Based Unsupervised Feature Extraction

Y.-H. Taguchi

Abstract We have performed in silico drug repositioning using tensor decomposition based feature extraction. It turned out that it is a very effective method to identify drugs effective to the diseases not targeted by these drugs.

1 Introduction

Drug discovery is ever time consuming and expensive process. It often took more than ten years to find useful drug compounds effective to some specific disease. It is natural for people to try to shorten this period and reduce the expense required for drug development. Most effective usage of in silico drug discovery for drug discovery is to reduce the size or a set of candidate compounds from which we can start drug discovery. If we can select one tenth of drug candidate compounds as promising candidates, it will drastically reduce the efforts paid for drug discovery.

There are multiple ways to propose drug candidate compounds using computers. The most effective and popular way is so called ligand base drug discovery (LBDD) that tries to seek compounds similar to known effective compounds computationally [1]. Since LBDD requires only the information about compounds, it is computationally very effective and has superior power to identify new drugs. The critical disadvantage of LBDD is not to identify any compounds effective to diseases without known drugs. Thus, LBDD is useless to develop newly identified diseases. In order to compensate this difficulty, alternative strategy was proposed: so called structure based drug discovery (SBDD) [5]. SBDD does not make use of similarity between compounds but make use of binding affinity of individual compounds target proteins. Although SBDD can develop effective compounds to diseases without

Y.-H. Taguchi (✉)
Department of Physics, Chuo University, Tokyo, Japan
e-mail: tag@granular.com

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_7

known drugs, it has its own drawback; high computational resources. In contrast to LBDD that only seeks compounds whose structure is similar to known compounds, SBDD must perform so called docking simulation that requires massive computational resources. It must simulate dynamics of huge number of atoms of which proteins and compounds are composed. In addition to this, huge number of water molecules around them must be also simulated. This task is often impossible even using super computers. Furthermore, since docking simulation requires the knowledge of protein structure, if there are no knowledge about the structure of proteins targeted, the protein structure itself must be predicted [10]; usually computationally predicted protein structure includes some errors which results in inaccuracy of drug screening as well.

In order to compensate this problem, gene expression based drug discovery was developed [4]. In this flame-work, instead of comparing structures of compounds, gene expression profiles of cell lines and model animals treated by drugs are considered; if new compounds share gene expression with those of known drugs, they are regarded as newly identified drugs. Although this strategy, gene expression based drug discovery, is free from the problem that LBDD can find only compounds that share similar structures with known drugs, it is still not free from the problem that no new drugs can be identified if there are no known effective drug compounds. In order to further compensate this problem, we developed a new strategy based upon tensor decomposition (TD) unsupervised feature extraction (FE) [18]. In the following sections, we introduce the applications of TD based unsupervised FE to drug discovery using gene expression.

2 LINCS Data Set Analysed by TD Based Unsupervised FE

We applied TD based unsupervised FE to LINCS data sets in order to identify new drug candidate compounds [17]. In LINCS data sets, gene expression of not all but 978 genes are measured. By limiting the number of genes whose expression was measured, LINCS could test numerous number of combinations of cell lines and compounds. In my study [17], I employed combinations listed in Table 1; the number of cell lines is as many as 13 whereas the number of compounds tested vary from one hundred to two hundreds dependent upon the cell lines tested.

The procedure how we can infer the combinations of compounds and proteins that the compounds target is as follows. At first, individual gene expression profiles in one of 13 cell lines are formatted as tensors, $x_{ijk} \in \mathbb{R}^{978\times 6\times K}$ that represent expression of ith gene ad jth dose of kth compounds where K is number of compounds tested (numbers denoted as "all compounds" in Table 1). Higher order singular value decomposition (HOSVD) [18] was applied to x_{ijk} and we got

$$x_{ijk} = \sum_{\ell_1=1}^{978} \sum_{\ell_2=1}^{6} \sum_{\ell_3=1}^{M} G(\ell_1 \ell_2 \ell_3) u_{\ell_1 i} u_{\ell_2 j} u_{\ell_3 k} \tag{1}$$

Table 1 The number of the inferred compounds and inferred genes associated with significant dose-dependent activity. The target genes predicted by means of the comparison with the data showing upregulation of the expression of individual genes ('predicted targets') are also shown

Cell lines	BT20	HS578T	MCF10A	MCF7	MDAMB231	SKBR3
Tumor	Breast					
Inferred genes	41	57	42	55	41	46
Inferred compounds	4	3	2	6	5	6
All compounds	110	106	106	108	108	106
Predicted targets	418	576	476	480	560	423
Cell Lines	A549	HCC515	HA1E	HEPG2	HT29	PC3
Tumour	Lung		Kidney	Liver	Colon	Prostate
Inferred genes	45	46	48	54	50	63
Inferred compounds	8	5	7	2	2	9
All compounds	265	270	262	269	270	270
Predicted targets	428	352	423	396	358	439
Cell lines	A375					
Tumour	Melanoma					
Inferred genes	43					
Inferred compounds	6					
All compounds	269					
Predicted targets	421					

where $G \in \mathbb{R}^{978\times 6\times M}$ is core tensor, $u_{\ell_1 i} \in \mathbb{R}^{978\times 978}$, $u_{\ell_1 i} \in \mathbb{R}^{6\times 6}$, $u_{\ell_1 i} \in \mathbb{R}^{M\times M}$, are singular value vectors and are orthogonal matrices. In order to identify which combinations of genes and compounds are associated with dose dependence, we identified $u_{\ell_2 j}$ first. As can be seen in Fig. 1, regardless to cell lines, the second singular value vectors attributed dose, u_{2j}, are always associated with monotonic dose dependence. Thus, our method, TD based unsupervised FE, successfully identified monotonic dose dependence for all 13 cell lines.

Next we would like to identify which combinations of $u_{\ell i}$, which is singular value vectors attributed to genes, and $u_{\ell_3 k}$, which is attributed to compounds, are associated with monotonic dose dependence. In order to do this, we tried to identify how many $u_{\ell i}$ and $u_{\ell_3 k}$ are associated with negligibly large absolute $G(\ell_1, 2, \ell_3)$s. Then we found that $\ell_1, \ell_3 \leq 6$ is enough excluding PC cell lines, for which $\ell_1, \ell_3 \leq 8$ is enough. Assuming that $u_{\ell_1 i}$ and $u_{\ell_3 k}$ obeys Gaussian distribution, We attributed P-values to i and k as

$$P_i = P_{\chi^2}\left[> \sum_{\ell_1 \leq 6} \left(\frac{u_{\ell_1 i}}{\sigma_{\ell_1}} \right)^2 \right] \tag{2}$$

$$P_k = P_{\chi^2}\left[> \sum_{\ell_3 \leq 6} \left(\frac{u_{\ell_3 k}}{\sigma_{\ell_3}} \right)^2 \right] \tag{3}$$

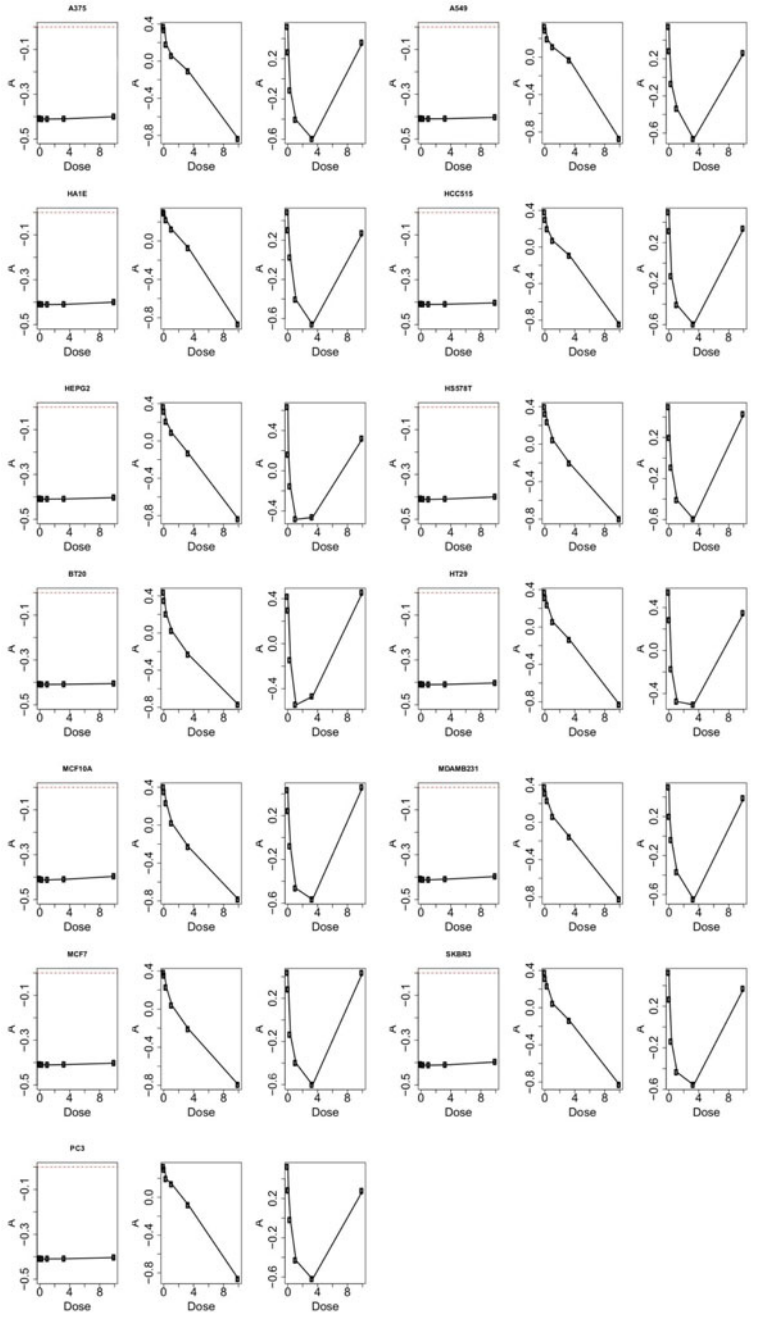

Fig. 1 $u_{\ell_2 j}$ with $1 \leq \ell_2 \leq 3$ (from left to right) for 13 cell lines

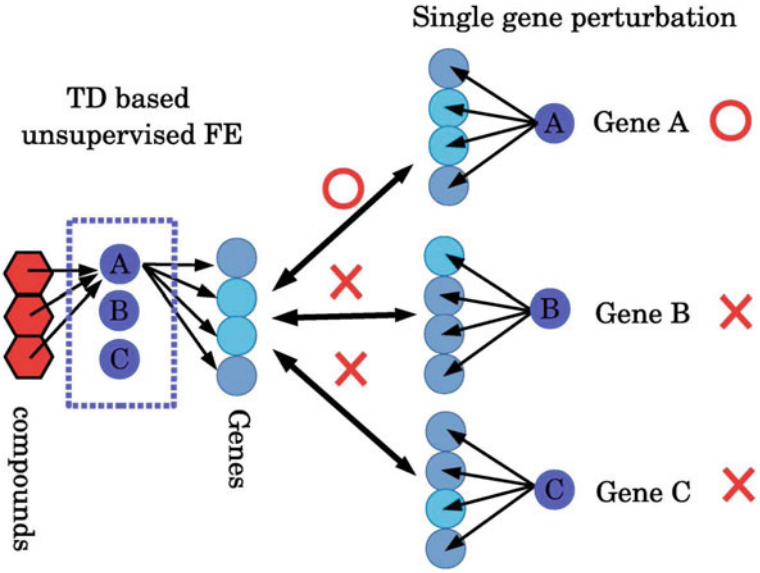

Fig. 2 How to infer target proteins. By means of TD-based unsupervised FE, a set of genes with the expression level alterations following the activity of specific compounds can be inferred ('inferred compounds' and 'inferred genes' in Table 1), but a compound's target genes (blue rectangle, 'predicted targets' in Table 1) cannot. Nonetheless, a list of inferred gene sets can be compared with that of the single-gene perturbations taken from Enrichr's 'Single Gene Perturbations category from GEO up', enabling identification of the compound's target genes

P-values were corrected by BH criterion [18] and is (genes) and ks (compounds) associated with adjusted P-values less than 0.01 are selected. The numbers of genes selected for individual cell lines are listed as "inferred genes" and the numbers of compounds selected for individual cell lines are listed as "inferred compounds" in Table 1.

Genes selected are not target proteins of selected compounds since interaction between proteins and compounds cannot affect amount of mRNA that code the proteins. Thus we need to infer target proteins based upon the list of "inferred genes". How we could perform this task was illustrated in Fig. 2. At first, we uploaded genes selected by TD based unsupervised FE applied to x_{ijk}, whose numbers are listed as "Inferred genes" in Table 1, to Enrichr [11] which is an comprehensive enrichment servers that validate various enrichment of biological terms in the set of uploaded genes. AMong categories in Enrichr, we focused category named "Single Gene Perturbations category from GEO up" which might be genes that code proteins targeted by compounds, since single gene perturbation is expected to affect expression of gene in downstream in the similar ways to compounds that bind to proteins (Fig. 2). Then by selecting genes associated with adjusted P-values less than 0.01, we can infer proteins whose numbers are denotes as "predicted target" in Table 1. Apparently, our strategy could identify proteins targeted by compounds.

Next step is to validate if "predicted proteins" are really target proteins of compounds or not. In order that, we compare our "predicted proteins" of "inferred compounds" with previous knowledge which was downloaded from two data bases: drug2gene.com [14] and DSigDB [23]. Table 2 shows the results of Fisher's exact test that validated if "predicted target" significantly overlap with the target proteins reported in either of two data bases. It is obvious that "predicted target" significantly overlap with target proteins of compounds reported in either of two data bases. Thus, the prediction by TD bases unsupervised FE seems to be trustable.

Although in order to predict target proteins from "inferred genes" we compared them with single gene perturbations, any kind of information that represent gene-gene interaction. Alternatively, we employed "PPI hub proteins" category in Enrichr (Table 3). Again, "predicted target" significantly overlap with the target proteins reported in the data bases. Thus, our strategy is robust and is supposed to be trustable. Although Tables 2 and 3 support our strategy, there are also some supportive evidences. First of all, although more than one hundred genes are tested for individual cell lines, "inferred compounds" is same regardless to cell lines. Considering that there are distinct cell lines with each other, the high coincidence cannot be accidental.

3 DrugMatrix Analyzed by TD Based Unsupervised FE

Although in the previous section we successfully screened promising drug candidate compounds together with their possible target proteins, these drugs are possibly for cancers, since all cell lines were generated from tumors. If one would like to seek drug candidate compounds, we need those for specific diseases, but it is generally impossible. For cancers, since tumor cells can be easily immortalized, other cells cannot be immortalized; this strictly reduces the opportunity to get cells use for drug candidate compounds evaluation. If we can seek drug candidate compounds without direct treatments with drug candidate compounds, it will be vary useful.

For this purpose, we tried to integrated gene expression profiles between model animal treated by drugs and disease patients [16]. If we can identify some drugs that can "reverse" the gene expression alteration caused by disease, it might be the promising drug candidate compounds. For this purpose, we made use of DrugMatrix data set [13] that store gene expression profiles of rat tissues treated with various drug compounds. Gene expression profiles stored in DrugMatrix was integrated with those of disease patients as follows in order to infer drug candidate compounds for diseases.

3.1 Heart Failure

In order to infer compound and target proteins based upon gene expression profiles. From DrugMatrix, we retrieved gene expression profiles of the rat left ventri-

Table 2 Compound–gene interactions presented in Table 1 that significantly overlap with interactions described in two datasets. For each compound in the table, the upper row: the drug2gene.com dataset was used for comparisons [14], the lower row: the DSigDB dataset was used for comparisons [23]. Columns represent cell lines used in the analysis: (1) BT20, (2) HS578T, (3) MCF10A, (4) MCF7, (5) MDAMB231, (6) SKBR3, (7) A549, (8) HCC515, (9) HA1E, (10) HEPG2, (11) HT29, (12) PC3, (13) A375

Compounds	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
dabrafenib													○
													○
dinaciclib							○	○	○	○	○	○	○
							○	○	○	○	○	○	○
CGP-60474			○	○	○	○	○		○			○	○
			×	×	×	×	×		×			×	○
LDN-193189	○				○								○
	○				○								○
OTSSP167							–	–		–		–	–
							○	○		○		○	○
WZ-3105		–		–	–		–	–	–			–	–
		○		○	○		○	○	○			○	○
AT-7519				○		○		○	○			○	
				○		○		○	○			○	
BMS-387032				○		○	○		○				
				○		○	○		○				
JNK-9L									○				
									○				
alvocidib	○	○	○	○	○	○			○				
	–	–	–	–	–	–			–				
GSK-2126458							–					–	
							–					–	
NVP-BEZ235							○					○	
							×					×	
torin-2							×					×	
							○					○	
NVP-BGT226					–			–			–	–	
					–			–			–	–	
QL-XII-47	–												
	–												
celastrol	○												
	–												
A443654		○		○									
		○		○									
NVP-AUY922					×	○							
					–	–							
radicicol						○							
						–							

○: a significant overlap between the datasets ($P < 0.05$); ×: no significant overlap between the datasets; —: no data; blank: no significant dose–response relation was identified. The confusion matrix.

Table 3 A significant overlap demonstrated between compound–target interactions presented in Table 1 and drug2gene.com. In this case, the 'PPI Hub Proteins' category in Enrichr was used. Labels (1) to (13) represent the same cell lines as described in Table 2

Compounds	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
dinaciclib											○	○	○
CGP-60474			○	○	○	○	○		○			○	○
LDN-193189					○								
AT-7519				○		○		○	○			○	
BMS-387032				○		○	○		○				
alvocidib		○	○	○	○	○			○				
NVP-BEZ235												○	
celastrol	○												
A443654		○		○									
NVP-AUY922					○	○							
radicicol						○							

cle (LV) treated with 218 drugs for which four time points (1/4, 1, 3, and 5 days after treatment) were available. It is formatted as tensor $x_{j_1 j_2 I} \in \mathbb{R}^{218\times 4\times 3937}$ that represents expression of ith gene at j_2th time points after the treatment of j_1th compound among 218 compounds used for treatment. The reason why the number of genes considered here is as small as 3937 is because we restricted genes to those shared with human in order to perform integrated analysis with human gene expression profile. For gene expression human patients, we obtained human heart gene expression profiles represent 82 patients with idiopathic dilated cardiomyopathy, 95 patients with ischemic stroke, and 136 healthy controls. They are formatted as a tensor $x_{j_3 i} \in \mathbb{R}^{313\times 3937}$ that represents expression of ith gene of j_3th patients. Then a tensor $x_{j_1 j_2 j_3 i} = x_{j_1 j_2 j_3 i} =\in \mathbb{R}^{218\times 4\times 313\times 3937}$ was generated and HOSVD was applied and we got

$$x_{j_1 j_2 j_3 i} = \sum_{\ell_1=1}^{218}\sum_{\ell_2=1}^{4}\sum_{\ell_3=1}^{313}\sum_{\ell_4=1}^{3937} G(\ell_1\ell_2\ell_3\ell_4)u_{\ell_1 j_1}u_{\ell_2 j_2}u_{\ell_3 j_3}u_{\ell_4 i} \tag{4}$$

where $G \in \mathbb{R}^{218\times 4\times 313\times 3937}$ is a core tensor, $u_{\ell_1 j_1} \in \mathbb{R}^{218\times 218}$, $u_{\ell_2 j_2} \in \mathbb{R}^{4\times 4}$, $u_{\ell_3 j_3} \in \mathbb{R}^{313\times 313}$, and $u_{\ell_4 i} \in \mathbb{R}^{3937\times 3937}$, are singular value vectors and are orthogonal matrices. Before selecting genes and compounds, we have to know which singular value vector of time points, $u_{\ell_2 j_2}$, is correlated with time (Fig. 3). Then, I decided to use the second singular value vectors ($\ell_2 = 2$) for heart failure. Next we validated singular value vectors, $u_{\ell_3 j_3}$, attributed to patients in order to see which one is associated with distinction between healthy controls and disease patients (Fig. 4a). Then we can regard that $u_{2 j_3}$ as well as $u_{\ell_3 j_3}$ are associated with the distinction between healthy control and patients. Next task is to identify which $u_{\ell_4 i}$ attributed to genes and $u_{\ell_1 j_1}$ attributed to compounds have the larger $|G(\ell_1, 2, 2, \ell_4)|$ or $|G(\ell_1, 2, 3, \ell_4)|$ (i.e. associated with selected $u_{\ell_2 j_2}$ and $u_{\ell_3 j_3}$ that represent time dependence and distinction

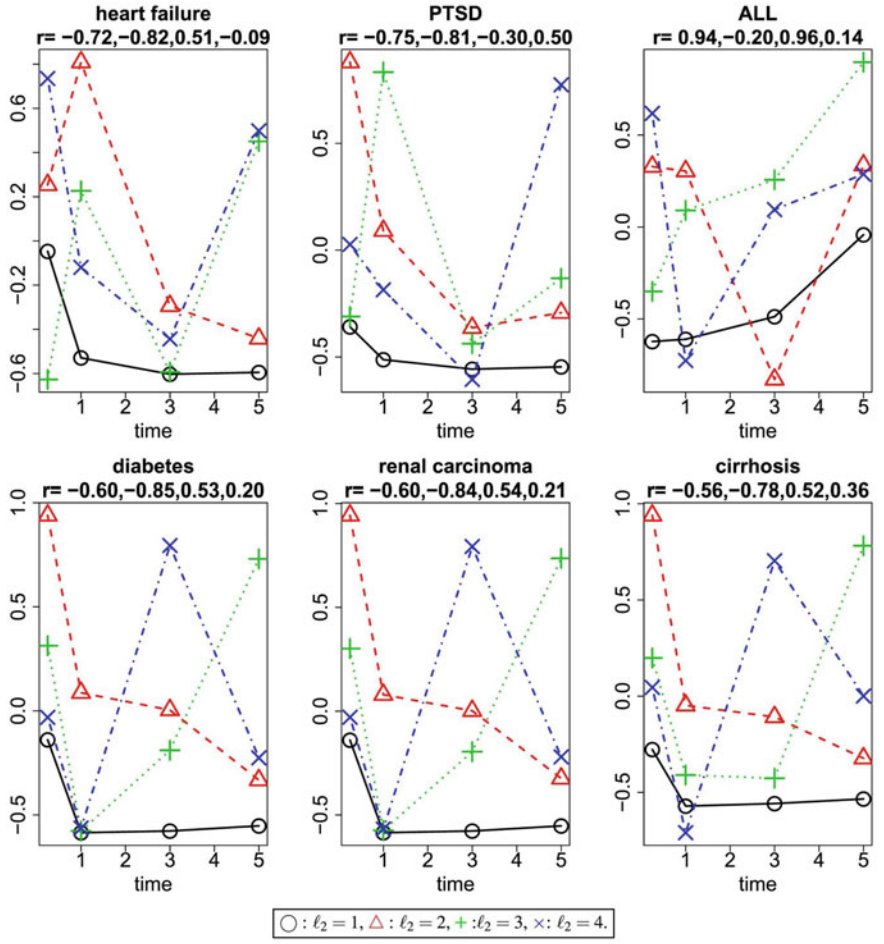

Fig. 3 Time point singular value vectors. r represents Pearson's correlation coefficients between time points (1/4, 1, 3, and 5 days after a treatment) and the first to forth singular value vectors of time points, $u_{\ell_2 j_2}$, $1 \leq j_2, \ell_2 \leq 4$. Black open circles: $\ell_2 = 1$, red open triangles: $\ell_2 = 2$, green crosses: $\ell_2 = 3$, and blue crosses: $\ell_2 = 4$. $j_2 = 1, 2, 3, 4$ correspond to time points 1/4, 1, 3, and 5 days, respectively

between healthy controls and patients). Since there are no definite threshold values for $|G|$, we selected top 10 ℓ_4 associated with larger $|G(\ell_1, 2, 2, \ell_4)|$ or $|G(\ell_1, 2, 3, \ell_4)|$ and found that they are $\ell_4 = 21, 25, 27, 28, 33, 36, 37, 38, 41, 42$. On the hand, only $\ell_1 = 2$ was associated with these top ten ranked $|G|$. Then P-values were attributed to i and j_1 as (Fig. 5)

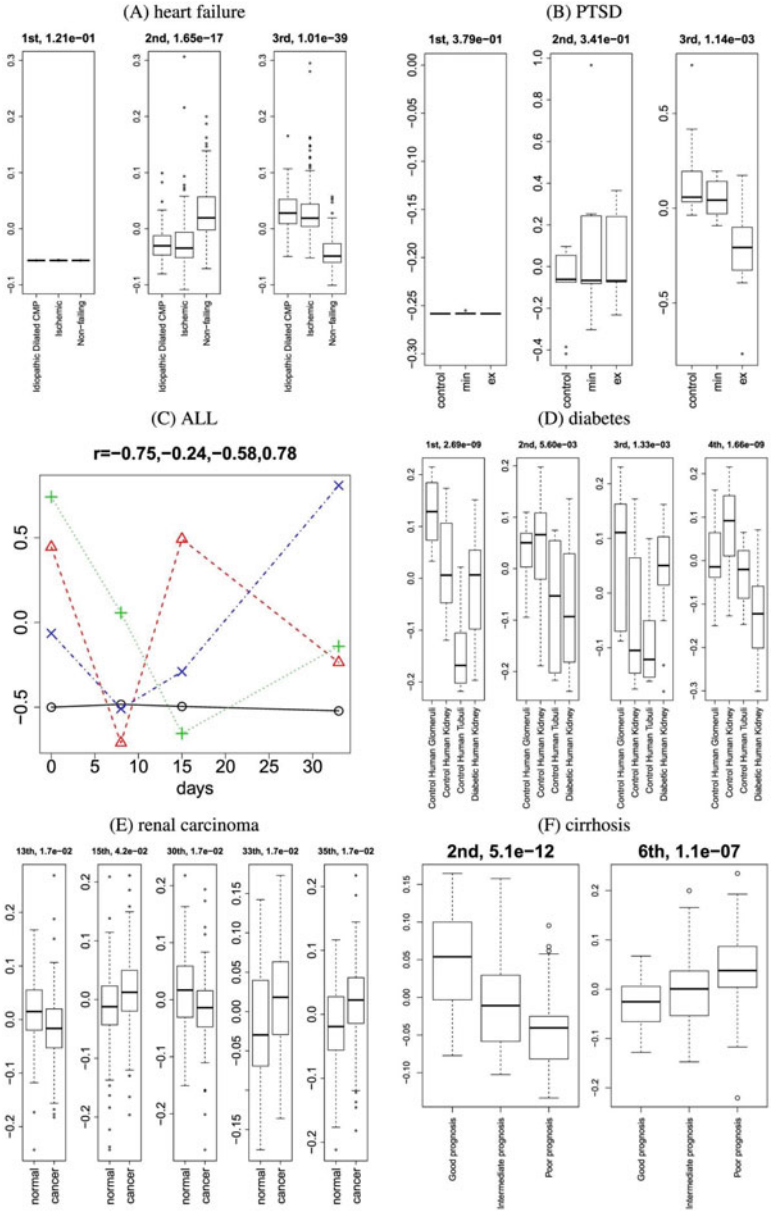

Fig. 4 Boxplots [for (A), (B), (D), (E), and (F)] and time dependence (C) for sample singular value vectors. The numbers above boxplots are P-values for (A), (B), and (D) and adjusted P-vales for (E) and (F), computed by categorical regression (in other words, ANOVA). The numbers above time dependence (C) are correlation coefficients. (C) Black open circles: $\ell_3 = 1$, red open triangles: $\ell_3 = 2$, green crosses: $\ell_3 = 3$, and blue crosses: $\ell_3 = 4$. $j_3 = 1, 2, 3, 4$ correspond to time points 0, 8, 15, and 33 days, respectively

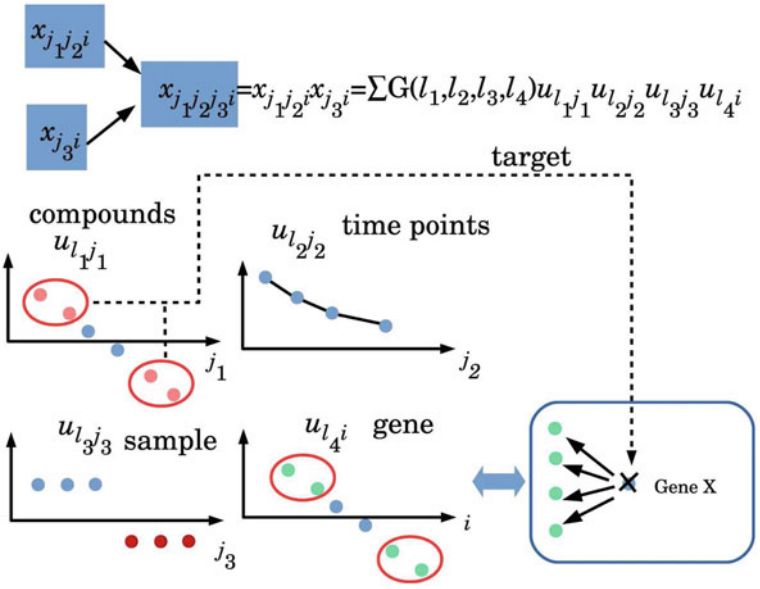

Fig. 5 Intuitive illustration of the present strategy. Suppose there is a tensor, $x_{j_1 j_2 i}$, which represents the ith gene expression at the j_2th time point after the j_1th compound is given to a rat; these data are taken from the DrugMatrix [13] data set. There is also a matrix, $x_{j_3 i}$, which represents the ith gene expression of the j_3th sample; samples typically include disease samples and control samples. Tensor $\tilde{x}_{j_1 j_2 j_3 i}$ was generated as a 'mathematical product' of $x_{j_1 j_2 i}$ and $x_{j_3 i}$. Then, tensor $\tilde{x}_{j_1 j_2 j_3 i}$ is decomposed, and singular value matrix of compounds $u_{\ell_1 j_1}$, singular value matrix of time points $u_{\ell_2 j_2}$, sample singular value matrix $u_{\ell_3 j_3}$, and gene singular value matrix $u_{\ell_4 i}$ are obtained. Among them, I selected the combinations of ℓ_k, $1 \leq k \leq 4$, which are simultaneously associated with all of the following: i) core tensor $G(\ell_1, \ell_2, \ell_3, \ell_4)$ with a large enough absolute value, ii) a singular value vector of time points, $u_{\ell_2 j_2}$, whose value significantly varies with time, and iii) sample singular value vector $u_{\ell_3 j_3}$. These parameters are different between a disease (red filled circles) and control samples (cyan filled circles). Finally, using gene singular value vector $u_{\ell_4 i}$ and compound singular value vector $u_{\ell_1 j_1}$, compounds (filled pink circles) and genes (filled light-green circles) associated with $G(\ell_1, \ell_2, \ell_3, \ell_4)$s with large enough absolute values are selected. Next, if the selected genes are coincident with the genes associated with a significant alteration when gene X is knocked out (or overexpressed), then the compounds are assumed to target gene X

$$P_i = P_{\chi^2}\left[> \sum_{\ell_4 \in \begin{pmatrix} 21, 25, 27, 28, 33 \\ 36, 37, 38, 41, 42 \end{pmatrix}} \left(\frac{u_{\ell_4 i}}{\sigma_{\ell_4}} \right)^2 \right] \tag{5}$$

$$P_{j_1} = P_{\chi^2}\left[> \left(\frac{u_{2 j_1}}{\sigma_2} \right)^2 \right] \tag{6}$$

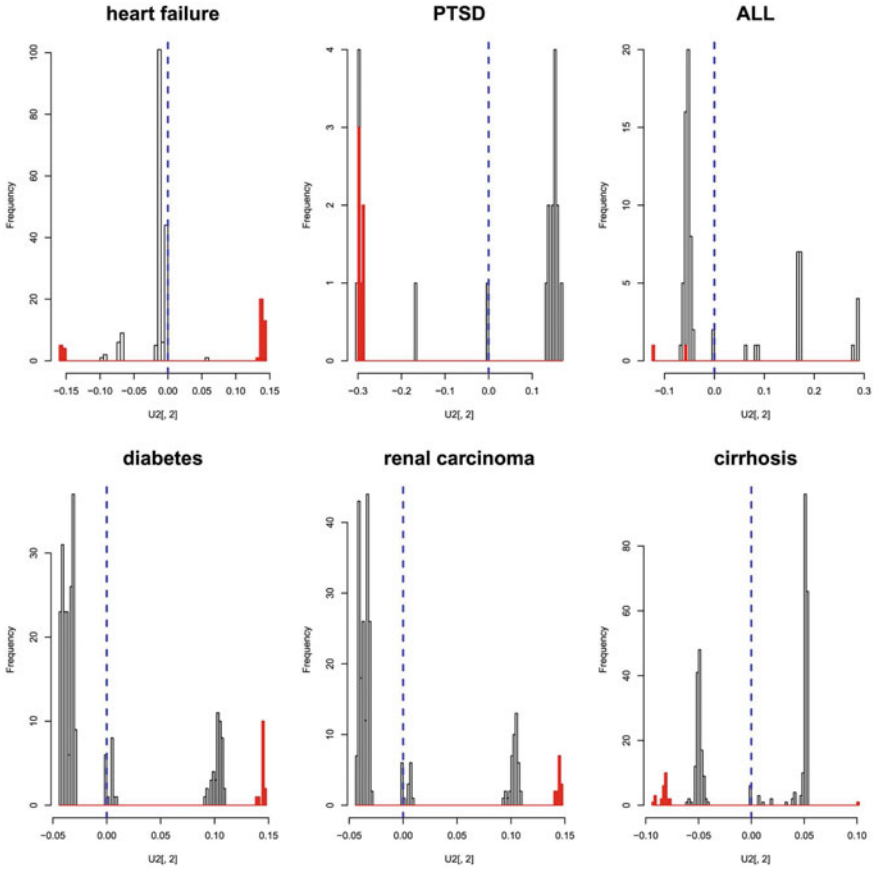

Fig. 6 A histogram of the second singular value vectors of compounds. Red parts represent drugs selected as outliers, and vertical blue dashed lines are origins of the axes, by the distances from which the outliers were identified

Then P-values were corrected by BH criterion [18] and 274 genes associated with adjusted P-values less than 0.01 were selected. Unfortunately, this strategy could not be applied to select compounds. Thus instead, we select 43 outlier compounds in u_{2j} (Fig. 6). In order to infer target proteins based upon the selected 274 genes, these genes were uploaded to Enrichr and 534 genes and 556 genes associated with adjusted P-values less than 0.01 for the categories 'Single Gene Perturbations from GEO up' and 'Single Gene Perturbations from GEO down' were selected, respectively. We next tried to validate if predicted target proteins were reported to be target proteins of 43 compounds selected in the above. To obtain the list of known drug target proteins, I used DINIES [21]. Table 4 shows the confusion matrices between predicted target proteins and known target proteins retrieved from DINES of 43 compounds selected by TD based unsupervised FE. Both Fisher's exact test and χ^2 test

Table 4 Fisher's exact test (P_F) and the uncorrected χ^2 test (P_{χ^2}) of known drug target proteins regarding the inference of the present study. Rows: known drug target proteins (DINIES). Columns: Inferred drug target proteins using 'Single Gene Perturbations from GEO up' or 'Single Gene Perturbations from GEO down'. OR: odds ratio

	Single Gene Perturbations from GEO up					Single Gene Perturbations from GEO down				
	F	T	P_F	P_{χ^2}	RO	F	T	P_F	P_{χ^2}	RO
heart failure	F 521	517	3.4×10^{-4}	3.9×10^{-4}	3.02	628	416	1.3×10^{-3}	7.3×10^{-4}	2.61
	T 13	39				19	33			
PTSD	F 500	560	3.8×10^{-2}	3.1×10^{-2}	2.67	532	529	6.1×10^{-3}	4.5×10^{-3}	3.81
	T 6	18				5	19			
ALL	F 979	89	2.7×10^{-1}	3.0×10^{-1}	2.19	1009	57	1.0×10^{0}	-	-
	T 10	2				12	0			
diabetes	F 889	177	1.2×10^{-2}	7.1×10^{-3}	3.00	936	130	3.6×10^{-4}	2.0×10^{-5}	5.13
	T 15	9				14	10			
renal carcinoma	F 847	219	2.0×10^{-2}	1.2×10^{-2}	2.75	895	169	4.3×10^{-2}	2.2×10^{-2}	2.64
	T 14	10				16	8			
cirrhosis	F 572	219	1.1×10^{-2}	8.1×10^{-3}	2.91	595	169	1.6×10^{-3}	1.1×10^{-3}	3.81
	T 8	10				7	8			

suggest that there are significant overlaps of target proteins between prediction by TD based unsupervised FE and target proteins listed in DINES. Thus our strategy was successful and we could infer candidate compounds for heart failure together with possible target proteins.

As can be seen in Table 4, there are as many as four to five hundreds false positive. One might wonder it it is acceptable. Nevertheless, we are not very sure if these apparent false positives are really false positives since proteins that were never tested toward any of 43 compounds must be counted as false positives. In order to know how well our strategy work need further evaluations of target proteins, which we cannot perform. Anyway, in spite of huge number of false positives, our strategy could derive the results significantly coincident with proteins reported to be targets in DENIS.

3.2 Other Diseases than Heart Failure

We have also repeated the procedures described in the previous subsection toward additional five diseases. This was summarised in Table 5 and Fig. 7, together with singular value vectors in Figs. 3 and 4. As can be seen in Table 4, other than ALL, for all other five diseases, we have found sets of target proteins that significantly overlap with target proteins reported for selected compounds.

Table 5 A summary of TDs and identification of various singular value vectors for identification of candidate drugs and genes used to find genes encoding drug target proteins. In all cases, ℓ_1 stands for singular value vectors of compounds, whereas ℓ_k with the last (largest) k denotes gene singular value vectors. ℓ_2 stands for singular value vectors of time points in DrugMatrix data. The remaining singular value vectors correspond to sample singular value vectors dependent on the properties of gene expression profiles of diseases. See also Fig. 7 for the corresponding data

	tensors				
diseases	DrugMatrix	disease	generated	core tensor	singular value vectors
heart	$\tilde{x}_{j_1 j_2 i}$	$x_{j_3 i}$	$x_{j_1 j_2 j_3 i}$	$G(\ell_1\ell_2\ell_3\ell_4)$	$u_{\ell_k,j_k}, k \leq 3, u_{\ell_4,i} \in \mathbb{R}^{N_k \times N_k}$
failure	$\in \mathbb{R}^{N_1 \times N_2 \times N_4}$	$\in \mathbb{R}^{N_3 \times N_4}$	$\in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4}$		$(N_1, N_2, N_3, N_4) = (218, 4, 313, 3937)$
Selected				$\ell_1 = 2; \ell_2 = 2; \ell_3 = 2, 3; \ell_4 = 21, 25, 27, 28, 33, 36, 37, 41, 42, 48$	
PTSD	$\tilde{x}_{j_1 j_2 i}$	$x_{j_3 j_4 i}, k = 4, 5$	$x_{j_1 j_2 j_3 j_4 j_5 i}$	$G(\ell_1\ell_2\ell_3\ell_4\ell_5\ell_6)$	$u_{\ell_k,j_k}, k \leq 5, u_{\ell_6,i} \in \mathbb{R}^{N_k \times N_k}$
rat model	$\in \mathbb{R}^{N_1 \times N_2 \times N_6}$	$\in \mathbb{R}^{N_3 \times N_k \times N_6}$	$\in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4 \times N_5 \times N_6}$		$(N_1, N_2, N_3, N_4, N_5, N_6) = (22, 4, 2, 15, 15, 7501)$
Selected				$\ell_1 = 2; \ell_2 = 2; \ell_3 = 1; \ell_4 = \ell_5 = 3; \ell_6 = 75, 77, 81, 83, 84, 85, 89, 90, 102$	
ALL	$\tilde{x}_{j_1 j_2 i}$	$x_{j_3 j_4 i}$	$x_{j_1 j_2 j_3 j_4 i}$	$G(\ell_1\ell_2\ell_3\ell_4\ell_5)$	$u_{\ell_k,j_k}, k \leq 4, u_{\ell_5,i} \in \mathbb{R}^{N_k \times N_k}$
	$\in \mathbb{R}^{N_1 \times N_2 \times N_5}$	$\in \mathbb{R}^{N_3 \times N_4 \times N_5}$	$\in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4 \times N_5}$		$(N_1, N_2, N_3, N_4, N_5) = (77, 4, 4, 74, 2597)$
Selected				$\ell_1 = 2, 3, 5, 6, 9, 10; \ell_2 = 3; \ell_3 = 4; \ell_5 = 1, 2, 3, 5$	
diabetes	$\tilde{x}_{j_1 j_2 i}$	$x_{j_3 i}$	$x_{j_1 j_2 j_3 i}$	$G(\ell_1\ell_2\ell_3\ell_4)$	$u_{\ell_k,j_k}, k \leq 3, u_{\ell_4,i} \in \mathbb{R}^{N_k \times N_k}$
	$\in \mathbb{R}^{N_1 \times N_2 \times N_4}$	$\in \mathbb{R}^{N_3 \times N_4}$	$\in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4}$		$(N_1, N_2, N_3, N_4) = (253, 4, 69, 3489)$
Selected				$\ell_1 = 2; \ell_2 = 2; \ell_3 = 1, 4; \ell_4 = 1, 4$	
renal	$\tilde{x}_{j_1 j_2 i}$	$x_{j_3 i}$	$x_{j_1 j_2 j_3 i}$	$G(\ell_1\ell_2\ell_3\ell_4)$	$u_{\ell_k,j_k}, k \leq 3, u_{\ell_4,i} \in \mathbb{R}^{N_k \times N_k}$
carcinoma	$\in \mathbb{R}^{N_1 \times N_2 \times N_4}$	$\in \mathbb{R}^{N_3 \times N_4}$	$\in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4}$		$(N_1, N_2, N_3, N_4) = (253, 4, 202, 4036)$
Selected				$\ell_1 = 2; \ell_2 = 2; \ell_3 = 13, 15, 30, 33, 35; \ell_4 = 186, 215, 233, 244, 251, 269, 274, 309, 312, 318$	
cirrhosis	$\tilde{x}_{j_1 j_2 i}$	$x_{j_3 i}$	$x_{j_1 j_2 j_3 i}$	$G(\ell_1\ell_2\ell_3\ell_4)$	$u_{\ell_k,j_k}, k \leq 3, u_{\ell_4,i} \in \mathbb{R}^{N_k \times N_k}$
	$\in \mathbb{R}^{N_1 \times N_2 \times N_4}$	$\in \mathbb{R}^{N_3 \times N_4}$	$\in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4}$		$(N_1, N_2, N_3, N_4) = (355, 4, 216, 3961)$
Selected				$\ell_1 = 2; \ell_2 = 2; \ell_3 = 2, 6; 2 \leq \ell_4 \leq 10$	

3.3 A Possible Drug Candidate Compound for Cirrhosis, Bezafibrate

Although we have successfully identified drug candidate compounds for various disease and possible target proteins of these compounds, one might wonder if we can really find effective drugs for target diseases. In order to demonstrate of powers that can infer practical drugs for disease, we consider cirrhosis, since three are no known drugs for cirrhosis; if we can infer some drugs, it is very useful. In order that, we selected top ranked genes, CYPOR and HNF4A, and one drug candidate compound, bezafibrate, which was infer to target these proteins. Based on docking simulation [16], bezafibrate likely bind to these proteins. In addition to this, bezafibrate was tested multiple times as a drug effective to cirrhosis [3, 6–8, 12, 20, 22]; there is even a review about the effectiveness of bezafibrate toward cirrhosis [15]. All of these suggest that our strategy is even an effective one to find one specific drug toward the targeted disease.

4 COVID-19 Drug Discovery

Although we have found combinations of drugs and their target proteins of various diseases, no infectious diseases were not included. Gene expression based drug discovery is not useful for infectious disease, since drugs must target not human patients

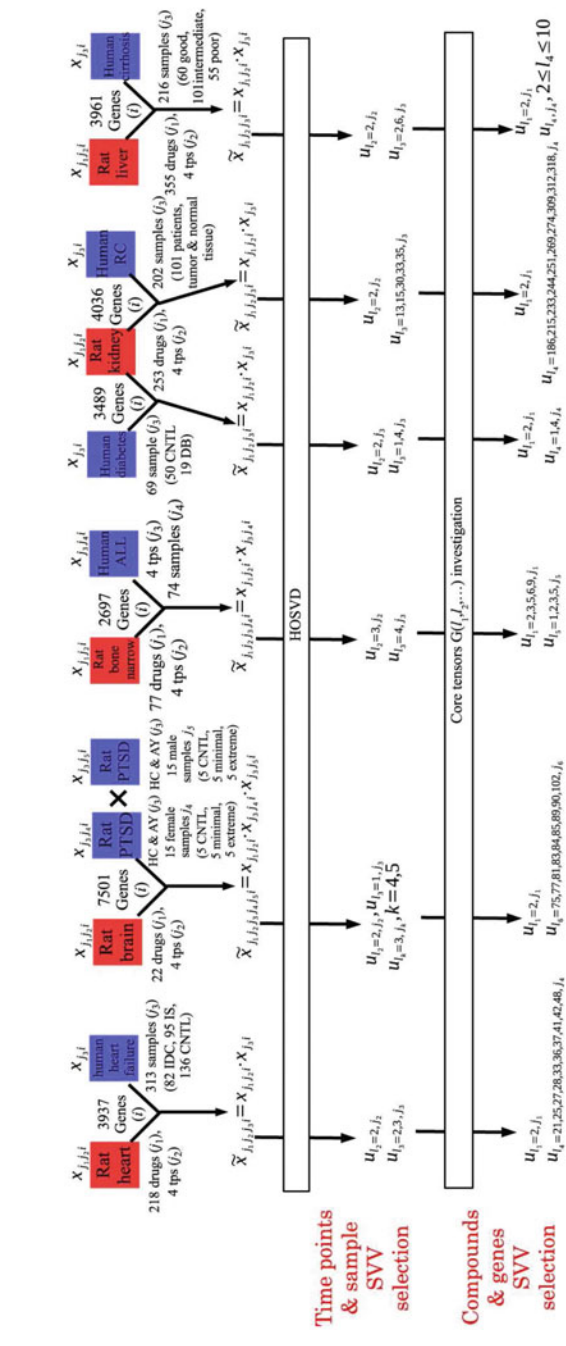

Fig. 7 Schematics that illustrate the procedure of TD-based unsupervised FE applied to the various disease and DrugMatrix datasets. SVV: singular value vector

(hosts) but pathogen. It is unlikely that there are many gene expression profiles of pathogen treated by various compounds. First of all, the number of pathogen is as many as disease. In that case, drug should directly target not pathogen but disease. Especially, infectious disease caused by virus is more difficult, since virus itself cannot have its own gene expression profiles.

In spite of that, in some cases, gene expression based drug discovery is still effective; if symptom was cased by interaction between virus and human, we can target human side of disease. For example, although COVID-19 can kill human beings, it is believed that the reason is primarily over reaction toward SARS-CoV-2 infection in human side [9]. In this sense, it is an interesting try to identify effective drugs towards COVID-19 with gene expression based drug discovery strategy. For this purpose, we applied TD based unsupervised FE to gene expression profiles of human lung cancer cell lines infected by SARS-CoV-2 [19].

Gene expression profiles were measured for five human lung cancer cell lines with three biological replicates. Then gene expression profile was formatted as a tensor $x_{ijkm} \in \mathbb{R}^{N \times 5 \times 2 \times 3}$ that represents expression of ith gene in jth cell lines infected ($k = 2$) or not infected ($k = 1$) mth biological replicate. Applying HOSVD to x_{ijkm} we got

$$x_{ijkm} = \sum_{\ell_1=1}^{5} \sum_{\ell_2=1}^{2} \sum_{\ell_3=1}^{3} \sum_{\ell_4=1}^{N} G(\ell_1, \ell_2, \ell_3, \ell_4) u_{\ell_1 j} u_{\ell_2 k} u_{\ell_3 m} u_{\ell_4 i} \tag{7}$$

$u_{\ell_1 j} \in \mathbb{R}^{5 \times 5}$, $u_{\ell_2 k} \in \mathbb{R}^{2 \times 2}$, $u_{\ell_3 m} \in \mathbb{R}^{3 \times 3}$, $u_{\ell_4 i} \in \mathbb{R}^{N \times N}$ are singular value matrices which are orthogonal matrices. The tensor was normalized as $\sum_i x_{ijkm} = 0$ and $\sum_i x_{ijkm}^2 = N$. $G(\ell_1, \ell_2, \ell_3, \ell_4) \in \mathbb{R}^{5 \times 2 \times 3 \times N}$ is a core tensor that represents a weight of the combination of $\ell_1, \ell_2, \ell_3, \ell_4$.

As usual, we need to investigate singular value vectors, $u_{\ell_1 j} u_{\ell_2 k} u_{\ell_3 m}$, attributed to samples in order to find which of them are associated with desired properties. Here desired properties are

- independent of cell lines
- independent of biological replicate
- distinct between infectious and non-infectious cell lines.

These properties are full filled if we can find $u_{\ell_1 j}$ whose value is constant regardless to j, $u_{\ell_3, m}$ whose value is constant regardless to m, and $u_{\ell_2 k}$ whose value is oppositely signed between $k = 1$ and $k = 2$. As can be seen in Fig. 8, these conditions were fulfilled for $\ell_1 = \ell_3 = 1$ and $\ell_2 = 2$. Thus next task is to find ℓ_4 that share the larger $|G|$ with u_{1j}, u_{2k}, u_{1m}. As can be seen in Table 6, u_{5i} has the largest $|G(1, 2, 1, \ell_4)|$. Thus we decided to use u_{5i} for selecting genes that play critical roles in SAR-CoV-2 infection. Then P-values were attributed to ith gene as

$$P_i = P_{\chi^2} \left[> \left(\frac{u_{\ell_4 i}}{\sigma_{\ell_4}} \right)^2 \right] \tag{8}$$

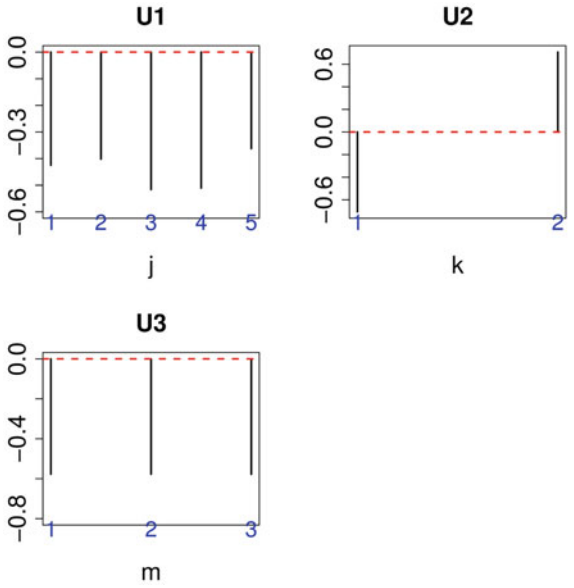

Fig. 8 Singular value vectors obtained by the HOSVD algorithm. U1:u_{1j}, U2:u_{2k}, U3:u_{1m}, See Materials and Methods for the definitions of j, k, and m

Table 6 $G(1, 2, 1, \ell_4)$s computed by the HOSVD algorithm

ℓ_4	$G(1, 2, 1, \ell_4)$	ℓ_4	$G(1, 2, 1, \ell_4)$
1	−21.409671	6	−12.388615
2	5.183297	7	8.437642
3	−21.426437	8	13.322888
4	10.030564	9	−1.850982
5	62.518121	10	9.211437

Computed P-values were adjusted by BH criterion [18] and 163 genes (is) associated with adjusted P-values were selected.

In order to validate whether selected 163 genes are biologically reliable, we uploaded 163 genes to Enrichr and found numerous reasonable biological terms significantly enriched [19]. In addition to this, we compared 163 genes with human protein genes known to be interacting with SARS-CoV-2 proteins during infection (Table 7). Interestingly, selecting 163 human genes are strongly coincident with human genes known to be interacting SARS-CoV-2 proteins.

Since we have selected limited number of human genes supposed to be interacting with SARS-CoV-2 proteins, to find drug compounds that can affect expression of 163 genes will be a good strategy to find effective drugs, since no virus can function without the interaction with human proteins. In order that, we investigated Enrihr

Table 7 Coincidence between 163 genes and human proteins whose numbers are reported in the previous publication [19]

SARS-CoV-2 proteins	P values	Odds Ratio
SARS-CoV2 E	6.55×10^{-27}	10.16
SARS-CoV2 M	1.38×10^{-26}	8.42
SARS-CoV2 N	4.61×10^{-24}	11.43
SARS-CoV2 nsp1	1.06×10^{-20}	10.00
SARS-CoV2 nsp10	3.40×10^{-20}	11.52
SARS-CoV2 nsp11	1.13×10^{-29}	10.66
SARS-CoV2 nsp12	4.87×10^{-20}	9.48
SARS-CoV2 nsp13	6.04×10^{-33}	11.17
SARS-CoV2 nsp14	1.75×10^{-22}	12.05
SARS-CoV2 nsp15	1.85×10^{-20}	10.23
SARS-CoV2 nsp2	4.81×10^{-33}	11.79
SARS-CoV2 nsp4	5.79×10^{-29}	10.26
SARS-CoV2 nsp5	3.78×10^{-25}	12.36
SARS-CoV2 nsp5_C145A	3.75×10^{-17}	11.39
SARS-CoV2 nsp6	9.47×10^{-26}	9.00
SARS-CoV2 nsp7	1.93×10^{-29}	10.81
SARS-CoV2 nsp8	1.11×10^{-29}	10.14
SARS-CoV2 nsp9	5.54×10^{-29}	12.24
SARS-CoV2 orf10	5.29×10^{-34}	12.37
SARS-CoV2 orf3a	2.06×10^{-28}	9.95
SARS-CoV2 orf3b	1.89×10^{-29}	11.80
SARS-CoV2 orf6	8.81×10^{-26}	10.37
SARS-CoV2 orf7a	1.69×10^{-28}	10.00
SARS-CoV2 orf8	5.94×10^{-28}	9.25
SARS-CoV2 orf9b	6.54×10^{-30}	12.12
SARS-CoV2 orf9c	1.11×10^{-28}	8.35
SARS-CoV2 Spike	8.22×10^{-26}	10.08

categories “LINCS L1000 Chem Pert up/down”, “Drug perturbations from GEO” and “Drug matrix”. Then we have found huge number of promising drug candidate compounds were detected [18]. Although we are not willing list all of drugs here, we would like to discuss one specific compounds, ivermectin, which was recently reported as a promising drug candidate compound toward COVID-19 [2]. This was detected in our analysis within DrugMatrix category (Table 8). Thus, our strategy was expected to be an effective one that can infer drug candidate compounds using gene expression profiles, even for infectious diseases.

Table 8 Ivermectin detected in DrugMatrix category in Enrichr

Term	Overlap	P-value	Adjusted P-value
Ivermectin-7.5 mg/kg in CMC-Rat-Liver-1d-dn	12/277	2.98E−06	9.93E−06
Ivermectin-7.5 mg/kg in CMC-Rat-Liver-5d-dn	12/289	4.60E−06	1.44E−05
Ivermectin-7.5 mg/kg in CMC-Rat-Liver-3d-dn	11/285	2.29E−05	5.56E−05
Ivermectin-7.5 mg/kg in CMC-Rat-Liver-1d-up	10/323	3.28E−04	5.39E−04
Ivermectin-7.5 mg/kg in CMC-Rat-Liver-5d-up	8/311	4.06E−03	5.10E−03
Ivermectin-7.5 mg/kg in CMC-Rat-Liver-3d-up	8/315	4.38E−03	5.46E−03

5 Conclusion

In this chapter, we tried to apply TD based unsupervised FE to various gene expression sets in order to perform drug discovery. The result is quite promising. We are now trying to extend this approach toward wider range of applications.

Acknowledgements The contents of this chapter were supported by KAKENHI, 17K00417, 19H05270, 20H04848, and 20K12067.

References

1. Bacilieri, M., Moro, S.: Ligand-based drug design methodologies in drug discovery process: an overview. Curr. Drug Discovery Technol. **3**(3), 155–165 (2006). https://doi.org/10.2174/157016306780136781. https://www.ingentaconnect.com/content/ben/cddt/2006/00000003/00000003/art00001
2. Caly, L., Druce, J.D., Catton, M.G., Jans, D.A., Wagstaff, K.M.: The fda-approved drug ivermectin inhibits the replication of sars-cov-2 in vitro. Antiviral Res. **178** (2020). https://doi.org/10.1016/j.antiviral.2020.104787
3. Corpechot, C., Chazouillères, O., Rousseau, A., Gruyer, A.L., Habersetzer, F., Mathurin, P., Goria, O., Potier, P., Minello, A., Silvain, C., Abergel, A., Debette-Gratien, M., Larrey, D., Roux, O., Bronowicki, J.P., Boursier, J., de Ledinghen, V., Heurgue-Berlot, A., Nguyen-Khac, E., Zoulim, F., Ollivier-Hourmand, I., Zarski, J.P., Nkontchou, G., Lemoinne, S., Humbert, L., Rainteau, D., Lefèvre, G., de Chaisemartin, L., Chollet-Martin, S., Gaouar, F., Admane, F.H., Simon, T., Poupon, R.: A placebo-controlled trial of bezafibrate in primary biliary cholangitis. New England J. Med. **378**(23), 2171–2181 (2018). https://doi.org/10.1056/nejmoa1714519
4. Evans, W.E., Guy, R.K.: Gene expression as a drug discovery tool. Nat. Genet. **36**(3), 214–215 (2004). https://doi.org/10.1038/ng0304-214
5. Ferreira, L.G., Dos Santos, R.N., Oliva, G., Andricopulo, A.D.: Molecular docking and structure-based drug design strategies. Molecules **20**(7), 13384–13421 (2015). https://doi.org/10.3390/molecules200713384
6. Guo, C., Yin, Q., Li, J., Zheng, Y., Chen, K., Lu, J., Wang, J., Lu, W., Li, S., Liu, T., Abudumijiti, H., Zhou, Y., Zhou, Y., Chen, R., Zhang, R., Xia, Y., Wang, F.: Systematic review and meta-analysis: bezafibrate in patients with primary biliary cirrhosis. Drug Des. Dev. Ther. 5407 (2015). https://doi.org/10.2147/dddt.s92041

7. Hazzan, R., Tur-Kaspa, R.: Bezafibrate treatment of primary biliary cirrhosis following incomplete response to ursodeoxycholic acid. J. Clin. Gastroenterol. **44**(5), 371–373 (2010). https://doi.org/10.1097/mcg.0b013e3181c115b3
8. Iwasaki, S., Tsuda, K., Ueta, H., Aono, R., Ono, M., Saibara, T., Maeda, T., Onishi, S.: Bezafibrate may have a beneficial effect in pre-cirrhotic primary biliary cirrhosis. Hepatol. Res. **16**(1), 12–18 (1999). https://doi.org/10.1016/S1386-6346(99)00033-9
9. Kalfaoglu, B., Almeida-Santos, J., Tye, C.A., Satou, Y., Ono, M.: T-cell hyperactivation and paralysis in severe covid-19 infection revealed by single-cell analysis. Front. Immunol. **11**, 2605 (2020). https://doi.org/10.3389/fimmu.2020.589380
10. Kuhlman, B., Bradley, P.: Advances in protein structure prediction and design. Nat. Rev. Molecul. Cell Biol. **20**(11), 681–697 (2019). https://doi.org/10.1038/s41580-019-0163-x
11. Kuleshov, M.V., Jones, M.R., Rouillard, A.D., Fernandez, N.F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S.L., Jagodnik, K.M., Lachmann, A., McDermott, M.G., Monteiro, C.D., Gundersen, G.W., Ma'ayan, A.: Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Res. **44**(W1), W90–W97 (2016). https://doi.org/10.1093/nar/gkw377
12. Kurihara, T., Furukawa, M., Tsuchiya, M., Akimoto, M., Ishiguro, H., Hashimoto, H., Niimi, A., Maeda, A., Shigemoto, M., Yamasha, K.: Effect of bezafibrate in the treatment of primary biliary cirrhosis. Curr. Therapeutic Res. **61**(2), 74–82 (2000). https://doi.org/10.1016/S0011-393X(00)88530-6
13. National Toxicology Program: DrugMatrix (2010). https://ntp.niehs.nih.gov/drugmatrix/index.html
14. Roider, H.G., Pavlova, N., Kirov, I., Slavov, S., Slavov, T., Uzunov, Z., Weiss, B.: Drug2gene: an exhaustive resource to explore effectively the drug-target relation network. BMC Bioinf. **15**(1), 68 (2014). https://doi.org/10.1186/1471-2105-15-68
15. Rudic, J.S., Poropat, G., Krstic, M.N., Bjelakovic, G., Gluud, C.: Bezafibrate for primary biliary cirrhosis. Cochrane Database Syst. Rev. (2012). https://doi.org/10.1002/14651858.cd009145.pub2
16. Taguchi, Y.H.: Identification of candidate drugs using tensor-decomposition-based unsupervised feature extraction in integrated analysis of gene expression between diseases and DrugMatrix datasets. Scient. Rep. **7**(1) (2017). https://doi.org/10.1038/s41598-017-13003-0
17. Taguchi, Y.H.: Drug candidate identification based on gene expression of treated cells using tensor decomposition-based unsupervised feature extraction for large-scale data. BMC Bioinf. **19**(Suppl 13), 388 (2019)
18. Taguchi, Y.h.: Unsupervised Feature Extraction Applied to Bioinformatics, A PCA Based and TD Based Approach. Springer International (2020). https://doi.org/10.1007/978-3-030-22456-1. https://app.dimensions.ai/details/publication/pub.1120509454
19. Taguchi, Y.h., Turki, T.: A new advanced in silico drug discovery method for novel coronavirus (sars-cov-2) with tensor decomposition-based unsupervised feature extraction. PLOS ONE **15**(9), 1–16 (2020). https://doi.org/10.1371/journal.pone.0238907
20. Takeuchi, Y., Ikeda, F., Fujioka, S.i., Takaki, T., Osawa, T., Yasunaka, T., Miyake, Y., Takaki, A., Iwasaki, Y., Kobashi, H., Yamamoto, K., Itoshima, T.: Additive improvement induced by bezafibrate in patients with primary biliary cirrhosis showing refractory response to ursodeoxycholic acid. J. Gastroenterol. Hepatol. **26**(9), 1395–1401 (2011). https://doi.org/10.1111/j.1440-1746.2011.06737.x
21. Yamanishi, Y., Kotera, M., Moriya, Y., Sawada, R., Kanehisa, M., Goto, S.: DINIES: Drug–target interaction network inference engine based on supervised analysis. Nucleic Acids Res. 42(W1), W39–W45 (2014). https://doi.org/10.1093/nar/gku337
22. Yano, K., Kato, H., Morita, S., Furukawa, R.: Long-term administration of bezafibrate on primary biliary cirrhosis. Kanzo **43**(2), 115–120 (2002). https://doi.org/10.2957/kanzo.43.115
23. Yoo, M., Shin, J., Kim, J., Ryall, K.A., Lee, K., Lee, S., Jeon, M., Kang, J., Tan, A.C.: DSigDB: drug signatures database for gene set analysis. Bioinformatics **31**(18), 3069–3071 (2015). https://doi.org/10.1093/bioinformatics/btv313

Challenges of Long Non Coding RNAs in Human Disease Diagnosis and Therapies: Bio-Computational Approaches

Manojit Bhattacharya, Ashish Ranjan Sharma, and Chiranjib Chakraborty

Abstract Long non-coding RNAs (lncRNAs) are the diverse and prevalent classes of cellular transcripts. Currently numerous research advocates that lncRNAs are the fundamental regulatory elements exist at each level of cell and molecular physiology, where their variations are linked with several human diseases. Here we highlighted and summarised the significant roles of lncRNAs in diverse human diseases, their application as biomarkers element and effective databases using the available bioinformatics resources. Subsequently, we also discuss the specific challenges and potential strategies for its clinical applications in light of computational biology. Therefore, it may conclusive that this study of lncRNAs not only enhances the knowledge a novel aspect for public repositories, dedicated resources, and other effective tools for functional analysis of lncRNAs linked human diseases moreover it definitely supports and reveals the wider range of opportunities for innovative treatment in future days.

Keywords Long non-coding RNAs · Human diseases · Computational biology · Biomarkers

1 Introduction

Recently the high-throughput technologies (e.g., NGS) find that the eukaryotic organism able to transcribe 90% of their genomic DNA. However, from these transcripts only 1–2% are capable to encode functional proteins [1, 2]. Whereas, rest all transcribed RNA considered as long non-coding RNAs (lncRNAs) [3]. These

M. Bhattacharya
Department of Zoology, Fakir Mohan University, Vyasa Vihar, Balasore, Odisha 756020, India

A. R. Sharma
Institute for Skeletal Aging & Orthopedic Surgery, Hallym University-Chuncheon Sacred Heart Hospital, Chuncheon-si 24252, Gangwon-do, Republic of Korea

C. Chakraborty (✉)
Department of Biotechnology, School of Life Science and Biotechnology, Adamas University, Barasat-Barrackpore Rd, Kolkata, West Bengal 700126, India

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_8

lncRNAs are characterized as RNAs size more than 200 nucleotides long, which are distinct, reduced sequence conservation and expression. The eukaryotic lncRNA was first identified in the year 1990s within the gene H19 [4]. This H19 gene showing high sequence conservation within the mammals even though the X chromosome inactivation gene 'Xist' identified along with a specific function as lncRNA [5, 6]. In 2001 the human genome was published and directed a superior focus about non-coding RNAs research, that considered to the identification of lncRNAs in substantial level.

The human genome was published in 2001, and it directed to a bigger attention on non-coding RNAs research. Consequently, the identification and characterization of lncRNAs also increase in the substantial number [7]. The synthesis of lncRNA occurs by RNA polymerase II dependent transcription factors and also further processed, for occurrence of splicing and subsequent polyadenylation. Moreover, these are alike to mRNA in structurally so that they were formerly recognised to as 'mRNA-like'by lacing of a stable ORF (open reading frame) [8]. Collectively, without any specified localization (either cytosol or/and nucleus) within the cell these lncRNAs assembled into multiple classes with variable functions [9]. The lncRNAs seems to performed as gene regulatory elements and are able to directly interact or bind with functional mRNAs that could leading to suppression of protein synthesis or mRNA loss of function [10, 11]. In summary the lncRNAs can also bind with proteins, varying their cellular localization, stability, activity, and can also impound them in a desired location [12].

While the application of next generation technologies in sequencing RNA endorsed the documentation of lncRNAs in large numbers with an unparalleled throughput system. But very limited numbers of them have been finally characterized since the structure and functional aspects. So that, the defining of individual lncRNAs functions still in a subjective challenging conditions [13]. Whereas the advanced, diverse computational tools allow researchers to recognise the specified biological roles and potential functions of lncRNAs; that definitely facilitated to highlight the importance of lncRNAs in a large range of diseases and systems function.

The main purpose of this chapter is to focused on certain prevalent bioinformatics resources and computational approaches (public repositories, dedicated resources, and other effective web tools, software) researchers deal with the lncRNAs functional study.

2 Long Non-coding RNAs as Biomarkers

Presently numbers of research has focused on the exploration of lncRNAs as potential biomarkers for several significant human diseases and related events. The increased use of laboratory experiment and computational modelling has assisted the documentation of potential lncRNAs that may reflects as biomarkers for multiple diagnosis and therapeutics purposes [14]. Chen et al. described the lncRNAs are the promising biomarker for cardiovascular disease [15]. Apparently the lncRNAs levels also be recognizable in blood plasma and urine samples of patients [16, 17]. Some lncRNAs

(NRON and MHRT) also known as latent predictive biomarkers of heart failure [18, 63].

Nevertheless, the studies about lncRNAs, owing to the novel investigation in several studies have been already performed on small cohorts, therefore it may need to be confirmed and subsequently validated in bigger cohort studies.

3 Roles of lncRNAs in Different Human Diseases

The lncRNAs are likely as novel targets for therapeutic purposes and has consider as a substance of crucial element not for only the basic molecular biology research, but also aimed for different pharmaceutical companies. If we consider the experiment about the gene modulation; it revealed the potentiality to prevent and reduce cardiac dysfunction and progression of linked abnormalities in heart disease. Therefore thus phenomenon emphasizing the probable impacts of lncRNAs as suitable therapeutic targets [9, 19].

The lncRNAs can play significant role in several biological processes that have concerned with extensive biomedical interest in their likely impacts on numbers of complex diseases (neurological disorders, autoimmune diseases, coronary artery diseases, multiple types of cancers etc.) [20]. Studies has been already reported that dysregulation of lncRNAs within a variety of human diseases and are related with the disease development and subsequent progression. Here, we documented particular lncRNAs linked with different complex human diseases (Table 1).

Moreover, the lncRNAs are become quickly acquired a crucial component for advanced biomedical science research. Growing research evidences in global scale has shown that lncRNAs performed significant roles in many critical biological practices and these are adding a new layer of clinical outputs within the field of complex human diseases. Therefore, we believe that the added mechanistic and functional research design of these resourceful non-coded RNAs will definitely enlarge our scientific understanding of knowhow practices in biological systems; and that could provide novel approaches led to the effective diagnosis and curable treatment of complex human diseases.

4 Functional Analysis of Long Non-coding RNAs (lncRNAs)

Several studies also highlighted the functional interaction among lncRNAs and miRNAs which might have a key role in gene regulated disease and its consequences. Various techniques in which such types of non-coding RNAs can control the activities and several molecular based mechanisms of miRNA-lncRNA interaction have been described [39, 64]. But, evidences from current research trends in cellular biology

Table 1 List of dysregulated lncRNAs in complex human diseases

Sl. No	Disease	lncRNAs	Reference
1	Breast cancer	Zfas1, SRA-1	[21]
2	Prostate cancer	PRNCR1, PTENP1, PCGEM1	[22, 23]
3	Alzheimer's disease	BACE1-AS, BC200	[24, 25]
4	Coronary artery disease;	ANRIL	[26]
5	Fragile X syndrome	FMR4	[27]
6	Psoriasis	PRINS	[28]
7	Bladder cancer	UCA1	[29]
8	Autoimmune thyroid disease	SAS-ZFAT	[30]
9	Myocardial infarction	MIAT	[31]
10	Spinocerebellar ataxia type 8	ATXN8OS	[32]
11	Neuroblastoma	NDM29, ncRAN	[33, 34]
12	Colon cancer	Kcnq1ot1, uc73A	[35, 36]
13	Leukemia and lymphoma	RMRP	[37]
14	Osteosarcoma	LOC285194	[38]

and animal models are completely inadequate and merely a small quantity of mRNA-lncRNA–miRNA axis based regulation and their accurate molecular mechanism have been perfectly studied.

4.1 Public Repositories and Databases

The lncRNAs related first database was developed in 2003 to provide information about the documented potential regulatory roles [40]. Subsequently, several new tools merging with genomic and transcriptomic information that could provide the expression mapping of current genomic annotations of lncRNAs has been developed. To understand and analysing the functions of lncRNAs two main drive also taken in connection of bioinformatics approaches; (1) to conclude the lncRNAs putative function in hypothesis-driven experiments, (2) to recognise the allied functional lncRNAs [41, 65].

Therefore, the implementation of reliable and efficient computational predictions of lncRNAs databases is highly important in together with systematic biological experiments. It will surely accelerate the significant study of lncRNA functions in

conditions. Some widespread, functioning databases are mentioned in following sections and listed in Table 2.

4.2 Noncode

This database collects and integrates lncRNAs data from PubMed and other online resources by text mining. End-users can predict and display their results of functional annotation through the ncFANs web server of protein-encoding potential. It also covers structure, expression, sequence, function, disease consequence of lncRNA, and other biological factors.

In comparison with other lncRNA databases, the NONCODE supplies more information about lncRNA transcripts and the unique annotations [42].

4.3 LncRBase

LncRBase is a web resource annotation database for interpreting the lncRNA functions of its sequences. It have recorded human and mouse lncRNA transcript information, the small ncRNA-lncRNA assemblage and lncRNA subtypes also included. The tissue specific expression of specific lncRNAs also shown by the microarray probes [43].

4.4 LncBook

This large scale annotated database accumulates predicted and functional lncRNAs information which are experimentally verified. It also contains the information on linked diseases, functions, methylation, expressions, mutations, and miRNA interactions based on the prediction by software. Within the LncBook the LncRNAWiki (an integrated database) also developed for the model of combined annotation [44, 45].

4.5 MONOCLdb

This database contains thousands of lncRNAs from the differentially expression profile sequencing of virus-infected mice. These lncRNAs are annotated by multiple techniques (rank-based methods, enrichment methods, and module-based). The correlation matrix score of such expression profiles of lncRNA and allied phenotypic data were also determined as consider the pathogenic associations [46].

Table 2 Computational resource (web server, tools) for analysing of lncRNAs

Sl. No	Bioinformatics tools/web server	Web link	Remarks	Reference
1	lncRNAdb	http://www.lncrnadb.org/	Describing the expression profiles, molecular features and functions of individual lncRNAs	[51]
2	GATExplorer	http://bioinfow.dep.usal.es/xgate/	Web platform which performed as gene loci browser with nucleotide level mappings of nucleotide probes from expression microarrays of ncRNAs	[52]
3	RNAdb	http://research.imb.uq.edu.au/RNAdb	Database updated with ncRNA datasets, and microarray-based expression data	[53]
4	Rfam	http://www.sanger.ac.uk/Software/Rfam	Specialised database comprising information about ncRNAs families and other structured RNA elements	[54]
5	NONCODE	http://www.noncode.org/	Interactive database for ncRNAs (excluding tRNAs and rRNAs)	[55]
6	fRNAdb	https://www.ncrna.org/frnadb	Computational platform for mining/annotating functional RNA candidates from ncRNAs sequences	[56]
7	LncRNADisease	http://www.rnanut.net/lncrnadisease/	Online database having experimentally supported lncRNA-disease associations	[57]
8	Human Body Map lincRNAs	https://genome.ucsc.edu/cgi-bin/hgTrackUi?db=hg19&g=lincRNAs	Web tools for human large intergenic ncRNAs having specific subclasses	[58]
9	ncRNAimprint	http://rnaqueen.sysu.edu.cn/ncRNAimprint	The comprehensive resource center for mammalian imprinted ncRNAs	[59]

(continued)

Table 2 (continued)

Sl. No	Bioinformatics tools/web server	Web link	Remarks	Reference
10	H-InvDB	http://www.h-invitational.jp/	Comprehensive annotation resource of human genes and transcripts	[60]
11	ncFANs	http://www.ebiomed.org/ncFANs/	User-friendly and practical web interface function annotation server for ncRNAs	[61]
12	EVLncRNAs	http://www.bio-bigdata.net/nsdna/	Small scale database for lncRNAs confirmed by low-throughput experiments	[62]

4.6 lncRNome

The wide-ranging human lncRNAs database collects information about the structure, sequence, annotation, genomic variations, networking proteins, and epigenetic modifications of large number of lncRNAs. The annotation profiling is created from published literature and databases search, comprising associated diseases, and disease-associated mapping of lncRNA gene loci [47].

4.7 LncRNASNP

This database primarily categories the information on the SNP loci of human and mice residing the lncRNA gene. The LncRNASNP also integrated the lncRNA transcripts and its expression for cancer mutations. The predicted interactions of lncRNA with miRNAs and related diseases showed the impact of lncRNA structures variations. Moreover, this database also accumulates predicted and experimentally verified lncRNA-disease associations in human [48, 66].

4.8 LncRNADisease

LncRNADisease is the open access, most commonly used lncRNA database that provides a significant outputs of experimental lncRNA-disease associations. Furthermore, it has shown the confidence score for paired lncRNA-disease on the basis of identified experimental information. This database also collets the regulatory networks of lncRNA [49].

4.9 Lnc2Cancer

This database has shown a wide-ranging association of thousands of lncRNAs and several types of human cancers. The specialised archives were formed by text mining within the PubMed. The database contains of drug-resistant, circulating, and prognostic-lncRNAs in relationships of cancers and lncRNAs. Furthermore, it assembles variant, transcription factor miRNAs, and molecular methylation information of lncRNAs in gene regulation [50].

5 Conclusion

From the last few years, the lncRNAs considered a crucial element for performing significant role in overall biological processes as an emergent paradigm of human disease and associated clinical research. Identification and characterization of disease-linked lncRNAs is much more essential for better understanding of molecular interaction and novel therapeutic target development. It has been attracted much attention within the research community and currently known as one of the vital part in biomedical research and development. While existing laboratory based experimental analysis model (in vivo/ in vitro) directly associated with disease-lncRNAs. But, these are fundamentally affected due to the constraint of lesser efficiency, and higher labour cost and time consuming. To overcome these limitations the advanced high-throughput technologies were applied, that could lead to faster growth in the computational application in biomedical research. For accurate identification and functional interpretation of disease linked lncRNAs become quite easier by applying present bioinformatics tools and techniques. The faster increasing of bioinformatics resources (computational models, software, algorithms, tools) and databases in lncRNAs and diseases has develop a favourable platform to facilitating the effective role of lncRNAs in clinical research, and that must be provide a impactful value for further experimental studies.

Conflicts of Interest The authors declare that they have no competing interests.

References

1. Mattick, J.S.: The genetic signatures of noncoding RNAs. PLoS Genet. **5**(4), e1000459 (2009)
2. Consortium, E.P.: Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature **447**(7146), 799 (2007)
3. Mattick, J., Makunin, I.: Non-coding. RNA Hum. Mol. Genet **15**, R17–R29 (2006)
4. Brannan, C.I., et al.: The product of the H19 gene may function as an RNA. Mol. Cell. Biol. **10**(1), 28–36 (1990)
5. Brockdorff, N., et al.: Conservation of position and exclusive expression of mouse Xist from the inactive X chromosome. Nature **351**(6324), 329–331 (1991)

6. Brown, C.J., et al.: A gene from the region of the human X inactivation centre is expressed exclusively from the inactive X chromosome. Nature **349**(6304), 38–44 (1991)
7. Jarroux, J., Morillon, A., Pinskaya, M.: History, discovery, and classification of lncRNAs. Long Non Cod. RNA Biol. **2017**, 1–46 (2017)
8. Erdmann, V.A., et al.: Collection of mRNA-like non-coding RNAs. Nucleic Acids Res. **27**(1), 192–195 (1999)
9. Collins, L., et al.: Regulation of long non-coding RNAs and MicroRNAs in heart disease: insight into mechanisms and therapeutic approaches. Front. Physiol. **11**, 798 (2020)
10. Yoon, J.-H., et al.: LincRNA-p21 suppresses target mRNA translation. Mol. Cell **47**(4), 648–655 (2012)
11. Gong, C., Maquat, L.E.: lncRNAs transactivate STAU1-mediated mRNA decay by duplexing with 3′ UTRs via Alu elements. Nature **470**(7333), 284–288 (2011)
12. Paneru, B., et al.: Crosstalk among lncRNAs, microRNAs and mRNAs in the muscle 'degradome' of rainbow trout. Sci. Rep. **8**(1), 1–15 (2018)
13. De Santa, F., et al.: A large fraction of extragenic RNA pol II transcription sites overlap enhancers. PLoS Biol. **8**(5), e1000384 (2010)
14. Li, Y., et al.: Extracellular vesicles long RNA sequencing reveals abundant mRNA, circRNA, and lncRNA in human blood as potential biomarkers for cancer diagnosis. Clin. Chem. **65**(6), 798–808 (2019)
15. Chen, C., et al.: The roles of long noncoding RNAs in myocardial pathophysiology. Biosci. Rep. **39**(11), BSR20190966 (2019)
16. Zhou, X., et al.: Identification of the long non-coding RNA H19 in plasma as a novel biomarker for diagnosis of gastric cancer. Sci. Rep. **5**(1), 1–10 (2015)
17. Terracciano, D., et al.: Urinary long noncoding RNAs in nonmuscle-invasive bladder cancer: new architects in cancer prognostic biomarkers. Transl. Res. **184**, 108–117 (2017)
18. Xuan, L., et al.: Circulating long non-coding RNA s NRON and MHRT as novel predictive biomarkers of heart failure. J. Cell Mol. Med. **21**(9), 1803–1814 (2017)
19. Di Mauro, V., Barandalla-Sobrados, M., Catalucci, D.: The noncoding-RNA landscape in cardiovascular health and disease. Non-coding RNA Res. **3**(1), 12–19 (2018)
20. Li, J., Xuan, Z., Liu, C.: Long non-coding RNAs and complex human diseases. Int. J. Mol. Sci. **14**(9), 18790–18808 (2013)
21. Askarian-Amiri, M.E., et al.: SNORD-host RNA Zfas1 is a regulator of mammary development and a potential marker for breast cancer. RNA **17**(5), 878–891 (2011)
22. Petrovics, G., et al.: Elevated expression of PCGEM1, a prostate-specific gene with cell growth-promoting function, is associated with high-risk prostate cancer patients. Oncogene **23**(2), 605–611 (2004)
23. Chung, S., et al.: Association of a novel long non-coding RNA in 8q24 with prostate cancer susceptibility. Cancer Sci. **102**(1), 245–252 (2011)
24. Faghihi, M.A., et al.: Expression of a noncoding RNA is elevated in Alzheimer's disease and drives rapid feed-forward regulation of β-secretase. Nat. Med. **14**(7), 723–730 (2008)
25. Mus, E., Hof, P.R., Tiedge, H.: Dendritic BC200 RNA in aging and in Alzheimer's disease. Proc. Natl. Acad. Sci. **104**(25), 10679–10684 (2007)
26. Broadbent, H.M., et al.: Susceptibility to coronary artery disease and diabetes is encoded by distinct, tightly linked SNPs in the ANRIL locus on chromosome 9p. Hum. Mol. Genet. **17**(6), 806–814 (2008)
27. Khalil, A.M., et al.: A novel RNA transcript with antiapoptotic function is silenced in fragile X syndrome. PLoS One **3**(1), e1486 (2008)
28. Sonkoly, E., et al.: Identification and characterization of a novel, psoriasis susceptibility-related noncoding RNA gene PRINS. J. Biolog. Chem. **280**(25), 24159–24167 (2005)
29. Wang, F., et al.: UCA1, a non-protein-coding RNA up-regulated in bladder carcinoma and embryo, influencing cell growth and promoting invasion. FEBS Lett. **582**(13), 1919–1927 (2008)
30. Shirasawa, S., et al.: SNPs in the promoter of a B cell-specific antisense transcript, SAS-ZFAT, determine susceptibility to autoimmune thyroid disease. Hum. Mol. Genet. **13**(19), 2221–2231 (2004)

31. Ishii, N., et al.: Identification of a novel non-coding RNA, MIAT, that confers risk of myocardial infarction. J. Hum. Genet. **51**(12), 1087–1099 (2006)
32. Moseley, M.L., et al.: Bidirectional expression of CUG and CAG expansion transcripts and intranuclear polyglutamine inclusions in spinocerebellar ataxia type 8. Nat. Genet. **38**(7), 758–769 (2006)
33. Zhu, Y., et al.: ncRAN, a newly identified long noncoding RNA, enhances human bladder tumor growth, invasion, and survival. Urology **77**(2), 510. e1–510. e5 (2011)
34. Castelnuovo, M., et al.: An Alu-like RNA promotes cell differentiation and reduces malignancy of human neuroblastoma cells. FASEB J. **24**(10), 4033–4046 (2010)
35. Tanaka, K., et al.: Loss of imprinting of long QT intronic transcript 1 in colorectal cancer. Oncology **60**(3), 268–273 (2001)
36. Calin, G.A., et al.: Ultraconserved regions encoding ncRNAs are altered in human leukemias and carcinomas. Cancer Cell **12**(3), 215–229 (2007)
37. Maida, Y., et al.: An RNA-dependent RNA polymerase formed by TERT and the RMRP RNA. Nature **461**(7261), 230–235 (2009)
38. Pasic, I., et al.: Recurrent focal copy-number changes and loss of heterozygosity implicate two noncoding RNAs and one tumor suppressor gene at chromosome 3q13. 31 in osteosarcoma. Cancer Res. **70**(1), 160–171 (2010)
39. Yoon, J.-H., Abdelmohsen, K., Gorospe, M. (2014). Functional interactions among microRNAs and long noncoding RNAs. In: Seminars in cell & developmental biology. Elsevier (2014)
40. Szymański, M., Erdmann, V.A., Barciszewski, J.: Noncoding regulatory RNAs database. Nucleic Acids Res. **31**(1), 429–431 (2003)
41. Sacco, L.D., Baldassarre, A., Masotti, A.: Bioinformatics tools and novel challenges in long non-coding RNAs (lncRNAs) functional analysis. Int. J. Mol. Sci. **13**(1), 97–114 (2012)
42. Fang, S., et al.: NONCODEV5: a comprehensive annotation database for long non-coding RNAs. Nucleic Acids Res. **46**(D1), D308–D314 (2018)
43. Chakraborty, S., et al. (2014) LncRBase: an enriched resource for lncRNA information. PLoS One **9**(9), e108010 (2014)
44. Ma, L., et al.: LncRNAWiki: harnessing community knowledge in collaborative curation of human long non-coding RNAs. Nucleic Acids Res. **43**(D1), D187–D192 (2015)
45. Ma, L., et al.: LncBook: a curated knowledgebase of human long non-coding RNAs. Nucleic Acids Res. **47**(D1), D128–D134 (2019)
46. Josset, L., et al.: Annotation of long non-coding RNAs expressed in collaborative cross founder mice in response to respiratory virus infection reveals a new class of interferon-stimulated transcripts. RNA Biol. **11**(7), 875–890 (2014)
47. Bhartiya, D., et al. (2013) lncRNome: a comprehensive knowledgebase of human long noncoding RNAs. Database **2013** (2013)
48. Miao, Y.-R., et al.: lncRNASNP2: an updated database of functional SNPs and mutations in human and mouse lncRNAs. Nucleic Acids Res. **46**(D1), D276–D280 (2018)
49. Bao, Z., et al. (2019) LncRNADisease 2.0: an updated database of long non-coding RNA-associated diseases. Nucleic Acids Res. **47**(D1), D1034-D1037 (2019)
50. Gao, Y., et al. (2019) Lnc2Cancer v2. 0: updated database of experimentally supported long non-coding RNAs in human cancers. Nucleic Acids Res. **47**(D1), D1028–D1033 (2019)
51. Amaral, P.P., et al. (2011) lncRNAdb: a reference database for long noncoding RNAs. Nucleic Acids Res. **39**(Database issue), D146–D151 (2011)
52. Risueño, A., et al.: GATExplorer: genomic and transcriptomic explorer; mapping expression probes to gene loci, transcripts, exons and ncRNAs. BMC Bioinf. **11**(1), 1–12 (2010)
53. Pang, K.C., et al. (2007) RNAdb 2.0—an expanded database of mammalian non-coding RNAs. Nucleic Acids Res. **35**(Database issue), D178–D182 (2007)
54. Gardner, P.P., et al. (2009) Rfam: updates to the RNA families database. Nucleic Acids Res. **37**(Database issue), D136–D140 (2009)
55. He, S., et al. (2008) NONCODE v2.0: decoding the non-coding. Nucleic Acids Res. **36**(Database issue), D170–D172 (2008)

56. Mituyama, T., et al. (2009) The functional RNA database 3.0: databases to support mining and annotation of functional RNAs. Nucleic Acids Res. **37**(Database issue), D89–D92 (2009)
57. Chen, G., et al.: LncRNADisease: a database for long-non-coding RNA-associated diseases. Nucleic Acids Res. **41**(D1), D983–D986 (2012)
58. Cabili, M.N., et al.: Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. Genes Dev **25**(18), 1915–1927 (2011)
59. Zhang, Y., et al.: ncRNAimprint: a comprehensive database of mammalian imprinted noncoding RNAs. RNA **16**(10), 1889–1901 (2010)
60. Yamasaki, C., et al. (2008) The H-Invitational Database (H-InvDB), a comprehensive annotation resource for human genes and transcripts. Nucleic Acids Res. **36**(Database issue), D793–D799 (2008)
61. Liao, Q., et al. (2011) ncFANs: a web server for functional annotation of long non-coding RNAs. Nucleic Acids Res. **39**(suppl_2), W118–W124 (2011)
62. Zhou, B., et al.: EVLncRNAs: a manually curated database for long non-coding RNAs validated by low-throughput experiments. Nucleic Acids Res. **46**(D1), D100–D105 (2018)
63. Roy, S.S., Hsu, C.H., Wen, Z.H., Lin, C.S., Chakraborty, C.: A hypothetical relationship between the nuclear reprogramming factors for induced pluripotent stem (iPS) cells generation–bioinformatic and algorithmic approach. Med. Hypotheses **76**(4), 507–511 (2011)
64. Chakraborty, C., Sekhar Roy, S., Hsu, C.H., Wen, Z.H., Lin, C.S.: Network building of proteins in a biochemical pathway: a computational biology related model for target discovery and drug-design. Curr. Bioinform. **5**(4), 290–295 (2010)
65. Chakraborty, C., Roy, S.S., Hsu, M.J., Agoramoorthy, G.: Network analysis of transcription factors for nuclear reprogramming into induced pluripotent stem cell using bioinformatics. Cell Journal (Yakhteh) **15**(4), 332 (2014)
66. Chakraborty, C., Roy, S.S., Hsu, M.J., & Agoramoorthy, G. (2011). Landscape mapping of functional proteins in insulin signal transduction and insulin resistance: a network-based protein-protein interaction analysis. PLoS One **6**(1), e16388

Protein Sequence Classification Using Convolutional Neural Network and Natural Language Processing

Abhishek Pandey and Sanjiban Shekhar Roy

Abstract Classifying protein sequences from biological data has lot of importance in the field of pharmacology. The application of machine learning to find the sequence of amino acids has recently received popularity from various researchers. This chapter proposes a protein sequence classification technique using 1D Convolutional Neural Network (CNN). We also have discussed how NLP algorithms can be used for protein sequencing. We have achieved an accuracy of 85% with the proposed 1D CNN and further improved to 92% after increasing the filter size.

Keywords Protein sequence · CNN · Machine learning · CNN—1D · NLP

1 Introduction

Structural classification has played a very important role to understand the functional relationship of the protein. Moreover, in the human body for various processes of cell organisms, protein plays an important role. In structural bioinformatics, proteins hold a specific position. Structural bioinformatics is used to determine the relationship between protein and its function. Finding the relationship between the protein is based upon the general principle of chemistry and physics. It helps to do further research on protein evolution and biological functions. Therefore, the study of classification and prediction of protein is helpful in medicine formulation [1]. In recent years, researchers have obtained better results in the field of automatic classification of protein sequences and Natural Language Processing (NLP) has received a lot of attention specially for automatic text generation and language analysis. The deep learning applications on NLP have found new breakthroughs. The representation of proteins can also be done with strings of amino acid letters. This way of expressing

A. Pandey · S. S. Roy (✉)
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: s.roy@vit.ac.in

A. Pandey
e-mail: abhishek.pandey@vit.ac.in

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_9

amiono acid letters is a common technique to many NLP methods. In this chapter, we have tried to find the conceptual similarities and differences between proteins and NLP.

Jaakkola et al. [2] have worked on generative and discriminative approaches for the detection of remote protein homologies. They worked on the Markov model for protein classification and prediction. The drawback of this work is that it took more cost for having accuracy and computational calculation of the fisher score. Supervised machine learning algorithms are also used for finding the protein sequence using kernel functions [3]. This technique performs at a low effective cut-off point for homology modelling of proteins. To stabilize and organize the architecture of protein the hydrophobic interaction is widely used as per Kauzmann [4].

In recent years, the protein database has grown exponentially which helps to study more on the protein sequencing and makes a broad range of protein folds available for researchers. A handful of authors [5–8] have presented interesting work on protein prediction and sequence classification. Sequence classification is the fundamental method utilised by scholars to search homology among sequences. This technique helps to classify the known families/classes. Since the protein related data is addressed by a series of characters, some of the well-known classification algorithms like Support Vector Machines (SVM), Naïve Bayes (NB), k-Kearest Neighbour (kNN) and decision trees (DT) are not much help in protein classification.

A handful of other researchers also have presented their work with motif extraction in biological sequences [9–12]. This technique is primarily based on the structure and function of molecules [12]. Some machine learning technologies have been directly used for protein classification problems [13–17]. In the early work, SVM and multilayer perceptrons have been used to construct a single classifier for fold pattern [14] and for improving these works fold recognition ensemble classifiers are prescribed by the researchers [18]. For better sequencing or classifying the protein structure, kernel-based learning was designed by Damoulas [17]. Xia et al. [19] have proposed a SVM and ensemble-based template searching for recognizing the protein folds.

The protein string is like a human language. In human language, we use some strings of letters whereas in protein strings consist of 20 common amino acids (AAs) (except some rare amino acids). Protein domain or motifs are akin to sentences, phrases and words in the human language [18–22]. Protein is not just a sequence of amino acids, it provides complete information like human language. Protein structure is context-dependent and dynamic which is represented by amino-acid sequences. It means that as per information theory, protein structure exists with its sequences [23]. Therefore, it is evident that we can apply natural language processing (NLP) to protein sequencing as well [20, 24–31].

In Fig. 1 we have shown the similarity of human language and protein sequence; a string is divided as per English grammar whereas, in protein sequence, we take some amino acid string.

Figure 2 shows how tokenization happens in the case of human language and protein sequence. Human language is tokenized in two ways: character-based and word-based and protein sequence strings are tokenized in three ways character-based, 2-length and 3 lengths tokenization. This is the first step to the classification of

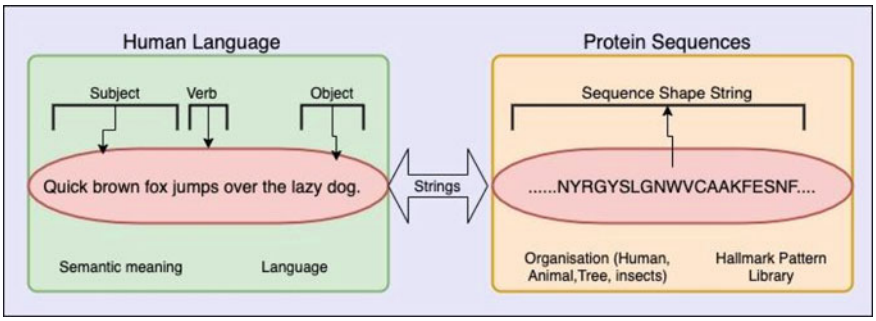

Fig. 1 Human language and Protein sequence both are presented as a string

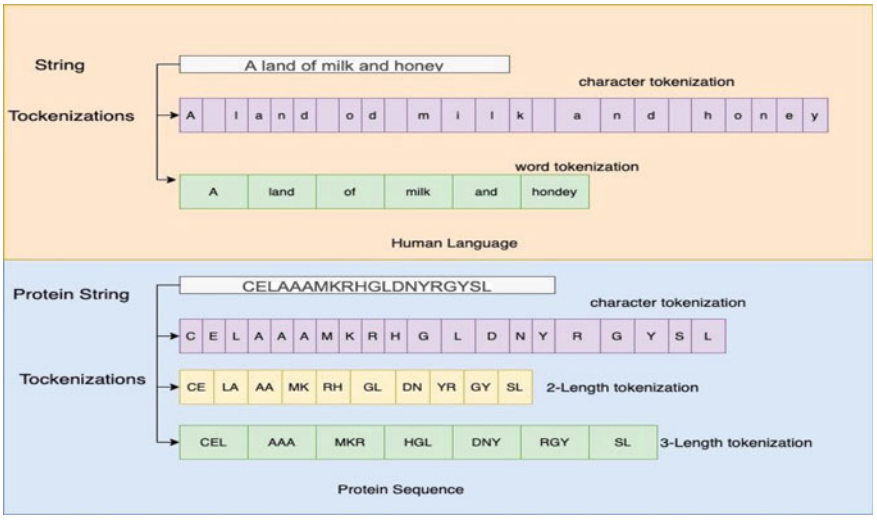

Fig. 2 Tokenization process of human language and protein sequence string

protein sequences. Finding the relationship between protein function and amino acid sequences is a pervasive problem in molecular biology. In this paper, a deep learning algorithm is used to understand the relationship between unaligned amino acid sequences and protein functionality. The deep learning method achieved good success in the field of sound recognition, computer vision and natural language processing [32]. In this chapter, we use deep learning methods for protein family classification; here we have used 1D-Convolution neural network for classification. We have experimented with the Structural Bioinformatics (RCSB), Protein Data Bank (PDB) dataset for the classification problem.

2 Methods and Materials

2.1 Existing Methods

There are some existing approaches for feature construction the Amino Acid Composition, such as N-gram, Active Motifs, and Discriminative Descriptor and the function domain composition. These methods are helpful in protein sequence classification.

2.2 The Amino Acid Composition

It consists of 20 amino acids of proteins which is part of 20 components of protein features. The following formula is used to determine the amino acid composition.

$$comp_j = \frac{k_j}{k} \tag{1}$$

Here comp_j is the amino acid composition and j represents the amino acid for a particular structural class. k which is also the number of amino acid residues of amino acid class type j. Here, j varies from 1 to 20 and K is the total no of residues. In this approach, we count the frequency of each protein in 20 components of protein sequences so that the primary sequence of the information is reduced by considering the amino acid composition alone. The residue amino acid composition is divided into four different classes which are presented in Table 1. In the literature, researchers

Table 1 The calculated value of precision, recall and f1-score

	Precision	Recall	f1-score	Support
Hydrolase	0.88	0.90	0.89	9217
Hydrolase/hydrolase inhibitor	0.79	0.76	0.77	2285
Immune system	0.90	0.95	0.92	3103
Lyase	0.97	0.94	0.95	2285
Transcription	0.97	0.96	0.97	6957
Transferase	0.88	0.88	0.88	1776
Transcription	0.95	0.93	0.94	7254
Transport protein	0.93	0.90	0.91	1674
Viral protein	0.91	0.90	0.91	1694
Virus	0.96	0.96	0.96	1426
Accuracy			0.92	37,671
Macro avg	0.91	0.91	0.91	37,671
Weighted avg	0.92	0.92	0.92	37,671

have proposed methods of using amino acid composition, there are several methods proposed and these methods have achieved good accuracy [33–35].

2.3 N-Grams Method

The N-gram method is the simplest method for defining the length of amino acid strings. It is used to present the continuous string sequences of n items. This method is used for feature extraction in text or speech and this is much useful in NLP for text generation or sentiment analysis [36–38]. If a protein sequence satisfies as a string of amino acids then the N-gram method can be applied for the feature extraction to predict the functionality of protein sequences.

2.4 Active Motifs

The motif is the small amino acid arrangement that is part of the protein sequence family. The active motif is the method to find those strings whose length is greater than the specified length of amino acid. The generalized suffix tree method is used to extract motifs and this method is an advanced version of the suffix tree [39]. The active motif method is used to develop functional recombinants of a protein family.

2.5 Proposed Method and Data Set Collection

In this paper, we have presented a classification method to classify the protein family based upon the sequence of amino acids. Deep learning has become a successful approach with Natural Language Processing (NLP). Convolution neural networks (CNN) perform tremendously in computer vision [7, 40–42] as well as in-text processing.

In this work, we use the Structural Protein Sequences dataset which was retrieved from Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank (PDB). PDB dataset is a repository of protein information that helps to determine the location of every atom related to molecules. It is used in methods such as NMR spectroscopy and cryo-electron microscopy. The website of the PDB dataset is available on the web (https://www.kaggle.com/shahir/protein-data-set). PDB is formed with the different structures of the protein or nucleic acids like ribosomes, oncogenes, drug targets, and even for viruses. This dataset has 400,000, protein structure sequences and also has protein metadata. This protein metadata contains all detailed mechanisms of extraction and classification methods. We have carried out preprocessing techniques such as NULL value removal and unlabeled data removal. In Fig. 3 we have presented all distributions of protein sequence classes or categories.

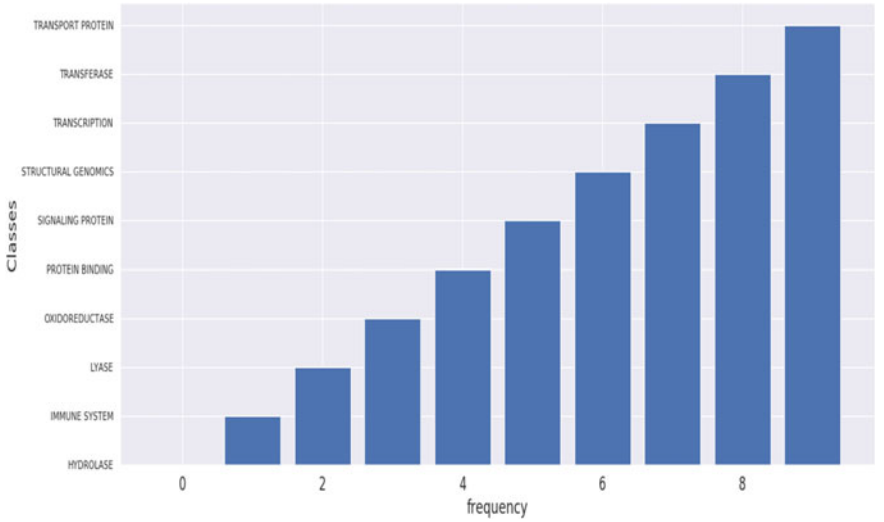

Fig. 3 Distribution of protein family

In Fig. 4 the data partitions of all the protein sequence string into three categories Acid (47.1%), Base (41.1%), and Neutral (11.5%) have been shown.

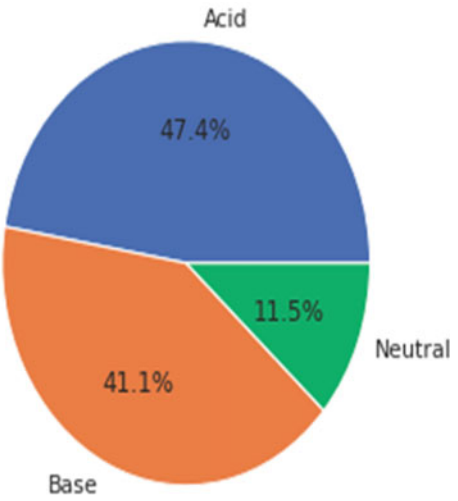

Fig. 4 Acid, base and neutral balance of protein

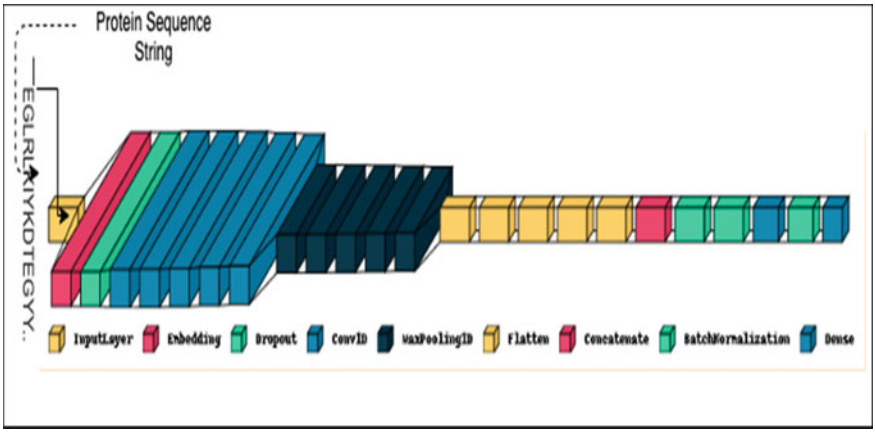

Fig. 5 1-D Convolution Neural Network for protein sequence classification

2.6 Convolution Neural Network

In NLP, CNN is also used for classification, sentiment analysis and topic categorization. Convolution is a mathematical equation or combination of two or more relationships that generate or produce a third or new relationship. CNN-1D has been used for this classification of the protein sequence. In Fig. 5 we have shown a simple CNN architecture.

In this three-layer 1-D convolution neural network, 16,32 and 64 filters are used. In every stage, a rectified linear unit (ReLU) was used as an activation function. Equation (2) is defined as the CNN-1D [43–46] for the classification of protein strings.

$$y(t) = \begin{cases} \sum_{j=0}^{q} p(t-j)m(j), & if \;\; t = q-1 \\ \sum_{j=0}^{q} p(t-j+(c-1))m(j), & otherwise \end{cases} \tag{2}$$

In the above equation, the input length of the convolution layer is t and the input is p. The length of the kernel is h. After every layer of convolution, the kernel window is getting shifted c position, where c is a number of strides. We have used the word embedding technique to the first layer of the proposed model. The 20 amino acid strings are considered as input strings. Further improving the performance we have used pooling and subsequent layers of the convolution process. The first layer has 64 filters with a convolution size of 6 and another layer has 32 filters of size 3.

3 Methods and Materials

In this work, the protein sequence dataset was first classified using the basic 1D-convolution, where the size of the convolution kernel is the same as the size of n-gram and the filter count is the same as the number of words present in the protein sequence. Usually, basic 1D-convolution results in poor accuracy and the validation loss normally is high. To overcome this, we have used the deep architecture model. The proposed model has two layers; the first layer has 6 convolution layers with 64 filters and the second layer has 3 convolutions with 32 filters. The last layer has a softmax activation function and its size is dependent upon the number of classes. In addition, we have used a filter bank instead of a linear chain of convolution that has improved the overall performance by 7%. Figure 6 presents the accuracy of the proposed model and loss. Afterwards, obtained the confusion matrix that has been shown in Fig. 7. The other metric used is F-measure to calculate the model performance.

$$F\text{-}measure = 2 * \left(\frac{precision \cdot recall}{precision + recall} \right) \tag{3}$$

$$precision = \left(\frac{TP}{TP + FP} \right) \tag{4}$$

$$recall = \left(\frac{TP}{TP + FN} \right) \tag{5}$$

$$accuracy = \left(\frac{TP + TN}{TP + FP + FN + TN} \right) \tag{6}$$

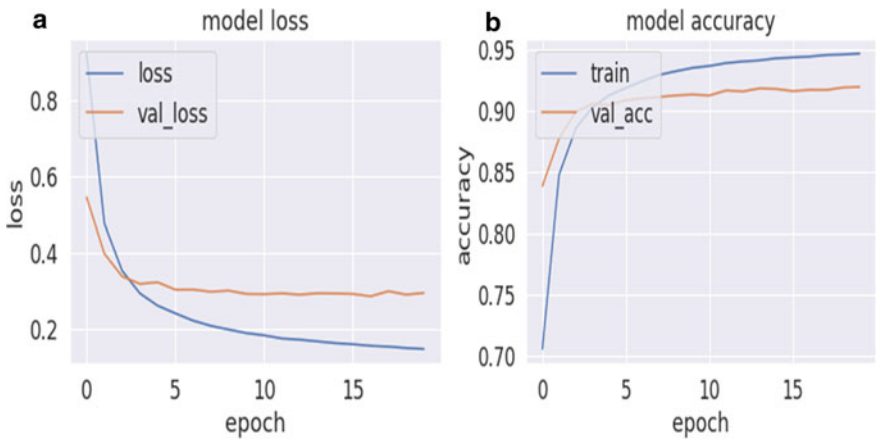

Fig. 6 **a** Model loss of the 1D-CNN Model applied on test set **b** Model accuracy of the1D-CNN model for test test

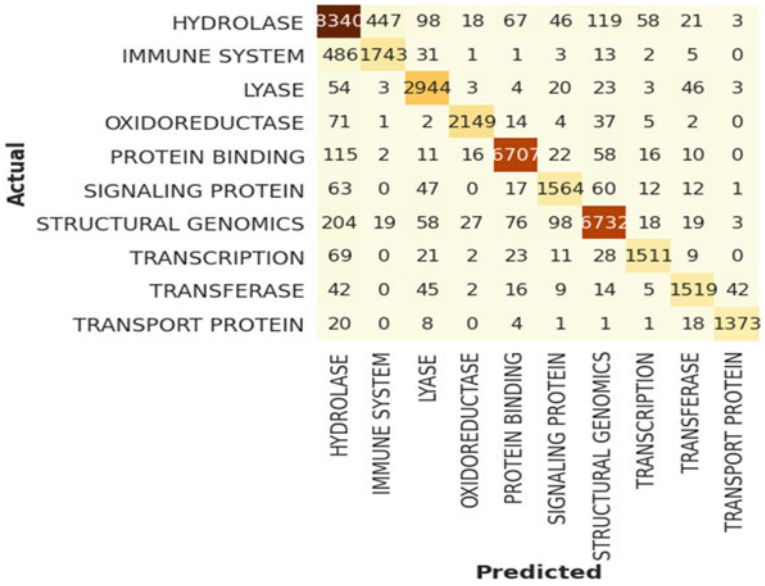

Fig. 7 Confusion matrix of the 1D-CNN model

Here, TP, TN, FP and FN are true-positive, true-negative, false-positive and false-negative respectively. F-measure is shown a performance metric.

Figure 6 represents the outcome of the proposed model in terms of accuracy. As mentioned earlier, we have checked the potential capability of the 1D-CNN model as a proposed model and also as a protein sequence classifier. From Fig. 6a the training and test loss graph can be seen. The test accuracy is around 92%. In Fig. 6b the accuracy graph can be seen in terms of training and test graph plot.

Figure 7 represents the confusion matrix of the classification problem that has 10 classes. Graph is generated by the heatmap function of python seaborn library, where x-axis represents the predicted class and y-axis for the actual class.

In Table 1, the calculated values of precision, recall, and f1-score for all 10 classes have been shown. The outcome shows that VIRUS has maximum precision and recall; on the other side, HYDROLASE/HYDROLASE INHIBITOR found has the lowest precision, recall and f1-the score value. In addition, we also have calculated the accuracy, macro and weighted average score of the model itself.

4 Conclusion

In this paper, we have shown that 1-D CNN can be a potential classifier for protein sequence classification. We have found that the accuracy of the proposed model is 92% on the test data. We have also discussed how the representation of proteins can

be done with strings of amino acid which can be represented as letters and it is related to the NLP method. We have tried to find the conceptual similarities and differences between proteins and natural language processing. The proposed NLP technique has used the embedding method as a first layer of 1D-CNN. Normally, the conventional 1D-CNN algorithm does not provide good accuracy, but in our proposed model we have improved it by enhancing filter size. This has been achieved using a filter bank instead of linear chain convolution.

References

1. Klotz, C., Aumont, M.C., Leger, J.J., Swynghedauw, B.: Human cardiac myosin ATPase and light subunits a comparative study. Biochimica et Biophysica Acta (BBA)-Protein Struct. **386**(2), 461–469 (1995)
2. Jaakkola, T., Diekhans, M., Haussler, D.: A discriminative framework for detecting remote protein homologies. J. Comput. Biol. **7**(1–2), 95–114 (2000)
3. Leslie, C.S., Eskin, E., Cohen, A., Weston, J., Noble, W.S.: Mismatch string kernels for discriminative protein classification. Bioinformatics **20**(4), 467–476 (2004)
4. Kauzmann, W.: Some factors in the interpretation of protein denaturation. Adv. Protein Chem. **14**, 1–63 (1959)
5. White, S.H., Jacobs, R.E.: Statistical distribution of hydrophobic residues along the length of protein chains. Implications for protein folding and evolution. Biophys. J. **57**(4), 911–921 (1990)
6. Roy, S.S., Mittal, D., Basu, A., Abraham, A.: Stock market forecasting using LASSO linear regression model. In: Afro-European Conference for Industrial Advancement, pp. 371–381. Springer, Cham (2015)
7. Roy, S.S., Gupta, A., Sinha, A., Ramesh, R.: Cancer data investigation using variable precision Rough set with flexible classification. In: Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, pp. 472–475 (2012)
8. Xiong, H., et al.: Periodicity of polar and nonpolar amino acids is the major determinant of secondary structure in self-assembling oligomeric peptides. Proceed. Natl. Acad. Sci. **92**(14), 6349–6353 (1995)
9. Liu, H., & Motoda, H. (Eds.).: Feature extraction, construction and selection: a data mining perspective, vol. 453. Springer Science & Business Media (1998)
10. Balas, V.E., Roy, S.S., Sharma, D., Samui, P. (Eds.).: Handbook of Deep Learning Applications, vol. 136. Springer (2019)
11. Roy, S.S., Taguchi, Y.H.: Identification of genes associated with altered gene expression and m6A profiles during hypoxia using tensor decomposition based unsupervised feature extraction. Sci. Rep. **11**(1), 1–18 (2021)
12. Nevill-Manning, C.G., Wu, T.D., Brutlag, D.L.: Highly specific protein sequence motifs for genome analysis. Proc. Natl. Acad. Sci. **95**(11), 5865–5871 (1998)
13. Maddouri, M., Elloumi, M.: Encoding of primary structures of biological macromolecules within a data mining perspective. J. Comput. Sci. Technol. **19**(1), 78–88 (2004)
14. Zhang, Y., Zaki, M.J.: EXMOTIF: efficient structured motif extraction. Algorithms Mol. Biol. **1**(1), 1–18 (2006)
15. Basu, A., Roy, S.S., Abraham, A.: A novel diagnostic approach based on support vector machine with linear kernel for classifying the erythemato-squamous disease. In: 2015 International Conference on Computing Communication Control and Automation, pp. 343–347. IEEE (2015)
16. Roy, S.S., Viswanatham, V.M., Krishna, P.V.: Spam detection using hybrid model of rough set and decorate ensemble. Int. J. Comput. Syst. Eng. **2**(3), 139–147 (2016)

17. Damoulas, T., Girolami, M.A.: Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. Bioinformatics **24**(10), 1264–1270 (2008)
18. Chung, I.F., Huang, C.D., Shen, Y.H., Lin, C.T.: Recognition of structure classification of protein folding by NN and SVM hierarchical learning architecture. In: Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003, pp. 1159–1167. Springer, Berlin, Heidelberg (2003)
19. Xia, J., Peng, Z., Qi, D., Mu, H., Yang, J.: An ensemble approach to protein fold classification by integration of template-based assignment and support vector machine classifier. Bioinformatics **33**(6), 863–870 (2017)
20. Kunik, V., Solan, Z., Edelman, S., Ruppin, E., Horn, D.: Motif extraction and protein classification. In: 2005 IEEE Computational Systems Bioinformatics Conference (CSB'05), pp. 80–85. IEEE (2005)
21. Steinegger, M., Söding, J.: MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat. Biotechnol. **35**(11), 1026–1028 (2017)
22. Strait, B.J., Dewey, T.G.: The Shannon information entropy of protein sequences. Biophys. J. **71**(1), 148–155 (1996)
23. Trifonov, E.N.: The origin of the genetic code and of the earliest oligopeptides. Res. Microbiol. **160**(7), 481–486 (2009)
24. Shannon, C.E.: Prediction and entropy of printed English. Bell Syst. Tech. J. **30**(1), 50–64 (1951)
25. Yu, L., Tanwar, D.K., Penha, E.D.S., Wolf, Y.I., Koonin, E.V., Basu, M.K.: Grammar of protein domain architectures. Proceed. Natl. Acad. Sci. **116**(9), 3636–3645 (2019)
26. Ptitsyn, O.B.: How does protein synthesis give rise to the 3D-structure? FEBS Lett. **285**(2), 176–181 (1991)
27. Samui, P., Kim, D., Jagan, J., Roy, S.S.: Determination of uplift capacity of suction caisson using Gaussian process regression, minimax probability machine regression and extreme learning machine. Iran. J. Sci. Technol. Trans. Civ. Eng. **43**(1), 651–657 (2019)
28. Ofer, D., Linial, M.: ProFET: feature engineering captures high-level protein functions. Bioinformatics **31**(21), 3429–3436 (2015)
29. Roy, S.S., Sikaria, R., Susan, A.: A deep learning based CNN approach on MRI for Alzheimer's disease detection. Intell. Decis. Technol. **13**(4), 495–505 (2019)
30. Roy, S.S., Krishna, P.V., & Yenduri, S.: Analyzing intrusion detection system: an ensemble based stacking approach. In: 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 000307–000309. IEEE (2014)
31. Savojardo, C., Martelli, P. L., Fariselli, P., Casadio, R.: DeepSig: deep learning improves signal peptide detection in proteins. Bioinformatics **34**(10), 1690–1696 (2018)
32. Wen, B., Zeng, W.F., Liao, Y., Shi, Z., Savage, S.R., Jiang, W., Zhang, B.: Deep learning in proteomics. Proteomics **20**(21–22), 1900335 (2020)
33. Eickholt, J., Cheng, J.: Predicting protein residue–residue contacts using deep networks and boosting. Bioinformatics **28**(23), 3066–3072 (2012)
34. Begleiter, R., El-Yaniv, R., Yona, G.: On prediction using variable order Markov models. J. Artifi. Intell. Res. **22**, 385–421 (2004)
35. Gromiha, M.M., Suwa, M.: A simple statistical method for discriminating outer membrane proteins with better accuracy. Bioinformatics **21**(7), 961–968 (2005)
36. Chen, Y., Abraham, A.: Tree-Structure Based Hybrid Computational Intelligence: Theoretical Foundations and Applications, vol. 2. Springer Science & Business Media (2009)
37. Cui, H., Mittal, V., Datar, M.: Comparative experiments on sentiment classification for online product reviews. In: AAAI, vol. 6, no. 30, pp. 1265–1270 (2006)
38. Ghiassi, M., Skinner, J., Zimbra, D.: Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. Expert Syst. Appl. **40**(16), 6266–6282 (2013)
39. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)

40. Hui, L.C.K., Crochemore, M., Galil, Z., Manber, U.: Combinatorial Pattern matching. Lecture Notes in Computer Science in Apostolico, Springer-Verlag **644**, 230–243 (1992)
41. Pandey, A.K., Mishra, S.K.: Transfer Learning-Based Approach for Diabetic Retinopathy Classification using Fundus Images
42. Elujide, I., Fashoto, S.G., Fashoto, B., Mbunge, E., Folorunso, S.O., Olamijuwon, J.O.: Application of deep and machine learning techniques for multi-label classification performance on psychotic disorder diseases. Inf. Med. Unlocked **23**, 100545 (2021)
43. Biswas, R., Vasan, A., Roy, S.S.: Dilated deep neural network for segmentation of retinal blood vessels in fundus images. Iran. J. Sci. Technol. Trans. Electr. Eng. **44**(1), 505–518 (2020)
44. Kim, Y.: Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics. New York University (2014)
45. Srinivasamurthy, R.S.: Understanding 1d convolutional neural networks using multiclass time-varying signals. Doctoral dissertation, Clemson University (2018)
46. Kaestner, K.H., Katz, J., Liu, Y., Drucker, D.J., Schütz, G.: Inactivation of the winged helix transcription factor HNF3α affects glucose homeostasis and islet glucagon gene expression in vivo. Genes Dev. **13**(4), 495–504 (1999)

Machine Learning for Metabolic Networks Modelling: A State-of-the-Art Survey

Marenglen Biba and Narasimha Rao Vajjhala

Abstract This chapter aims to review the machine learning algorithms and models applied for metabolic networks modeling. Metabolic models include structured repositories of information and prediction tools required to support metabolic engineering. This chapter introduces a background overview of various metabolic modeling approaches, including parametric and non-parametric models. In this chapter, we provide an overview of the various machine learning approaches used in metabolic modeling, with a focus on Hidden Markov Models (HMM), Probabilistic Context-Free Grammar (PCFG), and Bayesian Networks (BNs). We then present recent applications of machine learning in the context of metabolic network modeling concluding with a discussion on the limitations of current methods and challenges for future work.

Keywords Metabolic networks · Machine learning · Computational biology · Hidden Markov models · PCFG · Bayesian networks · Flux balance analysis

1 Introduction

Computational modeling techniques help understand biological systems by using mathematical and statistical approaches to analyze experimental data using inbuilt algorithms [1, 2]. Several genome-scale metabolic models have come up over the last few years and help predict phenotypes through constraint-based modeling techniques [3]. Machine learning (ML) is a subset of artificial intelligence (AI), including methods that can learn relationships from data without the need for strong assumptions about the underlying mechanisms [4, 5]. Machine Learning algorithms, along with other technologies, can help process and use large volumes of heterogeneous

M. Biba · N. R. Vajjhala (✉)
University of New York Tirana, Tirana, Albania
e-mail: narasimharao@unyt.edu.al

M. Biba
e-mail: marenglenbiba@unyt.edu.al

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_10

data [6, 7]. Machine learning algorithms first process the input data and then train the underlying model, making predictions on the new test data. Traditional neural networks were not ideal for solving several critical practical problems because of limitations in computational speeds [5, 8]. However, as computation speed is no longer the main problem, the issue now is the accuracy of the prediction, which is the primary motivation for deep learning.

The traditional approaches in metabolic engineering are limited to the 5–15 gene pathway [9]. Hence, full genome-scale engineering is needed for the rigorous biodesign of organisms. However, the traditional trial and error approaches are not feasible as they result in unrealistic and infeasible development times [10]. Machine learning algorithms can address some of these problems and provide reliable predictions with feasible development times. In this chapter, we focus on the state-of-the-art machine learning approaches for metabolic network modeling.

The rest of this chapter is organized as follows: the second section introduces the background theory. The third section explores recent research in the field of metabolic networks and modeling. The fourth section explores the various machine learning machine learning approaches used in metabolic network modeling. The last section provides concluding remarks and some recommendations.

2 Background

There are two main modeling approaches used in metabolic modeling: parametric and non-parametric models [1]. Complex interactions between their building components determine biological systems' behavior. The core issue in biological systems modeling is uncovering and modeling how the biological machinery's function and behavior are implemented through these interactions [11]. The three key foundational principles of metabolic modeling framework models, include reaction stoichiometry, thermodynamics, and kinetics [12]. Traditional neural networks were not ideal for solving several critical practical problems because of limitations in computational speeds. However, as computation speed is no longer the main problem, the issue now is the accuracy of the prediction, which is the primary motivation for deep learning, a machine learning field. In a traditional neural network, all the neutrons are in one layer, represented as the hidden layer. However, instead of placing the neutrons in a single layer in deep learning, the neutrons are placed in several thin layers. In this framework, the inputs go into the first layer, and the output from the first layer serves as the input to the second layer. DL methods have proved successful in supervised learning, and researchers are now focusing on the efficacy of these methods in unsupervised learning. Figure 1 shows part of the aromatic pathway for yeast [11].

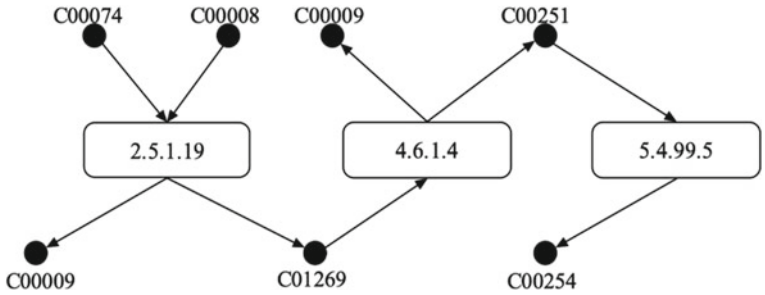

Fig. 1 Part of aromatic pathway for yeast [11]

3 Machine Learning Approaches in Metabolic Modeling

Genome-scale metabolic techniques are increasingly gaining significance, especially approaches such as Flux Balance Analysis (FBA) for predicting metabolic phenotypes [13, 14]. Machine and deep learning techniques are the fundamental tools for inspecting, interpreting, and exploring omic data [15]. Machine learning algorithms allow the creation of optimal solutions for the analysis of new data from previously determined information. Omic data offers a comprehensive approach for analyzing complete genetic profiles, including inter-relationships in contrast to genetics, where the focus is on single genes [15, 16]. Several machine learning algorithms, including SVMs, BNs, and fuzzy logic, have been applied in bioinformatics and metabolism analyses [17].

Machine learning approaches in metabolic networking modeling helped to reduce parameter uncertainty in sampling-based algorithms [12]. False positives are observed when the model falsely predicts an outcome, while false negatives are observed when an experimentally observed outcome is not predicted [18]. Machine learning algorithms are also essential in RNA folding and estimate the impact of mutations on splicing, exploration to gene expression profiles, and the reconstruction of phylogenetic trees [15]. Machine learning algorithms, including neural networks and random forests, improve the pK_a (key parameter in studying the drug molecules' pharmacokinetics) prediction accuracy [12].

Machine learning approaches include both supervised and unsupervised techniques [19]. Supervised machine learning approaches intend to predict one or more targets associated with a given sample [19]. Supervised learning techniques are applied in situations where both the predictors and responses are available. In this approach, machine learning algorithms are trained with labeled data [20]. Some of the supervised learning techniques used in metabolic network modeling include support vector machines (SVM) and artificial neural networks (ANN) [15]. Unsupervised learning approaches can learn only from patterns in the features of the input data [10, 19]. Unsupervised learning methods search for patterns that reduce dimensionality and help in human understanding [9].

Ensemble learning techniques using multiple models with different modeling algorithms or training sets is another approach for modeling metabolic pathways [1]. In this case, the aggregate of prediction results into one prediction is used for predicting the outcome of a pathway. For instance, several decision trees can be used in an ensemble random forest algorithm to improve predictive performance [9]. Machine learning methods' performance depends heavily on data representation, representing a critical difference between classic machine learning and deep learning [13, 21]. Conventional machine learning approaches use manual feature engineering to highlight the weaknesses of the current learning algorithms [22]. In contrast, features are learned automatically with multiple levels of representation in deep learning [23]. These semantic labels are assumed to be mutually exclusive and the feature learning methods do not capture the complexity of these semantic labels. However, in real-world applications there several thousands of semantic categories because of which representing the semantic relations between these categories is quite complex.

3.1 Hidden Markov Models (HMMs)

HMMs are commonly used for modeling time series data [24]. HMM involves a dynamic Bayesian network with a simple structure used in applications in several domains, including image processing, speech recognition, pattern-finding in computational biology, and natural language processing [25]. Most of the features are preselected based on domain knowledge in the HMM applications, and the procedure for feature selection is completely omitted [24]. The probability mass function (PMF) of X_t, is shown in Eq. 1 [26].

$$p_i(x) = \Pr(X_t = x | H_t = i) \text{ for } i = 1, 2, \ldots, m. \tag{1}$$

HMMs are commonly used to estimate difficult and unobservable variables based on the observed variables [25]. The observable variables are known as observation variables, while the unknown variables are known as hidden variables. An HMM's observations are random variables conditioned by a hidden state or a cluster [27]. Goetz et al. [28] proposed a statistical mechanism for Epithelial-to-Mesenchymal transition (EMT) by fitting an HMM of EMT with experimental data. Still, some parameters, for instance, in the case of transition among disease states, such as in the case of Johne's disease, are complex [29]. Researchers have also applied HMMs to identify multimorbidity patterns by integrating dynamic Bayesian networks with a temporal sequence of the observed patient data [27]. Profile HMMs are commonly used for defining and searching protein families and sequences. As the architecture of an HMM is similar to the multiple sequence alignment (MSA), HMMs are suitable for modeling single-point mutations and supporting insertions and deletions [30]. The graph structure of an HMM is shown in Fig. 2.

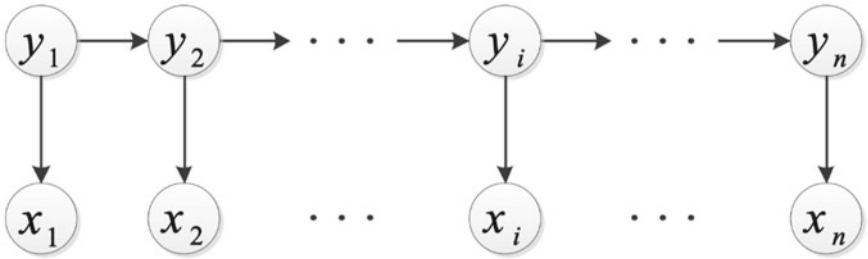

Fig. 2 Graph structure of an HMM, adapted from Zeng [25]

The HMM model has two assumptions: the homogeneous Markov chain hypothesis and the hypothesis of observational independence as shown in Eqs. 2 and 3 [31].

$$p_{ij} = P\left(i_{t+1} = s_j | i_t = s_i\right) \tag{2}$$

where $i_t = s_i$ is the hidden state at time t and $i_{t+1} = s_j$ is the hidden state at time t + 1.

$$q_j(v_k) = \mathrm{P}\left(o_t = v_k | i_t = s_j\right) \tag{3}$$

where $i_t = s_j$ is the hidden state at time t and the observed state is $o_t = v_k$.

Hidden Semi-Markov Models (HSMMs) are also commonly used in metabolic modeling. HSSMs have a similar structure as HMM. The syntactic analysis of sequences is represented as parse trees showing the hierarchical application of grammar rules [30]. PCFG is used in RNA modeling for secondary structure prediction. PCFG is also used for protein modeling, where the parse trees' shape corresponded to the spatial protein structures [30].

3.2 Probabilistic Context-Free Grammar (PCFGs)

PCFG is a Markov grammar where the production probabilities are estimated by decomposing the joint likelihood of categories on the right-hand side into a product of conditionals making a Markov assumption [32, 33]. PCFGs can describe infinite languages that regular grammars, and HMMs cannot describe [34]. PCFG model is extensively used for natural language parsing [33, 35]. PCFGs can describe hierarchical, tree-shaped structures, and these formalisms are used in several domains including statistical language parsing, stochastic language modeling, and in computational biology [36, 37]. PCFGs have been extensively applied in molecular biology, bioinformatics, computer vision, and robotics [34]. PCFG assigns the probability of

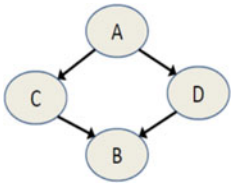

Fig. 3 Graph structure of a sample Bayesian network Adopted from Biba [39]

one to the set of all finite parse trees that it generates [37]. A PCFG is a quintuple (V_N, V_T, S, R, P), where (V_N, V_T, S, R) is a Context-Free Grammar (CFG) [33] and P: R → (0, 1) is a probability function as described in Eq. 4 [35, 37].

$$\forall N \in V_N : \sum_{\alpha : N \to \alpha \in R} P(N \to \alpha) = 1 \tag{4}$$

3.3 Bayesian Networks (BNs)

BNs are flexible probabilistic models using statistics and machine learning to model complex interaction systems [38]. A sample graph structure of a BN is shown in Fig. 3.

Three common approaches used in developing BNs include data-driven, expert-driven, and hybrid BNs. Expert-driven DNs use only knowledge, while data-driven DNs use only the data [40]. Hybrid DNs use both data and knowledge and are used commonly in precision medicine and clinical decision-making [40].

The conditional probability distribution (CPD) for A_i in the graph captures the random variable's conditional probability, given its parents in the graph.

$$\mathrm{P}(A_1, A_2, \ldots, A_n) = \prod_{i=1}^{n} P(A_i | Parents(A_i)) \tag{5}$$

Equation 5 is called the chain rule for Bayesian networks. BNs can capture implicit and explicit relationships between the nodes [41]. The edges in a BN can be directed or undirected [41].

4 Conclusion

The demand for machine learning methods capable of applying and adapting to these big data sets will increase over the next decade. This paper complements the existing literature on machine learning by covering methods, most recent development trends, and applications of machine and deep learning methods in metabolic network modeling. The combination of metabolomics with data-driven machine

learning has a significant potential for assessing computational biology systems' future state. A future research direction would be to focus on the trade-off between model complexity and predictive capability in the context of metabolomic techniques and biological systems. We hope that the issues presented in this chapter will advance the machine learning community's discussion about the next generation of machine learning techniques for metabolic network modeling. This paper also introduced the state-of-the-art machine learning methods and applications in metabolic network modeling. This paper also presented some of the critical challenges related to machine and deep learning techniques and methods.

References

1. Helmy, M., Smith, D., Selvarajoo, K.: Systems biology approaches integrated with artificial intelligence for optimized metabolic engineering. Metab. Eng. Commun. **11**, e00149 (2020)
2. Roy, S.S., Taguchi, Y.H.: Identification of genes associated with altered gene expression and m6A profiles during hypoxia using tensor decomposition based unsupervised feature extraction. Sci. Rep. **11**(1), 8909 (2021)
3. Chen, Y., Nielsen, J.: Mathematical modeling of proteome constraints within metabolism. Curr. Opin. Syst. Biol. **25**, 50–56 (2021)
4. Chopra, C., Sinha, S., Jaroli, S., Shukla, A., Maheshwari, S.: Recurrent neural networks with non-sequential data to predict hospital readmission of diabetic patients. In: Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics [Online]. https://doi.org/10.1145/3155077.3155081
5. Bose, A., Roy, S.S., Balas, V.E., Samui, P.: Deep Learning for Brain Computer Interfaces. In: Balas, V.E., Roy, S.S., Sharma, D., Samui, P. (eds.) Handbook of Deep Learning Applications, pp. 333–344. Springer International Publishing, Cham (2019)
6. Chagas, B.N.R., Viana, J., Reinhold, O., Lobato, F.M.F., Jacob, A.F.L., Alt, R.: A literature review of the current applications of machine learning and their practical implications. Web Intelligence (2405-6456) **18**(1), 69–83 (2020)
7. Agarwal, A., Jayant, A.: Machine learning and natural language processing in supply chain management: a comprehensive review and future research directions. Int. J. Bus. Insights Transform. **13**(1), 3–19 (2019)
8. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
9. Lawson, C.E., et al.: Machine learning for metabolic engineering: a review. Metab. Eng. **63**, 34–60 (2021)
10. Camacho, D.M., Collins, K.M., Powers, R.K., Costello, J.C., Collins, J.J.: Next-generation machine learning for biological networks. Cell **173**(7), 1581–1592 (2018)
11. Biba, M., Ferilli, S., Di Mauro, N., Basile, T.M.A.: A hybrid symbolic-statistical approach to modeling metabolic networks. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 132–139. Heidelberg, Berlin (2007)
12. Suthers, P.F., Foster, C.J., Sarkar, D., Wang, L., Maranas, C.D.: Recent advances in constraint and machine learning-based metabolic modeling by leveraging stoichiometric balances, thermodynamic feasibility and kinetic law formalisms. Metab. Eng. **63**, 13–33 (2021)
13. Lewis, J.E., Kemp, M.L.: Integration of machine learning and genome-scale metabolic modeling identifies multi-omics biomarkers for radiation resistance. Nat. Commun. **12**(1), 2700 (2021)
14. Martino, A., Giuliani, A., Todde, V., Bizzarri, M., Rizzi, A.: Metabolic networks classification and knowledge discovery by information granulation. Comput. Biol. Chem. **84**, 107187 (2020)

15. Zampieri, G., Vijayakumar, S., Yaneske, E., Angione, C.: Machine and deep learning meet genome-scale metabolic modeling. PLoS Comput. Biol. **15**(7), e1007084 (2019)
16. Angione, C.: Human systems biology and metabolic modelling: a review—from disease metabolism to precision medicine. BioMed. Res. Int. **2019**, 8304260 (2019)
17. Cuperlovic-Culf, M.: Machine learning methods for analysis of metabolic data and metabolic pathway modeling. Metabolites **8**(1) (2018)
18. Vijayakumar, S., Conway, M., Lió, P., Angione, C.: Seeing the wood for the trees: a forest of methods for optimization and omic-network integration in metabolic modelling. Brief. Bioinform. **19**(6), 1218–1235 (2018)
19. Plaimas, K., et al.: Machine learning based analyses on metabolic networks supports high-throughput knockout screens. BMC Syst. Biol. **2**, 67–67 (2008)
20. Ahmad, A., et al.: A systematic literature review on using machine learning algorithms for software requirements identification on stack overflow. Secur. Commun. Netw. 1–19 (2020)
21. Skënduli, M.P., Biba, M., Ceci, M.: Implementing scalable machine learning algorithms for mining big data: a state-of-the-art survey. In: Roy, S.S., Samui, P., Deo, R., Ntalampiras, S. (eds.) Big Data in Engineering Applications, pp. 65–81. Springer, Singapore (2018)
22. Panigrahi, A., Patra, M.R.: Chapter 6—Network intrusion detection model based on fuzzy-rough classifiers. In: Samui, P., Sekhar, S., Balas, V.E. (eds.) Handbook of Neural Computation, pp. 109–125. Academic Press (2017)
23. Mitra, S., Roy, S.S., Srinivasan, K.: 6—Classifying CT scan images based on contrast material and age of a person: ConvNets approach. In: Lee, K.C., Roy, S.S., Samui, P., Kumar, V. (eds.) Data Analytics in Biomedical Engineering and Healthcare, pp. 105–118. Academic Press (2021)
24. Cárdenas-Ovando, R.A., Fernández-Figueroa, E.A., Rueda-Zárate, H.A., Noguez, J., Rangel-Escareño, C.: A feature selection strategy for gene expression time series experiments with hidden Markov models. PLOS One **14**(10), e0223183 (2019)
25. Zeng, Y.: Evaluation of physical education teaching quality in colleges based on the hybrid technology of data mining and hidden Markov model. Int. J. Emerg. Technol. Learn. (iJET) **15**(01) (2020)
26. George, S., Jose, A.: Generalized Poisson hidden Markov Model for overdispersed or underdispersed count data. Revista Colombiana de Estadística **43**, 71–82 (2020)
27. Violán, C., et al.: Five-year trajectories of multimorbidity patterns in an elderly Mediterranean population using Hidden Markov Models. Sci. Rep. **10**(1), 16879 (2020)
28. Goetz, H., Melendez-Alvarez, J.R., Chen, L., Tian, X.-J.: A plausible accelerating function of intermediate states in cancer metastasis. PLOS Comput. Biol. **16**(3), e1007682 (2020)
29. Ceres, K.M., Schukken, Y.H., Gröhn, Y.T.: Characterizing infectious disease progression through discrete states using hidden Markov models. PLoS One **15**(11), e0242683 (2020)
30. Dyrka, W., Pyzik, M., Coste, F., Talibart, H.: Estimating probabilistic context-free grammars for proteins using contact map constraints. Peer J. **7**, e6559–e6559 (2019)
31. Liao, Y., Zhao, G., Wang, J., Li, S.: Network security situation assessment model based on extended hidden Markov. Math. Probl. Eng. **2020**, 1428056 (2020)
32. Roark, B., Bacchiani, M.: Supervised and unsupervised PCFG adaptation to novel domains. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1 [Online]. https://doi.org/10.3115/1073445.1073472
33. Mohri, M., Roark, B., Probabilistic context-free grammar induction based on structural zeros. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics [Online]. https://doi.org/10.3115/1220835.1220875
34. Lioutikov, R., Maeda, G., Veiga, F., Kersting, K., Peters, J.: Learning attribute grammars for movement primitive sequencing. Int. J. Robot. Res. **39**(1), 21–38 (2019)
35. Huang, L., Peng, Y., Wang, H., Wu, Z.: PCFG parsing for restricted classical Chinese texts. In: Proceedings of the First SIGHAN Workshop on Chinese Language Processing, vol. 18 [Online]. https://doi.org/10.3115/1118824.1118830

36. Corazza, A., Satta, G.: Cross-entropy and estimation of probabilistic context-free grammars. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics [Online]. https://doi.org/10.3115/1220835.1220878
37. Nederhof, M.-J., Satta, G.: Estimation of consistent probabilistic context-free grammars. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics [Online]. https://doi.org/10.3115/1220835.1220879
38. Becker, A.-K., et al.: From heterogeneous healthcare data to disease-specific biomarker networks: a hierarchical Bayesian network approach. PLoS Comput. Biol. **17**(2), e1008735 (2021)
39. Biba, M.: Integrating Logic and Probability: Algorithmic Improvements in Markov Logic Networks. University of Bari, Bari, Italy (2009)
40. McLachlan, S., Dube, K., Hitman, G.A., Fenton, N.E., Kyrimi, E.: Bayesian networks in healthcare: distribution by medical condition. Artif. Intell. Med. **107**, 101912 (2020)
41. Sazal, M., Mathee, K., Ruiz-Perez, D., Cickovski, T., Narasimhan, G.: Inferring directional relationships in microbial communities using signed Bayesian networks. BMC Genom. **21**(Suppl 6), 663–663 (2020)

Sincle Cell RNA-seq Analysis Using Tensor Decomposition and Principal Component Analysis Based Unsupervised Feature Extraction

Y.-H. Taguchi

Abstract In this chapter, we have applied principal component analysis and tensor decomposition based unsupervised feature extraction to single cell RNA-seq data. The first RNA-seq data is the midbrain development of human and mouse and the second one is that of neurodegenerative diseases. For both cases, genes selected are enriched by various biological terms. Thus, the proposed methods are also useful for the application to single cell RNA-seq analysis.

1 Introduction

Single cell RNA sequencing (scRNA-seq) is a newly developed promising tool to study molecular biology. First of all, individual cells have their own state inside cells, which can never be figure out with tissue level analysis. In addition to this, heterogeneities between cells are often critical, since they play distinct rolls for a tissue to work properly. Thus, if one would like to understand genetic systems in the view of system analysis, investigations of single cell level measurements are mandatory. In this sense scRNA-seq is useful tool to study genomic science. Because of rapid development of scRNA-seq technology, more and more data set started to come. In contrast to the development of measurement technology, analysis tools remains primitive. This is mainly because of lacking of characterization of individual cells. In the standard bioinformatics analysis, supervised learning is popular, since we often know the basic characterizations of samples, e.g., normal tissues vs tumors, or healthy controls and patients. In this case, the question is rather simple; which factors (genes) progress the different between labeled two classes? On the other hand, in single cell level measurements, individual cells are not labeled; labelling individual cells itself is a part of study. Therefore, most popular (initial) analysis of scRNA-seq data set is visualization, which should be preferably two dimensional. In order to fulfill this need, various tools that can give us two dimensional clustering of cells were

Y.-H. Taguchi (✉)
Department of Physics, Chuo University, Tokyo, Japan
e-mail: tag@granular.com

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_11

developed [3]; using these clustering method, we can get clear clusters of cells by which we can categorize numerous single cells to which scRNA-seq was performed. This strategy, identifying cell clusters prior to downstream analysis, however, can cause various problems. One of such possible problems is uncertainty of spatial relationship between generated clusters; how individual cells are cluster is rather robust, but how individual clusters are arranged in embedding space is not stable at all. In other word, global structure cannot be precisely captured; it is reasonable since there are no reasons that spatial relationship between clusters can be represented in two dimensional space. In this case, embedding clusters into two dimensional space often results in simple projection of high dimensional structure toward two dimensional structure. Second, it is unclear if we can weight all of genes equally in order to define "distances" between individual genes. It might be biologically reasonable to reduce the number of genes to the small number of genes that play critical roles in the considered biological problems. In this sense, feature selection is critical factors to get biologically reasonable clusters.

The problem is that popular feature selections are often supervised one which cannot be applied to scRNA-seq analysis that lacks cell labelling in the initial stage. If the clusters are affected by feature selection, we cannot select genes based upon the labeling based upon generated clusters. Thus, it is better for us to have unsupervised feature selection strategy. Our principal component analysis (PAC) and tensor decomposition (TD) based unsupervised feature extraction (FE) [9] is a suitable strategy to attack this problem. It can generate the features without labeling and can select features based upon generated features. Thus, it is relatively easy to make use of it in order to feature selections in scRNA-seq data analysis.

In this chapter, we will demonstrate how we can make use of PCA and TD based unsupervised FE in order to select features (genes) in scRNA-seq data set analysis.

2 scRNA-seq Analysis of Mice and Human Mid Brain Development

We have applied PCA based unsupervised FE to scRNA-seq of mice and human mid brain development [8]. The outline of PCA based unsupervised FE was as follows. Suppose that $x_{ij} \in \mathbb{R}^{N \times M}$ represents expression of ith gene in jth cell and is normalized as $\sum_i x_{ij} = 0, \sum_i x_{ij}^2 = N$. ℓth PC score attributed to ith gene can be computed as ith element of ℓth eigen vector, $u_{\ell i} \in \mathbb{R}^{N \times N}$, of gram matrix $x_{ii'} = \sum_j x_{ij} x_{i'j} \in \mathbb{R}^{N \times N}$,

$$\sum_{i'} x_{ii'} u_{\ell i'} = \lambda_\ell u_{\ell i} \tag{1}$$

where λ_ℓ is ℓth eigen value. ℓth PC loading attributed to jth cell, $v_{\ell j} \in \mathbb{R}^{M \times M}$, can be given as

$$v_{\ell j} = \sum_i u_{\ell i} x_{ij} \tag{2}$$

One should notice that $v_{\ell j}$ is also a eigen vector of $x_{jj'} = \sum_i x_{ij} x_{ij'} \in \mathbb{R}^{M \times M}$ since

$$\begin{aligned} \sum_{j'} x_{jj'} v_{\ell j'} &= \sum_{j'} \sum_i x_{ij} x_{ij'} \sum_{i''} u_{\ell i''} x_{i'' j'} = \sum_i x_{ij} \sum_{i''} u_{\ell i''} \sum_{j'} x_{ij'} x_{i'' j'} \\ &= \sum_i x_{ij} \sum_{i''} u_{\ell i''} x_{ii''} = \lambda_\ell \sum_i x_{ij} u_{\ell i} = \lambda_\ell v_{\ell j} \end{aligned} \tag{3}$$

After that, using the first a few $u_{\ell i}$s, P-values are attributed to ith gene with assuming $u_{\ell i}$ obeys Gaussin (null hypothesis) using χ^2 distribution,

$$P_i = P_{\chi^2} \left[> \sum_{\ell \in \Omega_\ell} \left(\frac{u_{\ell i}}{\sigma_\ell} \right)^2 \right] \tag{4}$$

where $P_{\chi^2}[> x]$ is cumulative χ^2 distribution, Ω_ℓ is a set of selected ℓs and σ_ℓ is standard deviation. P-values are corrected using BH criterion [9] and is associated with adjusted P-values less than 0.01 are selected.

PCA based unsupervised FE was applied to mice and human scRNA-seq data of midbrain development. mRNA expression profiles used were $x_{ij} \in \mathbb{R}^{19531 \times 1977}$ for human and $x_{ij} \in \mathbb{R}^{24378 \times 1907}$ for mouse, respectively (Table 1). We employed $\Omega_\ell = \{1, 2\}$ for human and $\Omega_\ell = \{1, 2, 3\}$ for mice in Eq. (4); 116 genes for human and 118 genes for mice associated with adjusted P-values less than 0.01 were selected. It is interesting that there are 53 genes commonly selected between human and mice. Considering the number of genes as large as $\sim 2 \times 10^4$, the number of commonly selected genes, 53, between 116 human genes and 118 mouse genes is highly significant. The question is if 53 genes are commonly selected because of biological reasons? Even if the size of overlap is significant, if it is because of abiological reasons, it is meaningless. In order to see if genes selected commonly are biological, we evaluated them by enrichment analysis using Enrichr [5]. Table 2 shows the significant terms associated with significant (less than 0.05) adjusted P-vales (five top ranked terms) in "MGI Mammalian Phenotype 2017" category of Enrichr. Among the top five ranked terms, four were brain-related terms. As all terms described abnormal morphology, this is an expected result since fetal gene expression is often distinct from adults that lack fetal-specific gene expression.

Tables 3 and 4 show the enrichment terms in "Allen Brain Atlas down". This also suggests the significance; genes in adults should be depressed in fatal tissues. This suggests, indirectly, genes expressed in adult brains are not expressed in fetal brains, commonly in human and mice. Tables 5 and 6 also show the suppression of adult brain genes. These everything suggest the genes commonly selected human and mice genes are biologically meaningful. Although all of those above is suppression, as can be seen in Table 7, human and mouse embryonic brain genes are also expressed as

Table 1 The number of cells analysed

Human	Weeks	6	7	8	9	10	11	Total	
	Cells	287	131	331	322	509	397	1977	
Mouse	Days	E11.5	E12.5	E13.5	E14.5	E15.5	E18.5	Unknown	Total
	Cells	349	350	345	308	356	142	57	1907

Table 2 Enrichment analysis by Enrichr, "MGI Mammalian Phenotype 2017," of 118 selected genes in mice (Top 5 ranked terms)

Term	Overlap	P-value	Adjusted P-value
MP:0000788_abnormal_cerebral_cortex_morphology	7/145	2.45×10^{-5}	4.55×10^{-5}
MP:0003651_abnormal_axon_extension	5/48	9.18×10^{-6}	4.55×10^{-3}
MP:0000812_abnormal_dentate_gyrus_morphology	5/58	$2.34 \times 10-5$	4.55×10^{-3}
MP:0000807_abnormal_hippocampus_morphology	5/86	1.56×10^{-4}	2.04×10^{-2}
MP:0000819_abnormal_olfactory_bulb_morphology	4/48	1.83×10^{-4}	2.04×10^{-2}

Table 3 Enrichment analysis by Enrichr, "Allen Brain Atlas down," of 116 selected genes in humans (Top 5 ranked terms)

Term	Overlap	P-value	Adjusted P-value
Periventricular stratum of cerebellar vermis	18/300	1.33×10^{-13}	3.72×10^{-11}
Simple lobule	18/300	1.33×10^{-13}	3.72×10^{-11}
Simple lobule, molecular layer	18/300	1.33×10^{-13}	3.72×10^{-11}
Simple lobule, granular layer	18/300	1.33×10^{-13}	3.72×10^{-11}
White matter of cerebellar vermis	18/300	1.33×10^{-13}	3.72×10^{-11}

Table 4 Enrichment analysis by Enrichr, "Allen Brain Atlas down," of 118 selected genes in mice (Top 5 ranked terms)

Term	Overlap	P-value	Adjusted P-value
Pyramus (VIII), granular layer	18/300	1.81×10^{-13}	4.66×10^{-11}
Pyramus (VIII)	18/300	1.81×10^{-13}	4.66×10^{-11}
Pyramus (VIII), molecular layer	18/300	1.81×10^{-13}	4.66×10^{-11}
Paraflocculus, molecular layer	18/300	1.81×10^{-13}	4.66×10^{-11}
Cerebellar cortex	18/300	1.81×10^{-13}	4.66×10^{-11}

Table 5 Enrichment analysis by Enrichr, "GTEx Tissue Sample Gene Expression Profiles down", of 116 selected genes in humans (Top 5 ranked terms)

Term	Overlap	P-value	Adjusted P-value
GTEX-Q2AG-0011-R10A-SM-2HMLA_brain_female_40-49_years	51/1467	1.47×10^{-27}	3.29×10^{-24}
GTEX-TSE9-3026-SM-3DB76_brain_female_60-69_years	49/1384	1.06×10^{-26}	1.19×10^{-23}
GTEX-S7SE-0011-R10A-SM-2XCDF_brain_male_50-59_years	44/1278	3.20×10^{-23}	1.43×10^{-20}
GTEX-QMR6-1426-SM-32PLA_brain_male_50-59_years	41/1066	2.57×10^{-23}	1.43×10^{-20}
GTEX-RNOR-2326-SM-2TF4I_brain_female_50-59_years	47/1484	2.02×10^{-23}	1.43×10^{-20}

Table 6 Enrichment analysis by Enrichr, "GTEx Tissue Sample Gene Expression Profiles down", of 118 selected genes in mice (Top 5 ranked terms)

Term	Overlap	P-value	Adjusted P-value
GTEX-U8XE-0126-SM-4E3I3_testis_male_30-39_years	15/376	6.13×10^{-9}	3.45×10^{-6}
GTEX-X4XX-0011-R10B-SM-46MWO_brain_male_60-69_years	23/938	5.25×10^{-9}	3.45×10^{-6}
GTEX-U4B1-1526-SM-4DXSL_testis_male_40-49_years	13/282	1.23×10^{-8}	3.71×10^{-6}
GTEX-Q2AG-0011-R10A-SM-2HMLA_brain_female_40-49_years	29/1467	5.11×10^{-9}	3.45×10^{-6}
GTEX-RNOR-2326-SM-2TF4I_brain_female_50-59_years	29/1484	6.62×10^{-9}	3.45×10^{-6}

Table 7 Selected gene enrichment in the embryonic brain of "Jensen TISSUE" by Enrichr

Species	Term	Overlap	P-value	Adjusted P-value
Human	Embryonic_brain	71/4936	2.52×10^{-16}	4.07×10^{-15}
Mouse	Embryonic_brain	75/4936	8.90×10^{-20}	1.06×10^{-18}

expected. Thus, in conclusion, our analysis correctly selected biologically meaningful genes commonly between human and mice. We are also interested in if we could identify regulatory elements. In order to see this, we also investigated "ENCODE and ChEA Consensus TFs from ChIP-X" by Enrichr. Then multiple transcription factors (TFs) were associated with adjusted P-values less than 0.01 (Table 8). Regnetwork server [7] was used to check their inter-regulatory relations (Fig. 1).

Table 8 TF enrichment in "ENCODE and ChEA Consensus TFs from ChIP-X" by Enrichr for human and mouse (Bold TFs are common)

Species	TF
Human	**ATF2**, **BCL3**, BCLAF1, BHLHE40, **BRCA1**, **CEBPB**, **CEBPD**, **CHD1**, **CREB1**, CTCF, **E2F1**, E2F4, **EGR1**, ELF1, ETS1, FLI1, GABPA, **KAT2A**, KLF4, **MAX**, **MYC**, NANOG, **NELFE**, NFYA, NFYB, NR2C2, **PBX3**, **PML**, **RELA**, SALL4, **SIN3A**, SIX5, SOX2, SP1, SPI1, **TAF1**, **TAF7**, **TCF3**, USF2, **YY1**, ZBTB33, **ZMIZ1**
Mouse	**ATF2**, **BCL3**, **BRCA1**, **CEBPB**, **CEBPD**, **CHD1** ,**CREB1**, **E2F1**, **EGR1**, **KAT2A**, KLF, **MAX**, **MYC**, **NELFE**, **PBX3**, **PML**, **RELA**, **SIN3A**, **TAF1**, **TAF7**, **TCF3**,**YY1**, **ZMIZ**

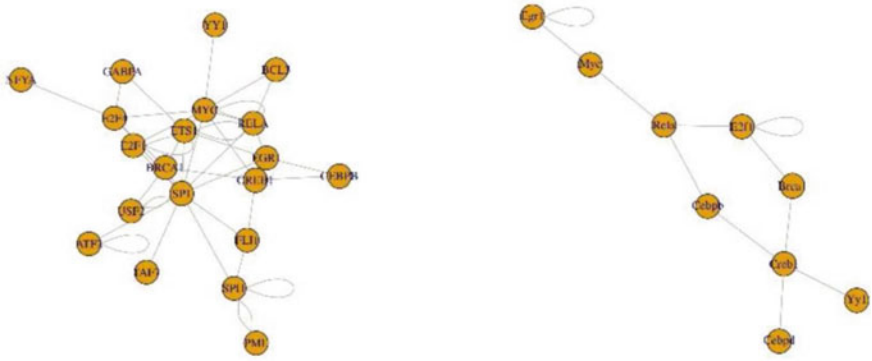

Fig. 1 TF network identified by regnetworkweb for TFs in Table 7 (Left: human, right: mouse)

It was rather obvious that our strategy could identify biologically relevant genes during midbrain development commonly between human and mice. Nevertheless, if other more conventional methods could do this, our strategy is useless. In order to confrim the superitority of our strategy over other conventional methods, we also9 applied a few conventional methods to the present data set. The first method applied is highly variable genes, which is described briefly in the below. We applied the following regression analysis to gene i,

$$\log_{10}\left(\frac{\sigma_i}{\mu_i}\right) = \frac{1}{2}\log_{10}\left(\frac{\beta}{\mu_i} + \alpha\right) + \epsilon_i \tag{5}$$

$$\mu_i = \sum_j \frac{x_{ij}}{M} \tag{6}$$

$$\sigma_i^2 = \sum_j \frac{\left(x_{ij} - \mu_i\right)^2}{M} \tag{7}$$

where α and β are regression coefficients. P-values are attributed to gene i with assuming ϵ_i obeys Gaussian distribution (null hypothesis) as

$$P_i = P_{\chi^2}\left[> \left(\frac{\epsilon_i}{\sigma_\epsilon}\right)^2 \right] \tag{8}$$

where σ_ϵ is the standard deviation. Then P-values are corrected by BH criterion and 168 genes for human and 171 genes for mouse associated with adjusted P-values less than 0.01 were selected. These numbers are similar to those of genes selected by PCA based unsupervised FE and the number of genes commonly selected between these is 44, which is also almost equal to 52, which is a number of genes commonly selected between human an mice, by PCA based unsupervised FE as well. Thus apparently, higher variable genes can achieve almost the same performances that TD based unsupervised FE could achieve.

In order to see if these commonly selected genes by highr variable genes are biologically meaningful, we uploaded them to Enrichr. The outcome is really disappointing. For example, "MGI Mammalian Phenotype 2017" for mouse did not include anything related to the brain, which is inferior to the results in Table 2. On the other hand, no biological terms were significantly enriched in "Allen Brain Atlas down" for human whereas many biological terms were significantly enriched in "Allen Brain Atlas down" for mouse. This inconsistency definitely suggests that apparent consistency of selected gene between human and mouse is abiological. As for "GTEx Tissue Sample Gene Expression Profiles down", the top five ranked terms did not include anything related to brain, but those related to skin and blood. This also suggests the failure of gene selection performed by highly variable genes. No embryonic brain-related terms were found in "Jensen TISSUES", either. In addition to this, when we select TFs that atr supposed to regulate selected genes, only one TF was selected commonly between human and mouse; this is very contrast to that most of TFs selected for mouse is also selected for human in Table 8. This suggests that even if highly variable genes successfully selected genes consistently between human and mouse, the selection is abiological. Possibly, genes were selected based upon some other criterion.

Next criterion of selecting genes are bimodel genes. If gene expression is distinct among multiple classes, distribution of gene expression cannot be unimodal, since unimodal distribution has no ablity to distince multiple classes by definition. Feature selection of bimodal gene attributes P-values to gene based upon the assumption

Table 9 Enrichment analysis by Enrichr, "MGI Mammalian Phenotype 201", of 200 genes selected as bimodal genes for mouse. Top 5 ranked terms

Term	Overlap	P-value	Adjusted P-value
MP:0001262_decreased_body_weight	27/1189	5.68×10^{-5}	4.58×10^{-2}
MP:0011100_preweaning_lethality _complete_penetrance	16/674	1.26×10^{-3}	1.54×10^{-1}
MP:0001265_decreased_body_size	16/774	4.94×10^{-3}	2.31×10^{-1}
MP:0011098_embryonic_lethality_during_organogenesis_complete_penetrance	13/559	4.23×10^{-3}	2.31×10^{-1}

that gene expression is unimodal (null hypothesis). Then obtained P-values are corrected and 11344 human and 10849 mouse genes associated with adjusted P-values less than 0.01 were selected. It is obvious bimodal genes lack the ability to select reasonable (limited) number of genes in contrast to PCA based unsupervised FE and highly variable genes. Although only this face is enough to conclude that bimocal genes are inferior to PCA based unsupervised FE and highly variable genes, we further evaluate gene selections by bimodal gene by selecting only top ranked (i.e., associated with smaller P-values). In oredr that, we selected 200 top ranked genes for human and mouse, respectively. Then we have found that the number of genes commonly selected between human and mouse is as small as 20, which is less than half of common genes selected by PCA based unsupervised FE (53 genes) and highly variable genes (44 genes). This also suggests the inferiority of bimodale genes toward PCA based unsupervised FE and highly variable genes.

In order to further validate selected genes biologically, we uploaded the selected genes to Enrichr. Generally its performance is inferior to PCA based unsupervised FE. When 200 bimodal genes selected for mouse was uploaded to "MGI Mammalian Phenotype 2017" no terms were associated with adjusted P-values less than 0.01 (Table 9). We also checked "Allen Brain Atlas down" category (Tables 10 and 11). Although they are significant, they are less significant than those obtained by PCA based unsupervised FE (Tables 3 and 4). Thus bimodal genes are clearly inferior to PCA based unsupervised FE. In addition, top five ranked terms by "GTEx Tissue Sample Gene Expression Profiles down" included no brain-related terms (Tables 12 and 13) whereas the corresponding Tables 5 and 6 for PCA based unsupervised FE have included many brain related terms. In conclusion, bimodal genes are inferior to PCA based unsupervised FE.

Although bimodal genes are inferior to PCA based unsupervised FE, it is a bit better than highly variable genes. Table 14 shows the enrichment analysis of Embryonic_brain for "Jensen TISSUES" in Enrichr. Their significance is better than PCA based unsupervised FE. Thus bimodal genes are better than highly variable genes that could not detect anything valuable in enrichment analysis.

Table 10 Enrichment analysis by Enrichr, "Allen Brain Atlas down", of 200 genes selected as bimodal genes for human. Top 5 ranked terms

Term	Overlap	P-value	Adjusted P-value
Shell part of the anterobasal nucleus	17/300	9.73×10^{-9}	1.82×10^{-5}
Suboptic nucleus	15/300	3.75×10^{-7}	2.34×10^{-4}
Paraflocculus, granular layer	14/300	2.10×10^{-6}	4.36×10^{-4}
Layer 1 of FCx	15/300	3.75×10^{-7}	2.34×10^{-4}
Medial trapezoid nucleus	14/300	2.10×10^{-6}	4.36×10^{-4}

Table 11 Enrichment analysis by Enrichr, "Allen Brain Atlas down", of 200 genes selected as bimodal genes for mouse. Top 5 ranked terms

Term	Overlap	P-value	Adjusted P-value
Intermediate stratum of r10Ve	15/300	3.75×10^{-7}	2.51×10^{-4}
Rhombomere 11	15/300	3.75×10^{-7}	2.51×10^{-4}
r10 part of spinal vestibular nucleus	15/300	3.75×10^{-7}	2.51×10^{-4}
r11 alar plate	13/300	1.09×10^{-5}	4.36×10^{-3}
Flocculus	13/300	1.09×10^{-5}	4.36×10^{-3}

Table 12 Enrichment analysis by Enrichr, "GTEx Tissue Sample Gene Expression Profiles down", of 200 genes selected as bimodal gene for human. Top 5 ranked terms

Term	Overlap	P-value	Adjusted P-value
GTEX-XBEC-1326-SM-4AT69_heart_male_50-59_years	150/7340	1.75×10^{-28}	1.33×10^{-25}
GTEX-Q734-0426-SM-48TZX_pancreas_female_40-49_years	139/6240	6.10×10^{-29}	1.33×10^{-25}
GTEX-XQ8I-1926-SM-4BOOK_pancreas_male_50-59_years	120/4696	1.50×10^{-28}	1.33×10^{-25}
GTEX-PX3G-1026-SM-48TZW_pancreas_female_20-29_years	105/3830	3.13×10^{-26}	1.79×10^{-23}
GTEX-WRHU-1226-SM-4E3IJ_heart_female_50-59_years	158/8581	1.14×10^{-25}	4.35×10^{-23}

Table 13 Enrichment analysis by Enrichr, "GTEx Tissue Sample Gene Expression Profiles down", of 200 genes selected as bimodal genes for mouse. Top 5 ranked terms

Term	Overlap	P-value	Adjusted P-value
GTEX-XBEC-1326-SM-4AT69_heart_male_50-59_years	147/7340	2.26×10^{-26}	4.98×10^{-23}
GTEX-WRHU-1226-SM-4E3IJ_heart_female_50-59_years	157/8581	5.71×10^{-25}	5.33×10^{-22}
GTEX-Q734-0426-SM-48TZX_pancreas_female_40-49_years	133/6240	7.26×10^{-25}	5.33×10^{-22}
GTEX-PX3G-1026-SM-48TZW_pancreas_female_20-29_years	100/3830	6.15×10^{-23}	3.38×10^{-20}
GTEX-T2IS-0426-SM-32QPE_heart_female_20-29_years	130/6360	3.89×10^{-22}	1.71×10^{-19}

Table 14 Enrichment of selected genes as bimodal genes in Embryonic brain by "Jensen TISSUES" by Enrichr

Species	Term	Overlap	P-value	Adjusted P-value
Human	Embryonic_brain	150/4936	6.42×10^{-51}	8.15×10^{-49}
Mouse	Embryonic_brain	122/4936	8.01×10^{-28}	2.36×10^{-26}

Apparently, bimodal genes are useless. Nevertheless, this impression is reversed if we consider TFs that might regulate genes selected by bimodal genes. In order to see this point, we investigated "ENCODE and ChEA Consensus TFs from ChIP-X" (Table 15). As can be seen, not only several TFs were commonly selected between human and mouse, but also many commonly selected TFs also appeared in Table 8. It is rather suprising, since there are no commonly selected genes between PCA based unsupervised FE and bimodal genes. This suggests, although bimodel genes failed to select biologically meaningful genes, it can selected genes regulated commonly by the some TFs. Anyway, it was obvious that PCA based unsupervised FE was superior to two conventional methods, highly variable genes as well as bimodal genes, used for gene selection in scRNA seq.

Although PCA based unsupervised FE worked well, we would like to apply an advanced method, tensor decomposition (TD) based unsupervised FE, to this scRNA-seq data [11]. In order to apply TD based unsupervised FE to the present data set, we generated a tensor from x_{ij} and x_{ik} as

$$x_{ijk} = x_{ij} x_{ik} \in \mathbb{R}^{N \times M \times K}. \tag{9}$$

A matrix $x_{ik} \in \mathbb{R}^{N \times K}$ was generated as

Table 15 TF enrichment enriched in "ENCODE and ChEA Consensus TFs from ChIP-X" by Enrichr for human and mouse bimodal genes. Bold TFs are common. Genes in red are also commonly selected between human and mouse in Table 8

Species	TF
Human	**ATF2**, **BRCA1**, CEBPD,**CHD1**,**CREB1**, E2F6, ELF1, ETS1, **FLI1**, **FOS**, **GABPA**, **IRF3**, KAT2A, **MAX**, **MYC**, NELFE, **NFYA**, **NFYB**, **NR2C2**,**NRF1**, **PBX3**, **PML**, RELA, RUNX1, SIN3A, **SIX5**, **SP1**, **SP2**, STAT5A, **TAF1**, **TAF7**, **TCF3**, TCF7L2, **USF1**, **USF2**, **YY1**, **ZBTB33**, **ZBTB7A**, **ZMIZ1**, **ZNF384**
Mouse	**ATF2**, BHLHE40, **BRCA1**, **CHD1**, **CREB1**, E2F1, E2F4, E2F6, ELF1, **FLI1**, **FOS**, **GABPA**, **IRF3**, KLF4, **MAX**, **MYC**, **NFYA**, **NFYB**, **NR2C2**, **NRF1**, **PBX3**, **PML**, RCOR1, RUNX1, **SIX5**, **SP1**, **SP2**, SPI1, **TAF1**, **TAF7**, **TCF3**, **TCF7L2**, UBTF, **USF1**, **USF2**, **YY1**, **ZBTB33**, **ZBTB7A**, **ZMIZ1**, **ZNF384**

$$x_{jk} = \sum_{i=1}^{N} x_{ijk} \tag{10}$$

Singular value decomposition (SVD) was applied to x_{jk} and we got

$$x_{jk} = \sum_{\ell=1}^{\min(M,K)} u_{\ell j} \lambda_\ell v_{\ell k} \tag{11}$$

and missing singular vector was obtained as

$$u_{\ell i} = \sum_{j=1}^{M} x_{ij} u_{\ell j} \tag{12}$$

$$v_{\ell i} = \sum_{k=1}^{K} x_{ik} v_{\ell k} \tag{13}$$

P_i^{human}, which is supposed to human mRNA, can be obtained by Eq. (4), and P_i^{mouse}, which is supposed to mouse mRNA, can be obtained as

$$P_i^{\text{mouse}} = P_{\chi^2}\left[> \sum_{\ell \in \Omega_\ell} \left(\frac{v_{\ell i}}{\sigma_\ell} \right)^2 \right] \tag{14}$$

Then human mRNAs, is, and mouse mRNAs, is, are selected adjusted P-values using BH criterion [9].

Table 16 Confusion matrix of coincidence between selected 55 singular value vectors selected among all 1,977 singular value vectors, $u_{\ell j}$, attributed to human cells and 44 singular value vectors selected among all 1907 singular value vectors, $v_{\ell k}$, attributed to mouse cells

		Human	
		Not selected	Selected
Mouse	Not selected	1833	12
	Selected	23	32

Here Ω_ℓ was selected as follows. We would like to find $u_{\ell j}$ and $v_{\ell k}$ with any kind of time dependence. Thus we apply the following categorical regression,

$$u_{\ell j} = a_\ell + \sum_{t=1}^{T} a_{\ell t}\delta_{jt} \tag{15}$$

$$v_{\ell k} = b_\ell + \sum_{t=1}^{T} b_{\ell t}\delta_{kt} \tag{16}$$

where $a_\ell, a_{\ell t}, b_\ell, b_{\ell t}$ are regression constants, $\delta_{jt}(\delta_{kt})$ takes 1 when $j(k)$th cell's expression is measured at tth time points, T is the number of time points. P-values are attributed to ℓs based upon the significance of the above regression analysis. Then P-values were corrected by BH criterion [9]. As a result, 55 and 44 ℓs associated with adjusted P-values less than 0.01 were selected and regarded as Ω_ℓ for human and mouse, respectively.

We have noticed that selected ℓs are largely overlapped between human and mouse. Table 16 shows the confusion matrix; they are highly coincident if we consider that the number of selected ℓs is as small as ~ 10 among $\sim 10^3$ ℓs. Figure 2 shows the actual coincidence of selected ℓs between human and mise. Considering that it is an integrated analysis of completely independent two data sets, it is really remarkable. Next we selected 456 human genes and 505 mouse genes as mentioned in the above using corrected P_is computed. Table 17 shows the confusion matrix; they are highly coincident if we consider that the number of selected genes is as small as $\sim 10^2$ among $\sim 10^4$ genes. Odds ratio is as large as 133 and P-values computed by Fisher's exact test is zero within the numerical accuracy. This high coincidence between human and mouse genes definitely suggests the success of our analysis.

As has been done in PCA based unsupervised FE, biological reliability of selected genes must be evaluated. At first, we checked if genes selected by TD based unsupervised FE are coincident with those selected by PCA based unsupervised FE in the above. Then we found that 102 human genes among 116 human genes selected by PCA based unsupervised FE were also selected by TD based unsupervised FE whereas 91 mouse genes among 118 mouse genes selected by PCA based unsupervised FE were also selected by TD based unsupervised FE. Thus, genes selected by

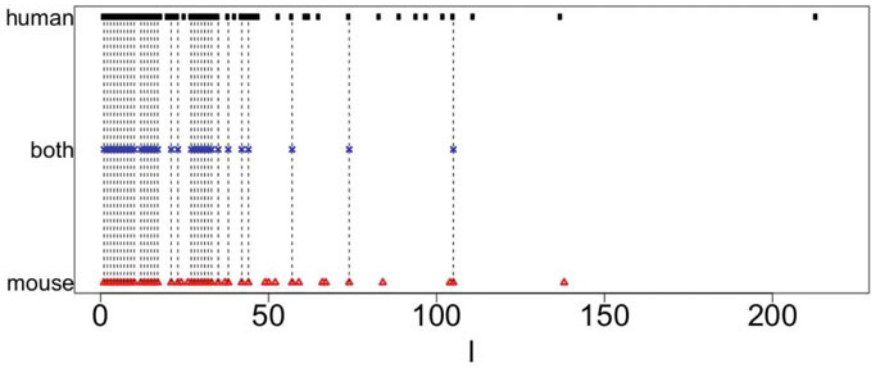

Fig. 2 Coincidence between singular value vectors shown in Table 16. Horizontal axis: singular value vector numbering ℓ. Black open circles, ℓ selected for human; blue crosses, ℓ selected for both human and mouse; red open triangles, ℓ selected for mouse. Vertical black broken lines connect ℓ selected for both human and mouse

Table 17 Confusion matrix of coincidence between selected 456 genes for human and selected 505 genes for mouse among all 13,384 common genes

		Human	
		Not selected	Selected
Mouse	Not selected	133233	151
	Selected	200	305

PCA based unsupervised FE were highly coincident with those selected by TD based unsupervised FE.

Next we tried to validate biological reliability of human and mouse genes selected by TD based unsupervised FE using enrichment analysis; the selected 456 humane genes as well as 505 mouse genes were uploaded to Enrichr. Table 18 shows the top ranked terms in "'Allen Brain Atlas up". In contrast to PCA based unsupervised FE that could identify terms in "Allen Brain Atlas down" (Tables 3 and 4), since TD based unsupervised FE could detect various terms in"Allen Brain Atlas up", it is obvious that TD based unsupervised FE could outperform PCA based unsupervised FE. Table 19 shows the enrichment analysis in "JENSEN TISSUES" category. Their associated P-values are $\sim 10^{-100}$ which are much more significant than those in Tables 7, $P \sim 10^{-10}$. Table 20 also strengthen the reliability of genes selected by TD based unsupervised FE, since the term "MIDBRAIN" is enriched highly in "ARCHS4 Tissues", and it is top ranked for both human and mouse. There are some additional significant enrichment related to brains (Tables 21 and 22), although they are less significant than those above. Finally, we tried to identify regulatory TFs that might regulate genes selected by TD based unsupervised FE (Table 23). Unfortunately, coincidence between mouse and human is less than PCA based unsupervised FE (Table 15) and only one TF was commonly selected between human and mouse in

Table 18 Five top ranked terms from "Allen Brain Atlas up" by Enrichr for selected 456 human genes and 505 mouse genes

Term	Overlap	P-value	Adjusted P-value
Human			
Paraventricular hypothalamic nucleus, magnocellular division, medial magnocellular part	31/301	2.68×10^{-12}	2.91×10^{-9}
Paraventricular hypothalamic nucleus, magnocellular division	31/301	2.68×10^{-12}	2.91×10^{-9}
Paraventricular hypothalamic nucleus, magnocellular division, posterior magnocellular part	28/301	3.39×10^{-10}	1.47×10^{-7}
Paraventricular hypothalamic nucleus	29/301	7.02×10^{-11}	5.08×10^{-8}
paraventricular nucleus, dorsal part	27/301	1.57×10^{-9}	4.88×10^{-7}
Mouse			
Paraventricular hypothalamic nucleus, magnocellular division, medial magnocellular part	31/301	4.03×10^{-11}	2.19×10^{-8}
Paraventricular hypothalamic nucleus, magnocellular division	31/301	4.03×10^{-11}	2.19×10^{-8}
Paraventricular hypothalamic nucleus, magnocellular division, posterior magnocellular part	31/301	4.03×10^{-11}	2.19×10^{-8}
lower dorsal lateral hypothalamic area	29/301	8.40×10^{-10}	3.65×10^{-7}
Paraventricular hypothalamic nucleus, magnocellular division, posterior magnocellular part, lateral zone	31/301	4.03×10^{-11}	2.19×10^{-8}

Table 19 Enrichment of Embryonic brain by "JENSEN TISSUES" in Enrichr

Term	Overlap	P-value	Adjusted P-value
Human			
Embryonic_brain	330/4936	3.36×10^{-104}	4.30×10^{-102}
Mouse			
Embryonic_brain	366/4936	3.59×10^{-115}	4.59×10^{-113}

Table 20 Enrichment of Embryonic brain by "ARCHS4 Tissues" in Enrichr

Term	Overlap	P-value	Adjusted P-value
Human			
MIDBRAIN	248/2316	1.02×10^{-129}	1.11×10^{-127}
Mouse			
MIDBRAIN	248/2316	1.44×10^{-99}	1.56×10^{-97}

Table 21 Five top ranked terms from "GTEx Tissue Sample Gene Expression Profiles up" by Enrichr for selected 456 human genes and 505 mouse genes. Brain related terms are asterisked

Term	Overlap	P-value	Adjusted P-value
Human			
GTEX-QCQG-1426-SM-48U22_ovary_female_50-59_years	105/1165	3.56×10^{-35}	1.04×10^{-31}
GTEX-RWS6-1026-SM-47JXD_ovary_female_60-69_years	116/1574	7.96×10^{-31}	7.74×10^{-28}
GTEX-TMMY-1726-SM-4DXTD_ovary_female_40-49_years	117/1582	2.97×10^{-31}	4.33×10^{-28}
GTEX-RU72-0008-SM-46MV8_skin_female_50-59_years	94/1103	1.99×10^{-29}	1.45×10^{-26}
GTEX-R55E-0008-SM-48FCG_skin_male_20-29_years	111/1599	3.67×10^{-27}	1.78×10^{-24}
Mouse			
*GTEX-WVLH-0011-R4A-SM-3MJFS_brain_male_50-59_years	139/1957	1.93×10^{-30}	5.63×10^{-27}
*GTEX-X261-0011-R8A-SM-4E3I5_brain_male_50-59_years	135/1878	5.24×10^{-30}	7.65×10^{-27}
*GTEX-T5JC-0011-R4A-SM-32PLT_brain_male_20-29_years	129/1948	3.51×10^{-25}	3.42×10^{-22}
GTEX-R55E-0008-SM-48FCG_skin_male_20-29_years	109/1599	4.93×10^{-22}	2.40×10^{-19}
GTEX-TMMY-1726-SM-4DXTD_ovary_female_40-49_years	107/1582	2.37×10^{-21}	7.69×10^{-19}

Table 22 Five top ranked terms from "MGI Mammalian Phenotype 2017" by Enrichr for selected 456 human genes and 505 mouse genes. Brain related terms are asterisked

Term	Overlap	P-value	Adjusted P-value
Human			
MP:0002169_no_abnormal_phenotype_detected	82/1674	2.52×10^{-11}	5.53×10^{-8}
MP:0001262_decreased_body_weight	63/1189	3.40×10^{-10}	3.72×10^{-7}
MP:0001265_decreased_body_size	46/774	3.20×10^{-9}	2.33×10^{-6}
*MP:0009937_abnormal_neuron_differentiation	15/106	1.81×10^{-8}	9.90×10^{-6}
*MP:0000788_abnormal_cerebral_cortex_morphology	17/145	3.64×10^{-8}	1.60×10^{-5}
Mouse			
MP:0002169_no_abnormal_phenotype_detected	89/1674	$1.36 \times s10^{-11}$	3.09×10^{-8}
MP:0011091_prenatal_lethality,_complete_penetrance	27/272	1.68×10^{-9}	1.91×10^{-6}
MP:0001262_decreased_body_weight	65/1189	3.93×10^{-9}	2.97×10^{-6}
MP:0011100_preweaning_lethality,_complete_penetrance	42/674	8.55×10^{-8}	3.88×10^{-5}
MP:0001265_decreased_body_size	46/774	8.22×10^{-8}	3.88×10^{-5}

Table 23 TFs enriched in "ENCODE and ChEA Consensus TFs from ChIP-X" by Enrichr for human and mouse. Bold TFs are common. Genes in red are also commonly selected between human and mouse in Table 8

human	BCL3, **BHLHE40**, **EGR1**, **GABPA**, **IRF3**, **PPARG**, **REST**, **RFX5**, SP1, SP2, SRF, **STAT3**, **TCF7L2**, TRIM28, TRIM28, **ZBTB33**,
mouse	**BHLHE40**, CTCF, E2F4, E2F6, **EGR1**, ESR1, ETS1, FLI1, **GABPA**, **IRF3**, NFIC, NRF1, **PPARG**, RCOR1, **REST**, **RFX5**, SPI1, **STAT3**, **TCF7L2**, USF1, USF2, YY1, **ZBTB33**, ZNF384,

both PCA and TD based unsupervised FE. This is possibly because TD based unsupervised FE identified more genes than PCA based unsupervised FE. We need more detailed investigation to judge which is more biologically reliable. Nevertheless, their are more inter-connected (Fig. 3) than Fig. 1.

In conclusion, TD based unsupervised FE can be an advanced method that can successfully integrate two independent scRNA-seq data sets.

3 scRNA-seq Analysis of Neurodegenerative Disease

Although we could successfully integrate two independent scRNA-seq data sets in the previous section, if it can be used for application study, it is much better. In this section, we try to apply TD based unsupervised FE to scRNA-seq data set with aiming drug discovery [10]. In this study, we deal with data set of gene expression profiles of two mouse brain parts with aging progression (four time points, (3, 6, 12, and 21 weeks after birth)). Individual single cells is in one of 96 cells of four 96

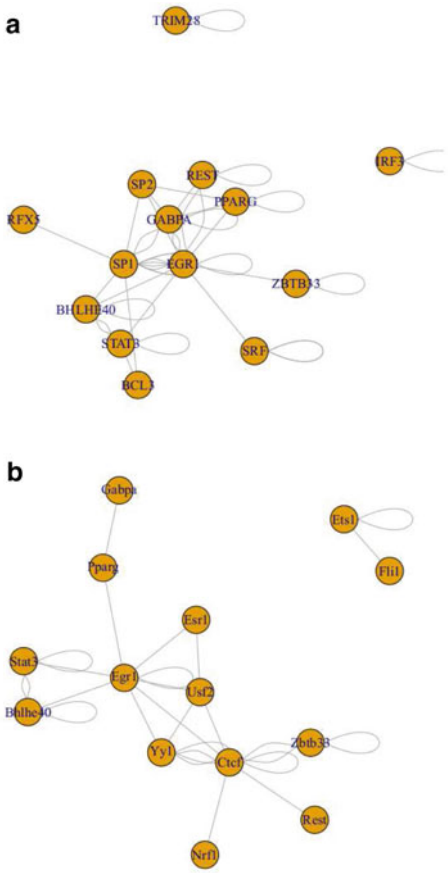

Fig. 3 TF network identified by regnetworkweb for TFs in Table 23. **a** human, **b** mouse

cell plates. Both sexes and two genotypes were also considered. Thus, gene expression is formatted as a tensor, $x_{j_1 j_2 j_3 j_4 j_6 j_6 i} \in \mathbb{R}^{96\times 2\times 2\times 4\times 2\times 4\times 29341}$ that represents ith gene expression of j_1th cell in 96 cells plate, genotype, j_2th genotype, j_3th tissue (brain part), j_4th time point, j_5th sex, and j_6th plate. Higher order singular value decomposition (HOSVD) [9] was applied to $x_{j_1 j_2 j_3 j_4 j_6 j_6 i}$ and we got

$$x_{j_1 j_2 j_3 j_4 j_5 j_6 i} = \sum_{\ell_1=1}^{96}\sum_{\ell_2=1}^{2}\sum_{\ell_3=1}^{2}\sum_{\ell_4=1}^{4}\sum_{\ell_5=1}^{2}\sum_{\ell_6=1}^{4}\sum_{\ell_7=1}^{29341} G(\ell_1,\ell_2,\ell_3,\ell_4,\ell_5,\ell_6,\ell_7)u_{\ell_1 j_1}u_{\ell_2 j_2}u_{\ell_3 j_3}u_{\ell_4 j_4}u_{\ell_5 j_5}u_{\ell_6 j_6}u_{\ell_7 i} \tag{17}$$

where $G(\ell_1,\ell_2,\ell_3,\ell_4,\ell_5,\ell_6,\ell_7) \in \mathbb{R}^{96\times 2\times 2\times 4\times 2\times 4\times 29341}$ is core tensor, $u_{\ell_1 j_1} \in \mathbb{R}^{96\times 96}$, $u_{\ell_2 j_2} \in \mathbb{R}^{2\times 2}$, $u_{\ell_3 j_3} \in \mathbb{R}^{2\times 2}$, $u_{\ell_4 j_4} \in \mathbb{R}^{4\times 4}$, $u_{\ell_5 j_5} \in \mathbb{R}^{2\times 2}$, $u_{\ell_6 j_6} \in \mathbb{R}^{4\times 4}$ and $u_{\ell_6 i} \in \mathbb{R}^{29341\times 29341}$ are singular value matrices that are orthogonal matrices.

At first, we need to specify what kind of properties should be associated with singular value vectors. Gene expression profiles should be independent of j_1 (cells), j_2 (genotype), j_3 (tissue), j_5 (sex), and j_6 (plate), but should depend upon age. As can

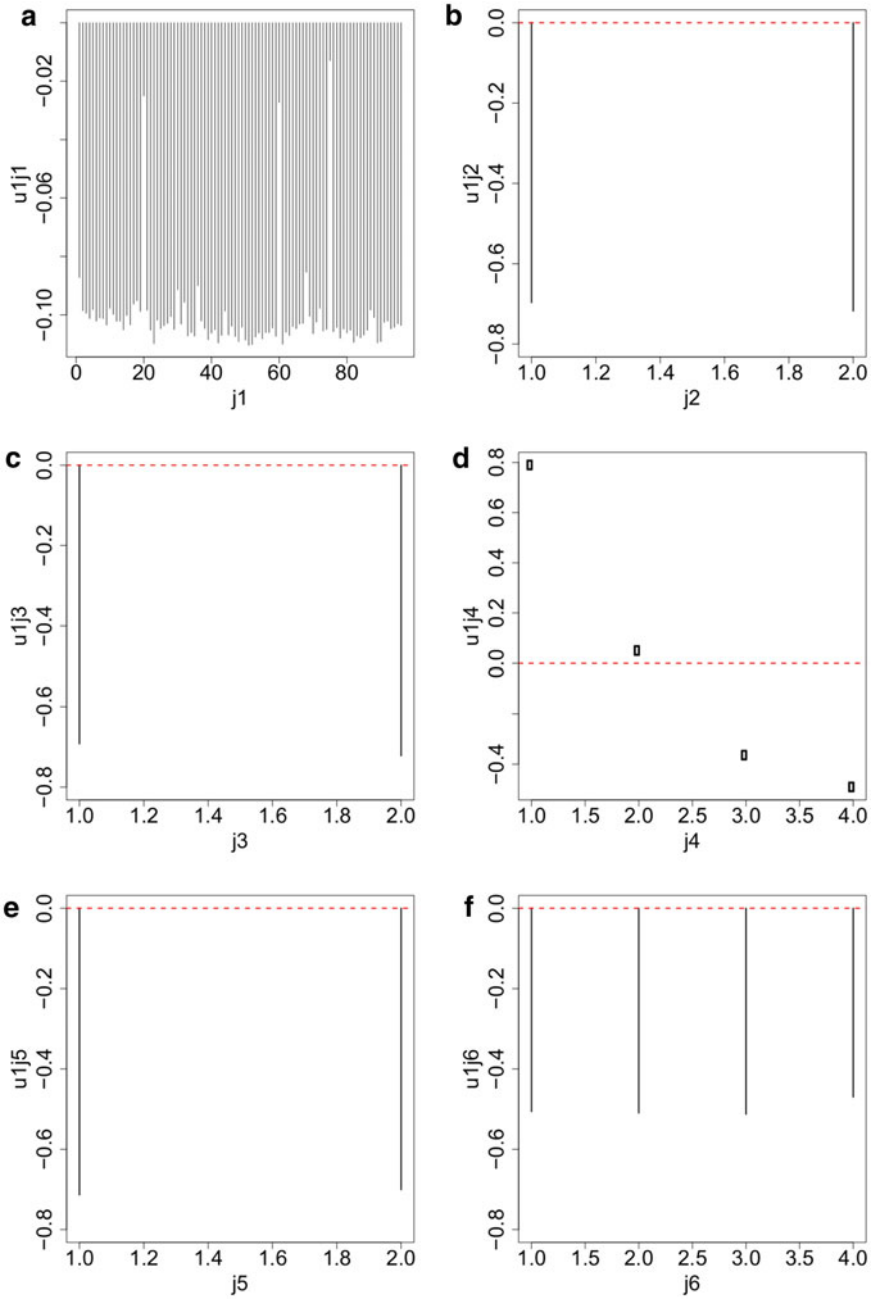

Fig. 4 Singular value vectors. **a** u_{1j_1} **b** u_{1j_2} **C** u_{1j_3} **d** u_{2j_4} **e** u_{1j_5} **f** u_{1j_6}

Table 24 Top ranked 10 compounds listed in "LINCS L1000 Chem Pert up" category in Enrichr. Overlap is that between selected 401 genes and genes selected in individual experiments

Term	Overlap	P-value	Adjusted P-value
LJP006_HCC515_24H-alvocidib-10	28/221	7.99×10^{-15}	2.21×10^{-10}
LJP006_HCC515_24H-AZD-8055-10	24/188	5.87×10^{-13}	8.13×10^{-9}
LJP009_PC3_24H-CGP-60474-3.33	25/217	1.99×10^{-12}	1.14×10^{-8}
LJP005_MDAMB231_24H-AS-601245-10	20/132	2.05×10^{-12}	1.14×10^{-8}
LJP009_PC3_24H-saracatinib-10	24/196	1.47×10^{-12}	1.14×10^{-8}
LJP006_HCC515_24H-CGP-60474-0.37	24/225	2.89×10^{-11}	1.14×10^{-7}
LJP009_PC3_24H-PF-3758309-10	23/212	5.33×10^{-11}	1.84×10^{-7}
LJP005_HCC515_24H-WZ-3105-3.33	20/144	1.07×10^{-11}	4.95×10^{-8}
LJP006_HEPG2_24H-AZD-5438-10	21/182	1.17×10^{-10}	3.24×10^{-7}
LJP006_HCC515_24H-A443654-10	22/203	1.44×10^{-10}	3.62×10^{-7}

be seen in Fig. 4, $\ell_1 = \ell_2 = \ell_3 = \ell_5 = \ell_6 = 1$ and $\ell_4 = 2$ fulfill the requirements. Then try to find which $G(1, 1, 1, 2, 1, 1, \ell_7)$ has the larger absolute value; then we found $\ell_7 = 2$ is associated with the largest absolute value of $G(1, 1, 1, 2, 1, 1, \ell_7)$. Then P-values were attributed to ith genes as

$$P_i = P_{\chi^2}\left[> \left(\frac{u_{2i}}{\sigma_2}\right)^2 \right] \tag{18}$$

and P-values were corrected by BH criterion. 401 is (genes) associated with adjusted P-values less than 0.01 were selected.

Now we can perform drug discovery using these genes. Generally speaking, brain of aging mouse is a model of Alzheimer's disease (AD). Thus we can find some compounds that can "reverse" the gene expression profile progression during aging. Such drug compounds can be easily identified using Enrichr, since Enrichr can identify drugs compounds that can significantly affect the expression of uploaded genes. By uploading selected 401 genes to Enrihr, we found that various compounds can affect expression of uploaded genes. Tables 25, and 26 are examples of such compounds that include multiple promising compounds. The top ranked compound in Table 24, alvocidib, was previously tested for AD [6]. The second top ranked compound in Table 24, AZD-8055, was also previously tested for AD [2]. The top, fifth and tenth ranked compound in Table 25, cyclosporin-A, was also previously tested for AD [4].

Table 25 Top ranked 10 compounds listed in "DrugMatrix" category in Enrichr. Overlap is that between selected 401 genes and genes selected in individual experiments

Term	Overlap	P-value	Adjusted P-value
Cyclosporin_A-350_mg/kg_in_Corn_Oil-Rat-Bone_marrow-5d-up	51/315	2.26×10^{-31}	1.78×10^{-27}
Isoprenaline-4.2_mg/kg_in_Saline-Rat-Heart-5d-up	49/304	4.55×10^{-30}	1.79×10^{-26}
Hydroxyurea-400_mg/kg_in_Saline-Rat-Bone_marrow-5d-up	46/307	7.54×10^{-27}	1.49×10^{-23}
Netilmicin-40_mg/kg_in_Saline-Rat-Kidney-28d-up	45/314	1.90×10^{-25}	1.50×10^{-22}
Cyclosporin_A-350_mg/kg_in_Corn_Oil-Rat-Bone_marrow-3d-up	45/312	1.45×10^{-25}	1.42×10^{-22}
Chlorambucil-0.6_mg/kg_in_Corn_Oil-Rat-Spleen-0.25d-up	47/314	2.13×10^{-27}	5.60×10^{-24}
Tobramycin-40_mg/kg_in_Saline-Rat-Kidney-28d-up	45/311	1.26×10^{-25}	1.42×10^{-22}
Gemcitabine-11_mg/kg_in_Saline-Rat-Bone_marrow-3d-up	47/344	1.27×10^{-25}	1.42×10^{-22}
Terbutaline-130_mg/kg_in_Corn_Oil-Rat-Heart-3d-up	45/321	4.89×10^{-25}	2.41×10^{-22}
Cyclosporin_A-70_mg/kg_in_Corn_Oil-Rat-Bone_marrow-3d-up	45/320	4.28×10^{-25}	2.25×10^{-22}

The top ranked compounds in Table 26, imatinib, was also previously tested for AD [1]. Since these coincidence found in three independent data sets, LINCS, DrugMatrix, and GEO, is unlikely accidental, we can regard our strategy, drug discovery based upon genes associated with altered gene expression, as being successful. This suggests that TD based unsupervised FE enables us to perform drug discovery using scRNA-seq data set.

4 Conclusion

Although scRNA-seq is a promising strategy, because of lack of sample (cell) labeling, it is difficult to make use of scRNA-seq for some purpose. Since PCA as well as TD based unsupervised FE does not require labeling information, they are suit-

Table 26 Top ranked 10 compounds listed in "Drug Perturbations from GEO up" category in Enrichr. Overlap is that between selected 401 genes and genes selected in individual experiments

Term	Overlap	P-value	Adjusted P-value
Imatinib DB00619 mouse GSE51698 sample 2522	81/288	2.27×10^{-70}	$2.05\times^{-67}$
Bleomycin DB00290 mouse GSE2640 sample 2851	80/329	6.09×10^{-64}	$2.75\times^{-61}$
Soman 7305 rat GSE13428 sample 2640	86/532	$3.87\times^{-53}$	3.50×10^{-51}
Coenzyme Q10 5281915 mouse GSE15129 sample 3464	76/302	$6.84\times^{-62}$	2.06×10^{-59}
N-METHYLFORMAMIDE 31254 rat GSE5509 sample 3570	70/283	$2.39\times^{-56}$	3.60×10^{-54}
Calcitonin 16132288 mouse GSE60761 sample 3446	65/220	$8.51\times^{-58}$	1.92×10^{-55}
Cyclophosphamide 2907 mouse GSE2254 sample 3626	78/413	$2.47\times^{-53}$	2.48×10^{-51}
Calcitonin 16132288 mouse GSE60761 sample 3447	59/177	$5.88\times^{-56}$	7.59×10^{-54}
PRISTANE 15979 mouse GSE17297 sample 3229	71/291	$1.03\times^{-56}$	1.87×10^{-54}
Coenzyme Q10 5281915 mouse GSE15129 sample 3456	76/396	$1.79\times^{-52}$	1.35×10^{-50}

able to be applied to scRNA-seq data. As a result, they have successfully selected genes associated with various biological terms related to the experiments. They are promising methods applicable to scRNA-seq data.

Acknowledgements The contents of this chapter were supported by KAKENHI, 19H05270 and 17K00417.

References

1. Eisele, Y.S., Baumann, M., Klebl, B., Nordhammer, C., Jucker, M., Kilger, E.: Gleevec increases levels of the amyloid precursor protein intracellular domain and of the amyloid-degrading enzyme neprilysin. Mol. Biol. Cell **18**(9), 3591–3600 (2007). https://doi.org/10.1091/mbc.e07-01-0035. PMID: 17626163
2. Hein, L.K., Apaja, P.M., Hattersley, K., Grose, R.H., Xie, J., Proud, C.G., Sargeant, T.J.: A novel fluorescent probe reveals starvation controls the commitment of amyloid precursor protein to the lysosome. Biochimica et Biophys Acta (BBA)—Mol. Cell Res. **1864**(10), 1554–1565 (2017). https://doi.org/10.1016/j.bbamcr.2017.06.011
3. Heiser, C.N., Lau, K.S.: A quantitative framework for evaluating single-cell data structure preservation by dimensionality reduction techniques. Cell Rep. **31**(5),(2020). https://doi.org/10.1016/j.celrep.2020.107576

4. Heuvel, C.V.D., Donkin, J.J., Finnie, J.W., Blumbergs, P.C., Kuchel, T., Koszyca, B., Manavis, J., Jones, N.R., Reilly, P.L., Vink, R.: Downregulation of amyloid precursor protein (app) expression following post-traumatic cyclosporin-a administration. J. Neurotrauma **21**(11), 1562–1572 (2004). https://doi.org/10.1089/neu.2004.21.1562. PMID: 15684649
5. Kuleshov, M.V., Jones, M.R., Rouillard, A.D., Fernandez, N.F., Duan, Q., Wang, Z., Koplev, S., Jenkins, S.L., Jagodnik, K.M., Lachmann, A., McDermott, M.G., Monteiro, C.D., Gundersen, G.W., Ma'ayan, A.: Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucleic Acids Res. **44**(W1), W90–W97 (2016). https://doi.org/10.1093/nar/gkw377
6. Leggio, G.M., Catania, M.V., Puzzo, D., Spatuzza, M., Pellitteri, R., Gulisano, W., Torrisi, S.A., Giurdanella, G., Piazza, C., Impellizzeri, A.R., Gozzo, L., Navarria, A., Bucolo, C., Nicoletti, F., Palmeri, A., Salomone, S., Copani, A., Caraci, F., Drago, F.: The antineoplastic drug flavopiridol reverses memory impairment induced by amyloid β_{1-42} oligomers in mice. Pharmacol. Res. **106**, 10–20 (2016). https://doi.org/10.1016/j.phrs.2016.02.007
7. Liu, Z.P., Wu, C., Miao, H., Wu, H.: RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. Database **2015** (2015). https://doi.org/10.1093/database/bav095
8. Taguchi, Y.H.: Principal component analysis-based unsupervised feature extraction applied to single-cell gene expression analysis. In: Intelligent Computing Theories and Application, pp. 816–826. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-95933-7_90
9. Taguchi, Y.H.: Unsupervised Feature Extraction Applied to Bioinformatics, A PCA Based and TD Based Approach. Springer International (2020). https://doi.org/10.1007/978-3-030-22456-1
10. Taguchi, Y.H., Turki, T.: Neurological disorder drug discovery from gene expression with tensor decomposition. Curr. Pharm. Des. **25**(43), 4589–4599 (2019)
11. Taguchi, Y.H., Turki, T.: Tensor decomposition-based unsupervised feature extraction applied to single-cell gene expression analysis. Front. Genet. **10**, 864 (2019). https://doi.org/10.3389/fgene.2019.00864. https://www.frontiersin.org/article/10.3389/fgene.2019.00864

Machine Learning: A Tool to Shape the Future of Medicine

Orsalia Hazapi, Nefeli Lagopati, Vasileios C. Pezoulas, G. I. Papayiannis, Dimitrios I. Fotiadis, Dimitrios Skaltsas, Vangelis Vergetis, Aristotelis Tsirigos, Ioannis G. Stratis, Athanasios N. Yannacopoulos, and Vassilis G. Gorgoulis

Abstract The constant evolution of biomedicine, biophysics and biochemistry has enabled scientists to investigate and study each cell identity via analyzing the transcriptome and its kinetics, the chromatin accessibility patterns but also via investigating the structure of proteins and RNA. Taking all this under consideration, scientists have developed algorithms and machine learning (ML) schemes that take advantage of the current state-of-the-art approaches to predict the cell states, discover the exact 3D

Yannacopoulos and Gorgoulis contributed equally to this study.

O. Hazapi · N. Lagopati · V. G. Gorgoulis
Molecular Carcinogenesis Group, Department of Histology and Embryology, School of Medicine, National and Kapodistrian University of Athens, 75 Mikras Asias Str., 11527 Goudi, Athens, Greece

O. Hazapi · D. Skaltsas · V. Vergetis
Intelligencia Inc., New York, NY, USA

N. Lagopati
Biomedical Research Foundation of the Academy of Athens, Athens, Greece

V. C. Pezoulas · D. I. Fotiadis
Unit of Medical Technology and Intelligent Information Systems, University of Ioannina, Ioannina, Greece

G. I. Papayiannis
Department of Naval Sciences, Section of Mathematics, Hellenic Naval Academy, Piraeus, Greece

G. I. Papayiannis · A. N. Yannacopoulos (✉)
Department of Statistics, Stochastic Modeling and Applications Laboratory, Athens University of Economics and Business(AUEB), 76 Patision Str., 104 34 Athens, Greece
e-mail: ayannaco@aueb.gr

A. Tsirigos
Department of Pathology and Institute for Computational Medicine, New York University School of Medicine, New York, NY, USA

I. G. Stratis
Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece

A. N. Yannacopoulos
Department of Statistics, Athens University of Economics and Business, Athens, Greece

S. S. Roy and Y.-H. Taguchi (eds.), *Handbook of Machine Learning Applications for Genomics*, Studies in Big Data 103,
https://doi.org/10.1007/978-981-16-9158-4_12

structure of proteins and RNA and most importantly evaluate personalized medicine approaches via predicting drugs and specific immunotherapy treatments. Moreover, the recent advances in ML and chemo-informatics have also paved the way for drug repurposing models, thus evaluating and establishing *in silico* novel treatments. The aim of this chapter is to provide and analyze the mathematics behind such ML techniques and review the current applications being developed that walk side by the side with the continuous progress of biosciences.

Keywords Machine learning · Artificial intelligence · Supervised-unsupervised learning · Biomedical applications

1 Introducing Machine Learning (ML)

The basic aim of ML is to connect certain characteristics, usually referred to as features, with certain target variables called responses. Denoting by X the feature space (typically R^n) and, by Y the response space (typically R^m with most applications reduced to $m = 1$) under the most general setting we can represent the data generation mechanism through a function $f : X \to Y$.

Once this function f has been acquired (learned) from the available data, then it can be used on new data characterized by the feature vector $x \in X$ to obtain the corresponding response $y \in Y$. Depending on the application task, ML techniques attempt to retrieve, uncover, or reconstruct that mechanism which characterizes the data.

ML methods are widely used in medicine, biology, genomics, medical imaging, etc. to model, explore and distinguish the data patterns in various applications. Actually, a more representative term is statistical learning since the applications field is in general restricted to data analysis, modeling, and pattern recognition. Learning methods can be roughly classified in two basic categories:

(a) Unsupervised Learning Methods, when certain patterns are trying to be estimated by studying data without any a priori knowledge about the "true" situation regarding data. The core techniques of this category are the clustering approaches, where available data are grouped (clustered) to various groups (clusters) using certain proximity measures and criteria. The main purpose in this procedure, is to conclude to a number of distinct groups where both

V. G. Gorgoulis
Division of Cancer Sciences, Faculty of Biology, Medicine and Health, Manchester Academic Health Science Centre, Manchester Cancer Research Centre, NIHR Manchester Biomedical Research Centre, University of Manchester, Manchester, UK

Center for New Biotechnologies and Precision Medicine, Medical School, National and Kapodistrian University of Athens, Athens, Greece

(✉)
Faculty of Health and Medical Sciences, University of Surrey, Surrey, UK
e-mail: vgorg@med.uoa.gr

the homogeneity between the elements in each group and the heterogeneity between different groups are maximized in terms of the metric sense that is used.

(b) Supervised Learning Methods, where given available datasets, structure relations are trying to be captured/recovered/explored. The relations are inferred with a function derived from labeled training data targeting a desired output value. This category constitutes of a great variety of statistical modeling techniques like regression methods and classification models either on parametric or nonparametric setting. This scientific field remains very active since more complex data structures require more specialized treatment than the classical methods can offer [82].

This chapter aims at a short overview of ML in biology. Due to space limitations, rather than providing a comprehensive overview, we are focusing on certain methods which are used more frequently in biological applications.

2 A Motivating Example of Machine Learning in Biology

We motivate this overview on ML methods with an illustrative example, presenting a classification regime in biology. In order to understand the goal of this approach it is important to define the framework of the biological experiment that is going to be presented. Thus, the goal is to profile the RNA of tumor cells [57] and thus recognize different cell types. Briefly to profile the mRNA in each cell, single-cell isolation is performed either isolating cells into wells on a plate or performing isolation based on droplet methods [17] while using a microfluidic device.

The isolation of the intracellular RNA is captured via performing biochemical cell lysis to capture the RNA, reverse transcribing to cDNA and amplifying each RNA per cell, while a droplet based-specific cellular barcode is used. Our goal then is to classify the cell types, based on the transcriptome of each cell, analyze the trajectories, the cell lineages and identifying the cells or clusters that are related but also demonstrating how they diffuse. This procedure enables to visualize and predict the cell state [71]. In Fig. 1 the tasks are described for such problem, followed by the analysis in the mathematics that need to be performed behind such task. Since, cell states and cell types are not a priori known and the respected data have quite a high dimension, before the implementation of a classification scheme, it is required to employ a dimension reduction technique and then to use the new condensed data to perform the clustering or classification tasks. In the following sections, the basic information for some popular ML methods used in biology and genomics are presented.

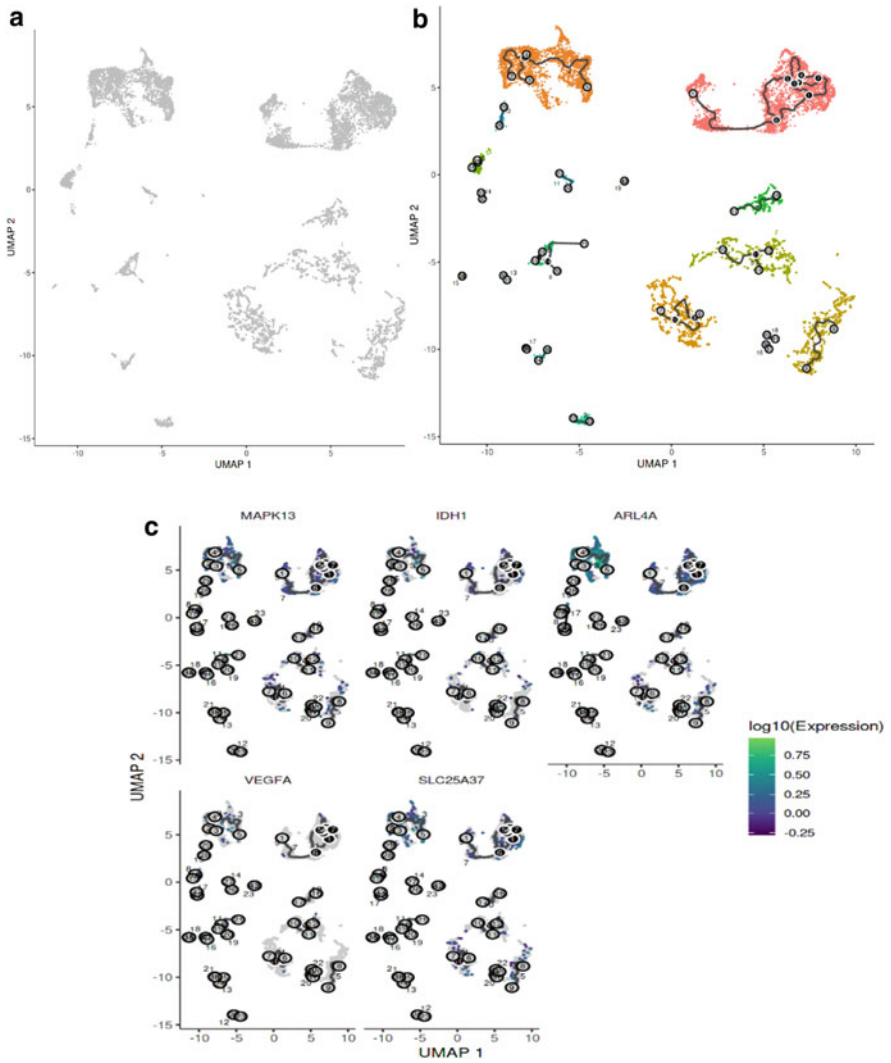

Fig. 1 scRNA-seq UMAP dimensionality reduction of breast cancer cell types in **a**, in **b** trajectory construction with Monocle [108] where the circled numbers are used to define cells and cell progenitors. In **c**, gene marker expressions per cell sub populations along with the cell trajectories are demonstrated

3 The Mathematical Concepts of Some Commonly Used ML Schemes in Biology

In this section we present the mathematical concepts behind some commonly used ML schemes in biology. Due to space restrictions this chapter focuses on certain recent methods that are currently commonly used in biological applications.

3.1 A Selection of Unsupervised ML Schemes

3.1.1 The k-Means Clustering

Let us use the scRNA-seq experiment as described above and assume that every cell is characterized by a set of n features and abstracted as a data-point on the space R^n. Hence, our data are considered as embedded in the Euclidean space R^n i.e. a sample is considered as a cloud of points in R^n and similarity between points is measured in terms of some distance, e.g. the Euclidean distance (the smaller the distance of two points the larger their similarity). Given a sample $X = \{x_1, \ldots, x_N\} \subset R^n$, (often interpreted as the data matrix $X_M = [x_1 \ldots, x_N] \in R^{n \times N}$), the aim of a clustering method is that each cell (point) will be allocated to specific clusters of cells (points) based on identified centroids where the distance from each centroid is least, therefore leading to collections (clusters) of like cells.

The k-means algorithm [80] is perhaps the most popular and simplest data clustering algorithm. According to this algorithm, given the data X, clusters of similar data points, can be obtained by arranging these data points in sets of points $C_r \subset R^n, r = 1, \ldots, k$. Each cluster C_r is characterized by its center m_r, and the points of each cluster are similar in terms of the distance among themselves and the cluster center. This process is easily visualized if $n = 2, 3$. But, in cases which are characterized by high dimensional data, which is common in applications with practical interest, there is the need of a more sophisticated abstract formulation. Hence, assuming that $x_i, i \in N_r, N_r \subset \{1, \ldots, N\}$, are the data points in cluster $C_r, r = 1, \ldots, k$ and the center of the cluster is the point $m_r \in R^n$, then the vector m_r minimizes the intra-cluster variability, $m_r = \arg\min_{z \in R^n} F(z)$, where $F(z) = \frac{1}{2} \sum_{i \in C_r} \|x_i - z\|^2$. The intra-cluster variance is defined as the minimum value of the (Fréchet) function F:

$$V_r = F(m_r) = \frac{1}{2} \sum_{i \in C_r} \|x_i - m_r\|^2. \tag{1}$$

The more compact (or homogeneous) the cluster, the smaller the quantity V_r is, and evidently, the above quantities depend on the choice of data points that are included in a particular cluster C_r, i.e., $m_r := m_r(C_r)$ and $V_r := V_r(C_r)$. Clearly, a successful

choice of the center of the cluster would allow to obtain a suitably small value of the corresponding variance within the cluster V_r.

With the above introduction and setting at hand, we may now express the problem of partitioning our data into a pre-defined number of disjoint clusters, say k, of composition to be determined. Iteratively seek for a set of k-disjoint clusters, say $C = \{C_1, \ldots, C_k\}$, that minimize the total in cluster variance of the data samples assigned on each cluster, as in:

$$C = \arg \min_{C_1,\ldots,C_k} \sum_{r=1}^{k} V_r(C_r). \tag{2}$$

This variational problem could be solved applying k-means clustering. This iterative process is based on a random selection of k data points from X, serving as the initial candidates for the clustering centroids. The remaining data points $x_j \in X$, where $x_j \neq m_r, r = 1, \ldots, k$ can be assigned into k clusters by calculating the Euclidean distance between each data point from each clustering centroid. If the distance of x_j from the r-th clustering centroid is smaller than the distance of x_j from the rest of the clustering centroids, then a data point x_j is assigned to the cluster C_i.

In other words, a data point x_j is assigned to the cluster C_i if it minimizes: $\|x_j - m_i\|^2$ i.e.,

$$i = \arg \min_{r=1,2,\ldots,k} \|x_j - m_r\|^2 \tag{3}$$

Once all data points have been assigned to the clusters, the cluster centers $m_r, r = 1, \ldots, k$, are recalculated (now no longer necessarily coinciding with the original data points) and the data are reassigned into new clusters as above, iterating the procedure until the clusters no longer change in composition. The outcome of the algorithm at the final step is the optimal assignment of the data points into the clusters and the resulting cluster composition. k-means clustering is an NP hard problem regarding its computational complexity.

Several variants of clustering methods have been proposed relaxing e.g., the assumption that similarity is expressed in terms of Euclidean distance and using spectral distances [113] or applicable for complex data such as correlation matrices or probability models [78]. An evaluation of k-means has been extensively used in cases of scRNA-seq in creating different subclasses of cell types [43].

3.2 Dimensionality Reduction and Feature Selection

The fundamental objective of dimensionality reduction methods is looking for the optimal data representation model by compressing data in order to describe the initial conditions, without losing significant information. This procedure is achievable, since there are many redundancies in most datasets. Particularly, some sets of

properties can be considered as an indicator of other latent features that remained initially unobserved. This means that the original features of the datasets might be correlated. In the absence of number of redundancies existing within a dataset compression, it is impossible to obtain significant results, regardless of the method that is used.

A dimensionality reduction technique focuses on representing data, which are initially represented as n dimensional, as elements of an m dimensional space ($m < n$) with the minimum loss of information. From a mathematical point of view, this corresponds to the following: given a set of data points $X = \{x_1, \ldots, x_N\} \subset R^n$, we wish to find a mapping $\psi : R^n \to R^m$, where $m < n$, such that $Y = \{y_1 = \psi(x_1), \ldots, y_N = \psi(x_N)\} \subset R^m$, is a lower dimensional representation of the original data set satisfying certain desired properties (which differ from method to method and will be clearly stated in the following sections). This map ψ is called an embedding of the original data set to a lower dimensional space. The goal of dimensionality reduction and feature selection methods is the construction of this map (equivalently the determination of the new data set Y) that may be used either for visualization or for further use with other ML algorithms on the lower dimensional space.

Often, even if the data X are initially described as elements of the space R^n, they are situated in nonlinear subsets of R^n, which are intrinsically of a lower dimension. For instance, points on the circumference of the unit circle which even though they are points in R^2, are actually points on an one dimensional curve (embedded in R^2), according to what it clearly indicated through the representation $x_i = (\cos(\theta_i), \sin(\theta_i)) \in R^2$, $\theta_i \in (0, 2\pi)$, $i = 1, \ldots, N$. Being able to obtain this information from the data leads to a natural 1-d representation of our original 2-d cloud of data points, which leads to more accurate representations of the data. What is described in this simple example is often the case (for more complicated geometrical objects) in biological data. Such geometric objects are called manifolds [59], and they are essentially nonlinear lower dimensional objects M embedded in R^n, but their local structure can be described in term of planes in R^m ($m < n$, with m called the dimension of the manifold) and in fact each neighborhood of M can be locally described in terms of m coordinates. For any neighborhood $N_y \subset M \subset R^n$, containing $y \in M$, a neighborhood $U \subset R^m$ and a mapping $\Phi_y : U \to N_y$ exist and this is a diffeomorphism (i.e., both the mapping and its inverse are smooth). Using this map Φ_y, we can represent all points in N_y (which are points in R^n in terms of the set of coordinates $(z_1, \ldots, z_m) \in U$, i.e. each point in $y' \in N_y$ can be represented by $y' = \Phi_y(z'_1, \ldots, z'_m)$ for some $z' = (z'_1, \ldots, z'_m) \in U$, and this local procedure can be performed for all neighborhoods of M, (notably, with different maps Φ). Importantly, this is done in a fashion such that these local mappings are compatible for overlapping neighborhoods. Dimensionality reduction methods based on such assumptions go under the general name of manifold learning and are very popular in biological applications. A large number of linear or nonlinear dimensionality reduction methods can be expressed in terms of trace optimization problems [52].

3.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely adopted dimensionality reduction method, aiming to create a feature space, by reducing the dimensions of the original dataset, while preserving as much variances as possible. In mathematical terms, and using the same notation as above, we consider as given the data matrix $X_M = [x_1, \ldots, x_N] \in R^{n \times N}$, where without loss of generality assume that the data are centered (if not we can do by removing the sample means of each feature $j = 1, \ldots, n$ using the matrix centering operation $\bar{X}_M = X_M C_N$, where $C_N = (I_d - \frac{1}{N} 1_N 1_N^T)$ is the centering matrix with $1_N \in R^{N \times 1}$ the vector with all elements equal to one and $\mathrm{I_N}$ the identity matrix of dimension N.). In this notation $S = \frac{1}{N} \bar{X}_M \bar{X}_M^T = \frac{1}{N} X_M C_N X_M^T$ corresponds to the (sample) covariance matrix that provides information concerning correlation of the various features (where we used the fact that C_N is idempotent).

PCA aims to find linear transformations of the original data matrix, resulting to a new feature matrix $Y = V^T X_M \in R^{m \times N}$, where $V \in R^{N \times m}$ is an orthonormal matrix that defines the embedding in R^m, and is to be determined. The new features can be considered as a new coordinate system called the principal components. In some sense, PCA is like projecting the data into a lower dimensional space R^m, than the original representation in R^n and this comes with the risk of producing new data with no actual information in them, which in this framework is related to the variance of the data. Hence, our aim is to project to data in the new space keeping as much of the original variance of the data in the new representation, which of course depends on the choice of transformation V. Note that the covariance matrix in the new representation is $\bar{Y} \bar{Y}^T = Y C_N Y^T = V^T X_M C_N X_M^T V$.

With the above considerations (and keeping into account that by construction the principal components are uncorrelated) PCA reduces to the optimization problem

$$max_{V \in R^{N \times m}} Tr(V^T \bar{X_M} \bar{X}_M^T V) \text{ subject to } V^T V = I_m. \tag{4}$$

The solution to this problem is provided by the orthogonal matrix $V = [v_1, \ldots, v_m] \in R^{N \times m}$, where v_i are the solutions to the eigenvalue problem $(\bar{X}_M \bar{X}_M^T) v_i = \lambda_i v_i$, corresponding to the m larger eigenvalues of the matrix $\bar{X}_M \bar{X}_M^T$. Note that this can also be provided in terms of the SVD decomposition of the matrix $\bar{X} = V \Sigma U^T$.

3.2.2 The t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a method based on Stochastic Neighbor Embedding [41, 111] in which a nonlinear mapping $f : R^n \to R^m$, is constructed for $m = 2, 3$ specifically. The low dimensionality of the target space makes the method ideal for visualization purposes. The algorithm strives to map as closely as possible the dissimilarity structure of the original data set X to a dissimilarity structure of the transformed low dimensional representation Y. The dissimilarity structures are understood in terms of probability distributions. In particular, the high dimensional points are described in terms of a probability distribution over pairs of high dimensional objects, for example the

similarity of the data point $x_j \in R^n$, to the data point $x_i \in R^n$ is expressed in terms of the quantity $p_{j|i}$ where:

$$p_{j|i} = \frac{\exp((-\|x_i - x_j\|^2)/(2\sigma_i^2))}{\sum_{k \neq i} \exp((-\|x_i - x_k\|^2)/(2\sigma_i^2))} \tag{5}$$

where σ_i is a suitable parameter, and $p_{j|i} = 0$ if $i = j$. In the original paper of Van Maaten et al., $p_{j|i}$ is interpreted as the probability that i chooses as neighbor j if neighbors are selected according to a normal distribution [111]. Since $\sum_{j \neq i} p_{j|i} = 1$ for all $i = 1, \ldots, N$ we can interpret $P_i = (p_{j|i}, \ldots, p_{N|i})$ as discrete probability distributions. We now assume that the mapping has been performed and we have the new data points y_i, $i = 1, \ldots, N$, and compute similar similarity measures for the new data in the same spirit as above. While the normal distribution can still be used in the target space, other options are possible, a popular one being the student t-distribution with one degree of freedom, which on account of the heavy tails provides a solution to the so-called crowding problem [111]. According to this, the similarities in the low dimensional space are

$$q_{j|i} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_i - y_k\|^2)^{-1}} \tag{6}$$

which admit a similar probabilistic interpretation as the $p_{j|i}$, and may also be used to construct the probability distributions $Q_i, i = 1, \ldots, N$. We then try to choose the embedding so that the probability distributions $P_i, Q_i, i = 1, \ldots, N$, match as much as possible, which is done by minimizing the Kullback-Leibler divergence or relative entropy [54], between these distributions, defined as:

$$L(Y) = \sum_{i=1}^{N} KL(P_i || Q_i) = \sum_{i=1}^{N} \sum_{i=1}^{N} p_{ij} \log(p_{ij}/q_{ij}), \tag{7}$$

which obviously depends on Y as the probabilities q depend on Y through (6). Note that we deliberately use the notation p_{ij} instead of $p_{j|i}$ symmetrized versions of these probabilities are used (similarly for q). The optimal mapping is obtained by minimizing L with respect to Y, a task usually performed in terms of gradient like algorithms [90].

3.2.3 Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS) is a class of algorithms whose goals are given a notion of dissimilarity (not necessarily implied by a metric) between high dimensions, provide an optimal lower dimensional configuration (i.e., suitable for visualization purposes). This happens through a map that as much as possible translates original dissimilarities to distances in the lower dimensional representation. MDS methods

are often divided in metric MDS or non-metric MDS, depending on whether the initial dissimilarity matrix derives from a notion of metric in the original space or not. Here we mention just a couple of these approaches, focusing within the class of metric MDS. Within this class one may consider for example metric least squares scaling. In this method given an original set of high dimensional data $\{x_1, \ldots, x_N\} \subset R^n$ with a dissimilarity matrix $D_X = (d_{ij}^X)$ one seeks a lower dimensional representation $\{y_1, \ldots, y_N\} \subset R^m$ with a dissimilarity matrix $D_Y = (d_{ij}^Y)$ chosen such that the weighted loss function $L(Y) = \sum_{i<j} w_{ij}(d_{ij}^Y - f(d_{ij}^X))^2$, for a given set of weights w_{ij} and a chosen parametric function f satisfying monotonicity is minimized. The square root of the loss function is called stress. The solution of this problem yields the optimal metric distance scaling function. The choice of weights is usually related to the elements of the dissimilarity matrix D_X, for example the choice $w_{ij} = (d_{ij}^X)/(\sum_{k<l)} d_{kl}^X)$, leads to the so-called Sammon mapping, other choices include $w_{ij} = (d_{ij}^X)^{-2}$ or $w_{ij} = [\sum_{k<l}(d_{kl}^X)^2]^{-1}$. The choice of dimension m for the lower dimensional representation can be chosen by the properties of the stress function which is typically monotone decreasing as a function of the dimension, and the chosen dimension is such that the stress function achieves sufficiently small values. Non-metric MDS methods are used to allow for representation of ordinal data.

3.2.4 Laplacian Eigenmaps

The method of Laplacian eigenmaps is based on the use of the graph Laplacian for constructing the weighted graph approximating the manifold M.

1. Local Neighborhood Structure: Choosing the method of K nearest neighbors or using local neighborhoods determined by balls of radius ϵ. The weights for the graph are either chosen using the exponential kernel $w_{ij} = \alpha \exp(-\|x_i - x_j\|^2/\beta)$, for suitable $\alpha, \beta > 0$, or setting $w_{ij} = 1$, if vertices i and j are connected and 0 otherwise (other choices being of course possible).
2. Setup the optimization problem for ψ: The embedding is chosen as the solution of the minimization problem

$$\min_{Y=[y_1,\ldots,y_N]} \sum_{i,j=1}^{N} w_{ij}\|y_i - y_j\|^2, \tag{8}$$

subject to the constraint $YDY^T = I_m$, where $D = diag(d_{11}, \cdots, d_{NN})$ with $d_{ii} = \sum_{j=1}^{N} w_{ij}$. This problem will help choose embeddings such that if w_{ij} is large (i.e., the points in the graph are close) the corresponding y_i, y_j in the embedding should also be close in the Euclidean space R^m. The above problem can be redressed in terms of the graph Laplacian matrix $L = D - W$, as a trace optimization problem

$$\min_{Y \in R^{m \times N}} Tr(YLY^T), \tag{9}$$

subject to the same constraints. The graph Laplacian is an important concept which provides important information on the structure of the graph, such as connectivity or the properties of random walks on the graph. It should be noted here that it can be shown that under condition the graph Laplacian converges to the Laplace-Beltrami operator of the manifold M, another important concept which provides characterization of some properties of M. In terms of the new formulation, it is easy to see that the optimal solution Y can be represented in terms of the generalized eigenvectors of the Laplacian, i.e., solutions to $Ly = \lambda Dy$, corresponding to m smallest eigenvalues of L. Note that through the transformation $z = D^{1/2}y$, the above problem becomes $\hat{L}z = \lambda z$, where $\hat{L} = D^{-1/2}LD^{-1/2}$ is the normalized Laplacian.

3. Spectral embedding: Ordering the eigenvalues of $\hat{L}$ as $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_{m+1}$ with the corresponding eigenvectors v_i, $i = 1, \ldots, m+1$, the optimal embedding will be provided in terms of $Y = [v_2, \ldots, v_{m+1}]^T D^{1/2} \in R^{m \times N}$.

3.2.5 Diffusion Maps

Diffusion maps is another popular methodology which is also based on the graph Laplacian, but now used differently in order to characterize connectivity patterns on the graph in terms of the time required by a random walker on the graph to connect between any two points.

1. Local Neighborhood Structure: As above construct the graph $G = (V, E, W)$.
2. Setup the optimization problem for ψ: The basic concept in this algorithm is a random walk on G, which is constructed as follows: given that you are currently at vertex x_i at the next time instance you may randomly jump to any vertex x_j, with probability $p_1(x_j \mid x_i) = w_{ij}/(\sum_{k \neq i} w_{ik}) = w_{ij}/d_{ii}$ (to simplify the exposition we take the case where W is symmetric). In this model, the propensity of moving from x_i to x_j depends on the weight w_{ij} (properly normalized) so that it is more probable to visit points which are closer (as that is determined by the affinity matrix). This concept introduces a more generalized idea of connectivity of the vertices of the graph G, on various time scales. Defining the matrix $P = D^{-1}W$, we can easily see that the evolution of the random walk in t times is given by the matrix P^t, in the sense that if we start with an initial probability distribution $p(0)$ on G, after the random walk has run for t times this probability distribution will evolve to $p(t) = P^t p(0)$. In fact, the elements of the matrix P^t, provide the transition probability between vertices in t iterations of the random walk, $(P^t)_{ij} = p_t(x_j \mid x_i)$. It is well known that the matrix P^t can be characterized in terms of its eigen-decomposition, so the behavior of the random walk can be captured in terms of the eigenvalue problem $Px = \lambda x$, or equivalently $Wx = \lambda Dx$. By elementary matrix operations this is related to the symmetric eigenvalue problem $D^{-1/2}WD^{-1/2}x = \lambda x$, (in fact the

two problems have the same spectrum) which is the eigenvalue problem for the matrix $\hat{W} = D^{-1/2} W D^{-1/2}$, which is related to the normalized Laplacian $\hat{L}$, in terms of $\hat{L} = I - \hat{W}$. The left and right eigenvectors of the matrix P, ϕ_i and ψ_i respectively (related to the eigenvectors of $\hat{W}$ by a simple transformation) play an important role in determining the structure of the random walk, with the left eigenvector corresponding to the eigenvalue 1, denoted by ϕ_0 corresponding to the stationary probability distribution on the graph, given in terms of $\phi_0(x_i) = D_{ii} / \sum_k D_{kk}$. This stationary distribution shows the propensity of the various edges of the graph to be populated by the random walk and is related to the frequency by which a random walker visits the edges, hence provides important information concerning connectivities. Finally the transition probabilities after t iterations of the random walk assume an expansion in terms of the eigensystem (λ_i, ϕ_i), $i = 0, \ldots, N-1$, as $p_t(x_j \mid x_i) = \phi_0(x_j) + \sum_{k \geq 1} \lambda_k^t a_k(x_i) \phi_k(x_j)$, where it can be easily proved that $a_k(x_i) = \psi_k(x_i)$. The above expansion shows the long-term behavior is captured by those eigenvalues whose modulus is large enough. It is important to stress that even if i and j are not directly connected in the graph G (i.e., the random walk cannot connect them in a single iteration) they can be connected in t iterations through alternative paths. This concept introduces a more generalized idea of connectivity of the vertices of the graph G, on various time scales. Based on this idea we may introduce a new diffusion distance for the graph as:

$$d_t^2(x_i, x_j) = \sum_{k \in V} |p_t(x_k|x_i) - p_t(x_k|x_j))|^2 [\phi_0(x_k)]^{-1} \tag{10}$$

where $p_t(x_k|x_i)$, $p_t(x_k|x_j)$, are the transition probabilities in t iterations of the random walk from x_i to x_k, and from x_j to x_k respectively and $\phi_0(x_k)$ is the invariant probability of the walk on the graph at vertex x_k. This distance reflects the notion that two vertices of the graph x_i, x_j are close if the probability of reaching any vertex $x_k \in V$ in t iterations, is roughly the same no matter if you start at x_i or x_j. Note that d_t^2 can also be interpreted as a weighted inner product distance in a space of transition probabilities, and more importantly using the expansion provided above for the transition probabilities may be interpreted as a weighted Euclidean distance in the space of the eigenvectors ψ_i, and in particular in terms of

$$d_t^2(x_i, x_j) = \sum_{k \geq 1} \lambda_k^{2t} (\psi_k(x_i) - \psi_k(x_j))^2 = \|\Psi_t(x_i) - \Psi_t(x_j)\|^2 \tag{11}$$

where $\Psi_t(x) = (\lambda_1^t \psi_1(x), \ldots, \lambda_{n-1}^t \psi_{n-1}(x))$ is an $n-1$ dimensional vector (the ψ_0 contribution is omitted since $\psi_0 = (1, \ldots, 1)$, which is interpreted as $\psi_0(x_i) = 1$, for any vertex of the graph).

3. Spectral embedding: It is important to observe that even though the vector $\Psi_t(x)$, for any x, is $n-1$ dimensional, by the weighting in terms of the eigenvalues $\lambda_k < 1$, as k increases the components of the vector corresponding to such

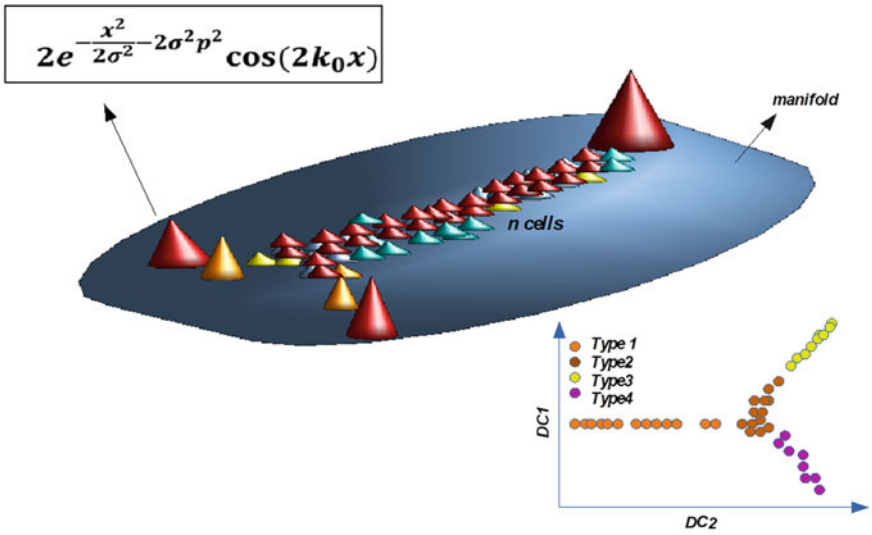

Fig. 2 Example of generating diffusion maps of cells from scRNA-seq experiment where each cell is allowed to move using an isotropic Gaussian wave function along a certain volume σ

k become negligibly small, so for all practical purposes the vector Ψ_t can be considered as lower dimensional. This is the key to dimensionality reduction in this method. We keep only the first m eigenvalues of $\hat{W}$, truncate the sum providing d_t^2 so the m dimensional vectors:

$$\hat{\Psi}_t(x_i) = (1, \lambda_1^t \psi_1(x_i), \ldots, \lambda_{m-1}^t \psi_{m-1}(x_i)), \ i = 1, \ldots, N, \qquad (12)$$

provide the approximate low dimensional embedding.

In Fig. 2, an example of generating Diffusion maps of cells from scRNA-seq experiment is presented, where each cell is allowed to move using an isotropic Gaussian wave function along a certain volume σ.

3.2.6 Uniform Manifold Approximation and Projection (UMAP)

UMAP is another algorithm for providing a low dimensional approximation for manifold data in terms of an appropriate embedding Y. The nature of this algorithm is different from the ones considered so far as it is not based on a spectral distance. The details of the algorithm are based on category theory, an abstract mathematical construction that allows mappings between different mathematical objects satisfying certain rules, which neither space nor the focus of this chapter allows us to present. We simply state that the key idea of the method is to map the data to (fuzzy) simplicial sets which are topological objects carrying far less information than manifolds as

they only carry the minimal information concerning connectivities. The algorithm can be summarized in the following steps.

Construct a graph representing the data. This is done in two steps.

1a. The vertices V are the data $\{x_1, \ldots, x_N\}$, and for each vertex local neighborhoods are constructed using the k-nearest neighbors approach.

1b. The geodesic distance on the manifold M is locally reconstructed for each vertex and corresponding neighborhood, by creating a local metric structure. These metric structures are not necessarily compatible, hence resulting to a collection of local metric spaces, one for each data point.

2 We now need to glue together all these local metric spaces. This is done using category theory arguments as follows: The local metric structures are mapped into fuzzy simplicial set structures in terms of appropriate mappings called morphisms which represent the topological properties of the local manifold approximations and then take their union.

Once the approximation of M has been completed as above, i.e., in terms of the fuzzy simplicial set let us call it X_S, we then seek this low dimensional representation of the data set $Y \in R^m$, in terms again of a fuzzy simplicial set Y_S on R^m (endowed with the Euclidean metric). This fuzzy simplicial set Y_S is chosen so that it minimizes a notion of disparity between fuzzy simplicial sets called the cross entropy. This is defined as follows: Assuming that $X = (A, \mu)$ and $Z = (A, \nu)$ are two fuzzy simplicial sets with A the set of simplicial objects (which carry the information of topological structure) and μ, ν measures providing an idea of the propensity of the elements of the simplicial sets to appear in the fuzzy simplical sets, their cross entropy is

$$C(X, Z) = \sum_{a \in A} (\mu(a) \log(\mu(a)/\nu(a)) + (1 - \mu(a)) \log((1 - \mu(a))/(1 - \nu(a))). \tag{13}$$

By finding Z that minimizes $Z \mapsto C(X, Z)$ with Z chosen among the set of fuzzy simplicial sets on R^m, we find the optimal low dimensional representation of M. This optimization problem can be handled by standard tools such as for example stochastic gradient descent.

The locality of the metric spaces on each vertex comes from the procedure used in the approximation of the Riemannian structure on M, which is supported by a differential geometric argument and requires scaling the metric for each neighborhood by the radius of the ball bounding the neighborhood. This is approximated for each neighborhood N_i by the distance from x_i, of the most distant of its k-nearest neighbors, denoted by r_i. If d_i is the local Riemannian metric on M used in N_i, an approximation of $d_i(x_i, x_j)$ would involve $d_X((x_i, x_j))/r_i$. Similarly if d_j is the local Riemannian metric on M used in N_j, an approximation of $d_j(x_i, x_j)$ would involve $d_X((x_i, x_j))/r_j$, where r_j is the distance (in X) from x_j of the most distant of its k-nearest neighbors. Clearly, in general $r_i \neq r_j$, so the metric structures (N_i, d_i) and (N_j, d_j) are different.

3.3 Supervised Learning Schemes

The philosophy behind such classification tasks is to analyze the a priori labeled training data and produce a derived function which can be used for the classification of new data. A vast number of algorithms have been developed, such as Decision Trees and Random Forests, k-NN, Naive Bayes, Neural networks, Regression techniques as well with Relevance Vector Machines (RVM) and Support Vector Machines (SVM). A brief overview of some of these methods will aid to the description of the applications of ML in biomedicine.

3.3.1 The k-Nearest Neighbors Algorithm (k-NN)

The k-nearest neighbors algorithm (k-NN) is a supervised method where the neighbors are taken from a set of objects for which the class is known. In our scRNA-seq problem for the classification of each cell, k-NN can be applied to these group cells in a specific class or state according to the closest Euclidean distance. This can be simplified via finding the k nearest data points and use their labels to predict the classes of new data points.

There are several versions of the k-NN algorithm, for example a weighted version which uses a weighted version of the distance when computing nearest neighbors. Another extension of the k-NN method is through the use of Voronoi graphs [120] to partition a plane to adjacent sites relatively close to a set of objects. Voronoi graphs are extensively being used to tackle hard biological problems such as with the modeling of the 3D structure of proteins (Fig. 3), to derive hot interacting locations on the proteins surface [45].

3.3.2 Naive Bayes Classifier

Naive Bayes [118] is considered as a popular probabilistic approach for supervised learning estimating the conditional probability for a random datum point being in class j. We assume that we have data points described by n features and abstracted as points $x = (x_1, \ldots, x_n) \in R^n$, and a target feature y or the class that the individual belongs to. This method assigns probabilities that a data point characterized by features x is classified in terms of y. This is achieved using the conditional probability $P(y|x)$ that is called the posterior probability. This is the intended outcome of this method, that is obtained in terms of information, which is available from the training data. According to Bayes theorem the conditional probability $P(y|x)$ is given by $P(y|x) = \frac{P(y)P(x|y)}{P(x)}$ where $P(x|y)$ denotes the likelihood while $P(x)$ and $P(y)$ the probabilities of the input and target features respectively. We then classify a data point x, to the class y for which $P(y|x)$ is maximized, leading to an estimation for the classifier in terms of $y \in \arg\max_y P(y|x)$, with $P(y|x)$ given in terms of the training data. To simplify the optimization problem required for the classification,

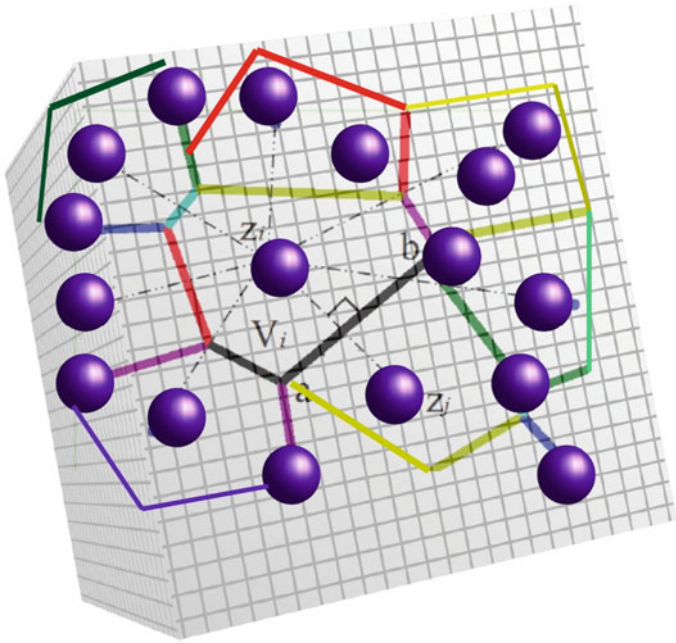

Fig. 3 Voronoi graph clustering of 3D dimensional molecules composing a protein

the likelihood term can be reduced by assuming that the conditional probabilities of each feature given the target feature y are independent (or in some cases conditionally independent), so that approximately $P(x|y) = \prod_{i=1}^{n} P(x_i|y)$, leading to a simplification of the above formula, where upon ignoring the unimportant factor $1/P(x)$, which does not depend on y, we obtain the following estimator

$$y \in \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y), \tag{14}$$

estimator Eq. 23: which is simpler to calculate, since the Maximum A Posteriori (MAP) rule [61] can be used to find an estimate, that maximizes the product of the likelihood and the prior probabilities.

Bayesian classification procedures have been applied to a variety of biological problems. This include from determining gene expression difference from RNA-seq analysis to the analysis of more complex systems such as in scRNA-seq analysis where probabilistic models are able to learn cell-specific parameters in order to drive normalization [24].

3.3.3 Support Vector Machine (SVM) Classifier

A Support Vector Machine (SVM) [75, 117] is a classifier which, given an input dataset consisting of n features that form a n-dimensional space, aims to find an optimal hyperplane that categorizes (classifies) them. Recall that a hyperlane H is defined as $H = \{z \in R^n \ : \ w \cdot z + w_0 = 0\}$, where $w \in R^n$ is a vector and $w_0 \in R$. A hyperplane H may separate R^n into two dinstinct parts H_+ and H_-, hopefully containing data points which are alike. Thus, the key concept is to find hyperplanes H (decision planes) that will allow the partitioning of data set into classes by distinctly defined decision boundaries.

Given a collection of data points $x_i = (x_1, \ldots, x_n), i = 1, \ldots, N$, which are to be classified in two qualitatively different classes 1 and 2, with the indices of points belonging to class 1 considered as members of a subset $N_1 \subset \{1, \ldots, N\}$ and indices of points belonging to class 2 considered as members of a subset $N_2 \subset \{1, \ldots, N\}$, with N_1 and N_2 being distinct, our goal is to find if a separating hyperplane H exists (equivalently a vector w and a scalar w_0 such that all points of class 1, satisfy the condition $x_i \in H_+$ while all points of class 2, i.e., points x_i with $i \in N_2$, satisfy the condition $x_i \in H_-$

The success of the separation scheme depends on how clearly the two classes are separated, and the following condition

$$\frac{(w \cdot x_i + w_0)}{\|w\|} > \frac{\epsilon}{\|w\|}, \ i \in N_1, \ \ -\frac{(w \cdot x_i + w_0)}{\|w\|} > \frac{\epsilon}{\|w\|}, \ i \in N_2, \tag{15}$$

for some $\epsilon > 0$, can be a reasonable choice, with the term on the left-hand side corresponding to the distance of the points from H. If we define the new variables y_i so that $y_i = 1$, if $i \in N_1$ and $y_i = -1$ if $i \in N_2$, and consider the problem of maximizing the distance $\frac{\epsilon}{\|w\|}$ and assuming that $\epsilon = 1$, then condition (15) can be expressed in a uniform fashion for all data points and the choice of the SVM reduces to a standard convex optimization problem:

$$\min_{(w,w_0) \in R^n \times R} \frac{1}{2}\|w\|^2, \quad \text{subject to} \ \ y_i(w \cdot x_i + w_0) > 1, \ i = 1, \ldots, N. \tag{16}$$

This is a problem of minimizing a quadratic function subject to affine constraints, belonging to a well-studied general class of convex optimization problems called quadratic optimization problems, that can be solved using duality techniques [53]. The particular problem, in the terminology of ML called the hard margin support vector machine (HM-SVM) [123]. The classification function may be redressed as $f(x) := sgn(w \cdot x + w_0)$, with the classification assignment for any data point x_i given in terms of the response $y_i = f(x_i)$.

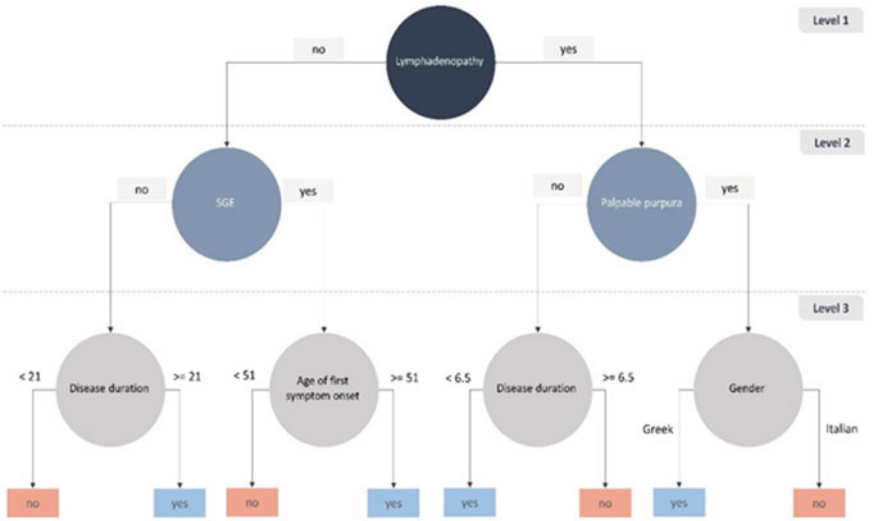

Fig. 4 An illustration of a decision tree

3.3.4 Decision Trees and Random Forests

Decision trees are used for both classification and regression supervised learning problems. A decision tree is an hierarchical data structure which is mathematically represented by a directed graph which consists of the following primary components: (i) the root node, which represents the most important feature in the dataset, (ii) a number of decision nodes, where each node implements a function that splits the data points into branches that can either be parents to other nodes or terminal nodes, and (iii) a number of terminal nodes (i.e., leaves) which correspond to the class labels. In fact, the decision tree is generated by a splitting process which starts from the root node and is recursively executed until the terminal nodes are reached. In decision tree classifiers, each unknown input follows a pathway that starts from the root node and ends to a leaf where the classification outcome is determined. The most widely used methods that are implemented towards the evaluation of the performance of a split are the Gini impurity index and the entropy. An illustration of a decision tree structure is depicted in Fig. 4. The random forests approach is an extension of the decision trees method and consists of a collection of randomized regression trees [8]. connected in terms of a topological node structure.

3.3.5 Neural Networks

Artificial Neural Networks (ANNs) are popular ML techniques that simulate the learning mechanism of biological organisms [9]. In analogy to the human nervous

system, consists of a set of interconnected neurons (similar to cells) and the synapses (similar to the regions connected by axons and dendrites on a biological brain). The strengths of synaptic connections alternate depending on the external stimuli that is received. This learning procedure is mimicked by the ANNs through a computational mechanism relying on a set of layers, weights, and activation functions. The layers stand for the various stages that information signals are processed. The first layer, the input layer, is the one with the original information of the network, i.e., the initial data. Next follow the hidden layers where each one of them consists of several artificial neurons tasked with transforming inputs from the previous layer to produce activations (new signals) for the next layers. These activatio ns are produced by the activation functions which typically are nonlinear functions which applied to the aggregated signals provided by the neurons of each layer, determine the type of input for the next layer's neurons. Clearly, the simplest case is when a single hidden layer is used while, when many hidden layers are used in the ANN's architecture constitute a deep neural network (a multilayer ANN). The last layer of an ANN is called the output layer and is the one that produces the results (estimated responses). In a feed-forward ANN, the information flow has a single direction: from the input layer towards the output layer, via the hidden layer(s).

To clarify the operation of an ANN let us discuss the simple case of a single layer and k neurons. The key components there are the set of input vectors $x=(x_1, \ldots, x_n)$, associated biases $b = (b_1, \ldots, b_k)$, and weights $w = \left(w_{ij},\ i = 1, \ldots, k,\ \ j = 1, \ldots, n\right)$ considered as a $k \times n$ matrix. Each neuron i takes the input vector x, nonlinearly transforms it in terms of an activation function and returns an output z_i which depends on the weights and the activation function as $z_i = h_i\left(\sum_{j=1}^{n} w_{ij}x_j + \mathrm{b_i}\right)$, where h_i is the activation function and its argument is often called the *activation* of the neuron. The choice of activation function is a critical part for the neural network's design and performance. In the simplest case like the perceptron (only one neuron is used), the use of sign function is motivated by the fact that a binary class label needs to be predicted. However, many other cases can occur, for example if the target variable to be predicted is real-valued, then the identity function will be an obvious choice along with a least-squares criterion for choosing weights which coincides with linear regression approach. If the target variable is the probability of a binary class, then the choice of sigmoid would make sense. Other popular choices are the sigmoid activation functions, ReLU, softmax, and others. The vector $z = (z_1,\ \ldots,\ z_k)$ is considered as the output of the neurons. If only a single hidden layer is present z can be used, along with a properly selected output matrix to provide the output z. If more than one hidden layer is used, then the output vector from layer L, denoted by z_L, is used as input for the next hidden layer L+1 (which clearly may consist of a different number of neurons, employ a different weight matrix or a different bias vector and activation function) to provide and intermediate output z_{L+1}, in the same fashion as above. Many variants of this general scheme have been proposed, with different architectures but this is the essential mechanism involved.

Based on training data the parameters of the ANN, which are essentially the weight matrices and the biases at each layer, are computed as the minimizers of an appropriate loss function Q which is used to measure the model's error. Similarly with the activation functions, the loss function's choice is critical in defining the network's outputs (estimations of the target variable) in a way that is sensitive to the application at hand. Clearly, the nature of the target variable should lead to an appropriate loss function choice. For example, a real-valued target variable may be best approximated by using a squared loss function, e.g., $\left(y - \hat{y}\right)^2$. If $y \in \{-1, 1\}$, the hinge loss function $max\left\{0, 1 - y \cdot \hat{y}\right\}$ could be a choice. Figure 5 demonstrates the activation function and the weights assigned to each node of the network. In this sense, the loss function quantifies the distance of the actual responses y from the estimated responses (the network's output) $\hat{y} = \hat{y}(w; x)$ which depend on the networks input x and is used to construct a distance criterion relying on the sense of distance that Q offers, i.e., $J(w) = \sum_{l=1}^{k} Q(y_l, \hat{y}_l(x; w_l)$. The minimization of the criterion $J(w)$ with respect to w (typically through back-propagation and stochastic gradient schemes) returns the mathematical model for the neural network and once an appropriate level of convergence has achieved the ANN can be used to provide estimations/predictions for new inputs.

For example, in a case of a binary classification problem, the goal is to configure the optimal hyperplane which will demonstrate a separation boundary between the two classes with the lowest cost function. For an ANN with a single-hidden layer (like the one in Fig. 5), quadratic loss function and activation function h, the hyperplane is estimated by solving the optimization problem

$$min_w \, J(w) = min_w \, \frac{1}{2} \sum_{i=1}^{N} (y_i - h(x_i; w))^2$$

where the weights $w = (w_1, w_2, \ldots, w_n)'$ are chosen employing gradient descent methods (or stochastic gradient descent) leading to the iterative scheme

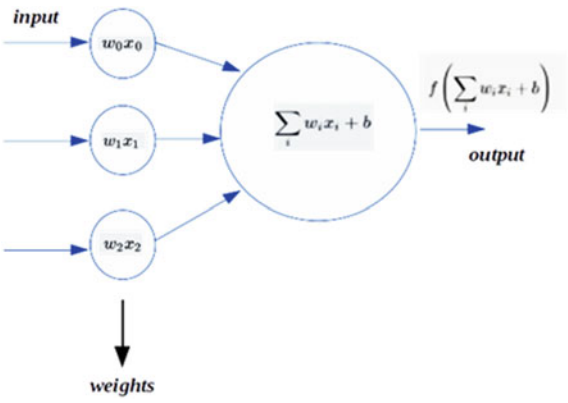

Fig. 5 Basic MLP design and activation function

$$w^{(l+1)} = w^{(l)} - \eta \cdot \Delta w^{(l)} = w^{(l)} - \eta \cdot \frac{\partial J\left(w^{(l)}\right)}{\partial w}, \quad l = 0, 1, 2, \ldots$$

where $\eta > 0$ denotes the learning rate and the step $\Delta w^{(l)}$ is chosen either from gradient or stochastic-gradient descent approaches or alternatively by evolutionary optimization methods like Particle Swarm Optimization (PSO). The above iterative procedure is typical for selecting the weights in feed-forward neural networks and when multiple hidden layers have been chosen this procedure is repeated until reaching the target weight, e.g., j-th weight on the k-th layer w_{jk} using the chain rule on J or more typically referred to as the *backpropagation* approach.

3.3.6 Self-organizing Maps (SOM)

Self-organizing maps [51] are a class of neural network methods, which can be used in classifying high dimensional data $X = \{x_1, \ldots, x_N\} \subset R^n$, into like groups depending on topological criteria. While it is an unsupervised technique, it is presented in this section as it relies on the concept of neural networks. For the sake of concreteness, we present here the idea in a simple problem, that of clustering the data X into k clusters. For any input datum $x = (x_1, \ldots, x_n) \in R^n$, we consider each feature as an input layer of n neurons, whereas the output layer will consist of k neurons (one for each corresponding cluster). Each neuron in the input layer is connected with a neuron in the output layer and it is the aim of the method to train the network as to which input goes to which output. Each output neuron j is characterized in term of the weight vector $w_j \in R^n$, with the k weight vectors to be learned in the procedure. Once the neural network has been trained using the data X and the vectors $w_j \in R^n$, $j = 1, \ldots, k$ (equivalently the weight matrix $W = [w_1, \ldots, w_k] \in R^{n \times k}$) have been determined, then any input point $x \in R^n$, is associated to the cluster $j^* \in \{1, \ldots, k\}$ for which the Euclidean distance $\|x - w_j\|^2$ is minimum, i.e. the cluster j whose weight vector w_j is the closer to the input x.

An important feature in this method is that the output layer is structured using a pattern of neighborhoods (i.e., for each output neuron j we can define a set of neurons N_j with are considered as its neighbors). To train the neural net we use the following procedure.

1. Initialize k weight vectors $w_j(0)$, $j = 1, \ldots, k$.
2. Cycling through the training data set, for any datum x at the ℓ iteration we

2(a) Find $j^* \in \{1, \ldots, k\}$ such that $\left||x - w_{j^*}\right|| = \min_{j \in \{1, \ldots, k\}} \left||x - w_j\right||$ and
2(b) Update the weight vectors w_j for $j \in N_{j^*}$ according to the Kohonen rule

$$w_j(\ell + 1) = w_j(\ell) + \alpha[x - w_j(\ell)]$$

where α is the learning rate.

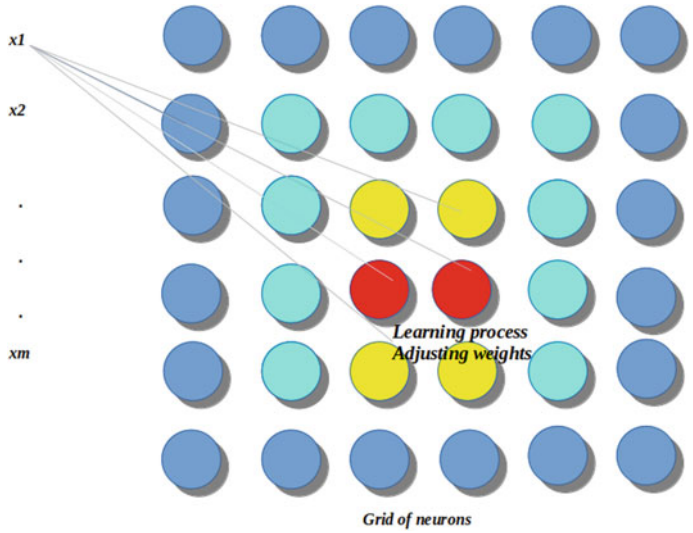

Fig. 6 SOM network representation while learning via adjusting the weights which can be visualized with different color coding

3. Continue until convergence, possibly updating the learning rate of diminishing the neighborhood structure.

Numerous variants to this scheme have been proposed, experimenting e.g., with various topologies, that find important applications in biology. Figure 6 represents a SOM network while learning via adjusting the weights which can be visualized with different color coding.

3.3.7 From Neural Networks to the Construction of Deep Learning Networks

The construction of large topological domains, the so-called deep networks, are based on constructing a high number of neural networks with or without prior knowledge for training. In order to achieve the maximum performance, via extracting from the X input vector matrix the most important features for learning, one can apply kernel functions in 1 or 3 dimensions of a matrix with the use of convolutional layers. Sub-sampling techniques can be applied via using the median, maximum or L2-norm functions to determine the ultimate features to lead to the best classification regime. The output from these multilayer architectures is concatenated and merged into a fully connected neural network. The output of this ML scheme will be a set of weights to be evaluated on the tested data. Other popular types of architectures such as encoders-decoders consist of encoding layers, which extract only the most important compressed type of information and a decoding layer which increases the features

which contribute the most to the classification regime. Other formulations, such as Recurrent neural networks (RNN), or the long short-term memory, LSTM have the ability to learn recursively via having all inputs relate to each-other. Moreover, RNNs contain cyclic connections, thus making them a very powerful tool for the modeling of sequence data. Bidirectional LSTM (BLSTM) networks have the ability to operate on an input sequence in both directions in order to make a decision for the current input. These architectures as we will investigate have contributed greatly towards the prediction of many biological mechanisms. Figure 7 depicts three widely used deep learning architectures (a) CNN, (b) encoder decoder and (c) LSTM.

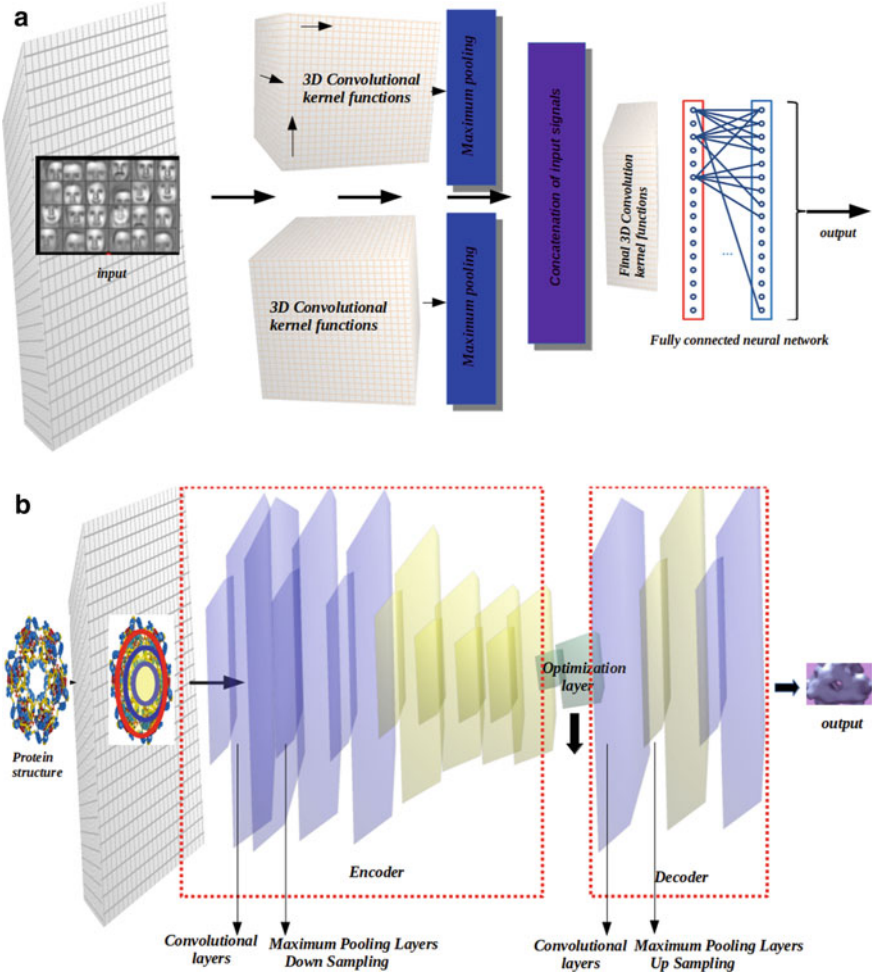

Fig. 7 Illustrating 3 widely used deep learning architectures **a** CNN, **b** encoder decoder and **c** LSTM

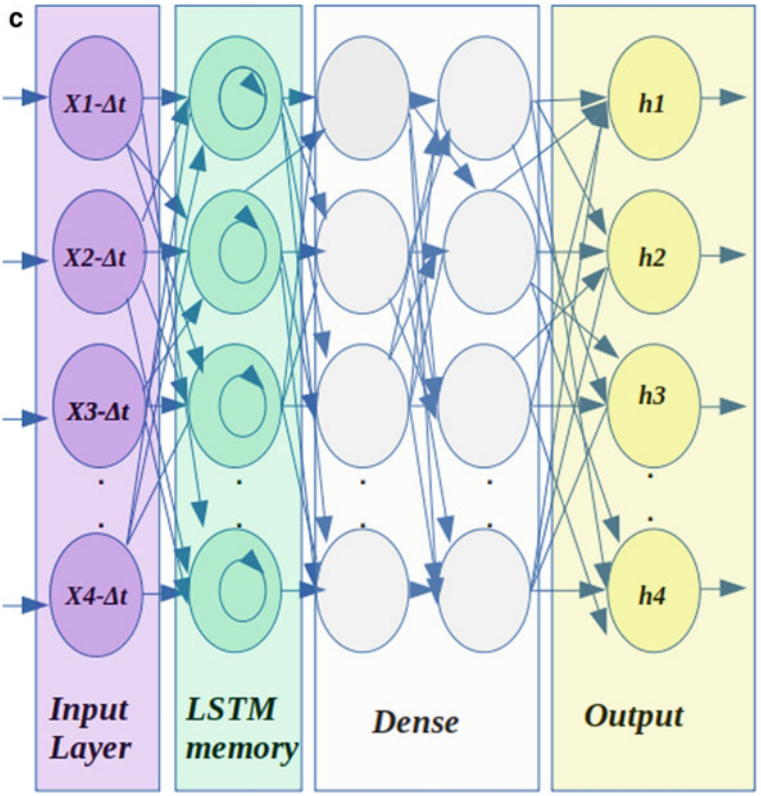

Fig. 7 (continued)

4 Applications of ML in Genomics

High throughput protocols for genome wide studies are constantly evolving. We can now measure and determine precisely the condition of each cell on a living organism, determine the post and pre-transciptional mechanisms and the genetic variations that are being introduced. These vast plethora of information needs novel ML schemes, in order to extract hidden fundamental biological functions such as predict the cell-states, determine specific gene signatures and the pathways involved and understand the processes that govern diseases. We will focus on the analysis of the current ML architectures for the prediction of scRNA-seq cell states extraction of trajectories and construction of diffusion maps. Graph-embedding approaches will be analyzed, and specific examples will be investigated. The same principles and algorithms will be examined for other important learning applications, such as RNA structure prediction in two and three dimensions as well as the prediction of protein-RNA/DNA interactions. We will thus investigate how the previously examined mathematical formulations hold in practice.

4.1 Deep Learning Approaches Towards the Prediction of the Cell State and Fate

The current approaches of scRNA-seq technologies have paved the way for the understanding of the heterogeneity among cells, their current cell state and function as a whole to thus compose the plethora of organs. The first step in a scRNA-seq experiment involves the isolation of individual cells, where currently various approaches have been developed. These involve the isolation of single cells with FACS or dedicated microfluidic devises such as 10X (10X Genomics, Pleasanton, CA) and Drop-sec [72], laser microdisections [25] or the sorting and capturing of cells via tagging specific cell membrane markers such as in CITE-seq [102]. Different types of biochemistry involve mainly the way the cell lysis and bead-capture are performed, including various amplification protocols, strand specificity as well with the way different bar-coding techniques have been developed for using appropriately UMIs (unique molecular identifiers) to distinguish the per cell transcriptome and genome state. Another difference is that certain protocols can produce full length transcript via reverse transcriptase such as (Smart-seq2 [83], SUPeR-seq [27], and MATQ-seq [98], whereas other methods can produce and capture the 3' end such as in Drop-seq, Seq-Well [32], DroNC-seq [36], SPLiT-seq [92] or the 5′-end such as in STRT-seq [42]) of the transcripts.

The analysis of such data is a complicated task, the data is sparse as many dropouts are included. Furthermore, the curse of dimensionality follows throughout the analysis thus ML techniques need to be established to cope with such processes. The issue of drop-outs in sc-RNAseq data can be resolved via imputation techniques. MAGIC [110] computes a distance cell by cell matrix which learns an affinity matrix via using a Gaussian kernel function. These affinities are normalized and compose a Marcov model which defines transition matrix across cells. The original data are multiplied via an exponential Markov matrix thus smoothing the data. Deep Impute [4] splits the genes into subsets and builds sub-networks in a divide-and-conquer approach. Each sub-neural network learns the relationship of certain category of genes. A sub neural network consists of layers of 256 neurons activated via a Relu function. A 20% dropout layer is used to accumulate only the important information to be set as an output to a dense fully connected network. Deep learning Imputation model DISC [39] is a semi-supervised learning (SSL) method for Single Cell transcriptomes. This can learn the structure of genes and cells from sparse data. This method uses deep autoencoders and an RNN, with an SSL approach to train the model parameters via learning information from both positive- and zero-count genes, which can be treated as labeled and unlabeled data, respectively. Similarly, GraphSCI [88] uses a graph convolutional network and an autoencoder to impute the dropout events in scRNA-seq by systematically integrating the gene expression with gene-to-gene relationships. AutoClass [62] integrates an autoencoder which at first reduces the data dimension and compresses the input data where the decoder expands the data dimension and reconstructs the original input to be used to a neural network classifier

for learning. Another similar ML scheme is DCA [26] which uses a deep autoencoder in an unsupervised manner to model a reconstruction error as the likelihood of the distribution of the noise model, instead of reconstructing the input data itself. During training the classifier learns from the count distribution, the overdispersion and sparsity per gene-sets and tries to reconstruct this error. Zero inflation methodologies such as ZIFA [84] consider the mean level of non-zero expressed genes (the log read counts) as μ and the dropout rate as P_o where the relationship is modeled via $P_o=\exp(-\lambda\mu^2)$, having λ be a fitted parameter shared across genes.

ML applications have been widely adopted to predict and model the cellular states and configure the cell trajectories and the cell-to-cell interaction. These systems adopt force directed graphs which consider edges as applying forces (repulsive or attractive) to nodes thus, edge-weights are applied to define a relation between any node. GraphDDP (Graph-based Detection of Differentiation Pathways) [20] is an example of a force directed layout graph. The Layout algorithms starts via creating an unweighted instant graph G=(V,E), having V to be a set of vertices and E the set of edges where also a distance matrix is formed based on the expression profile table where pairwise similarities are examined between cells. The pairwise similarities are estimated using k-nearest neighbor algorithm. Furthermore, the notion of particle dynamics is adopted where each particle is connected by springs of strength k_{ij}, and the best layout is the one that minimizes the total energy of the system. To derive the minimum energy a two-dimensional Newton–Raphson method [23] is applied. This allows only one particle moving from one point to the other while the other particles are stable. Figure 8 presents a force directed layout alogirthm, using as

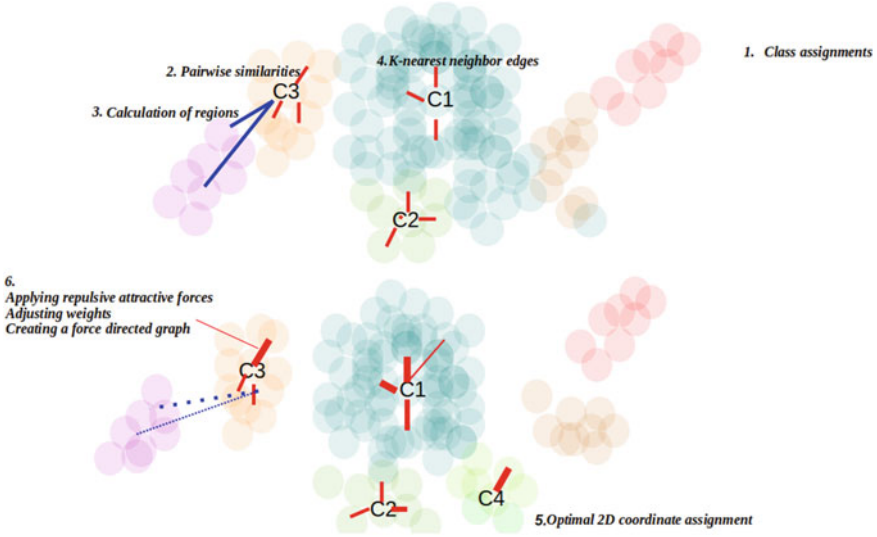

Fig. 8 Example of drawing a force directed graph for scRNA-seq to construct cell trajectories (colored spheres demonstrate different cell clustering) with repelling attractive forces red lines, weights are denoted with the width of the lines

input scRNA-seq data.

Haghverdi et al. propose diffusion maps as a tool for analyzing single-cell differentiation data in their study [37]. Monocle 2 [86], is an unsupervised algorithm that uses reversed graph embedding to describe multiple cell fate decisions. Monocole 2 starts via identifying genes in a pathway. The procedure proceeds by selecting the differentially expressed genes between the clusters of cells as derived from tSNE. RGE finds a mapping between the high dimensional gene expression space and a lower dimensional via learning a principal curve that passes through the "middle" of a data set while learning the graph properties in the lower dimensions.

Apart from deciphering the transcriptome per cell, new technological advances have enabled to distinguish simultaneously the chromatin state per cell via profiling the chromatin accessibility thus enabling to distinguish the per cell type composition as well with the cell-to-cell variation. The idea behind such method is that the DNA is wrapped around nucleosomes consisting of histone octamers, except from when genes need to be transcribed, where chromatin regions become accessible for transcription factors and other regulatory elements to bind. The ATAC-seq protocol uses a hyperactive Tn5 transposase which integrates adaptors by binding to the DNA and a fragmentation stage where the transposase is released [13, 79]. In a scRNA-seq application a single-cell assay for Transposase-Accessible Chromatin is inserted into the lysis buffer of the microfluidic device and this enables to simultaneously capture the open chromatin regions and mRNA per cell.

Other methods can capture the single-cell proteome with multiplexed LC-ESI-MS/MS to quantify proteins from single cells while using an isobaric carrier labelling [100]. Other methods include capturing of cells that contain specific cell membrane markers via a method called CITE-seq [102], where conjugating antibodies use streptavidin–biotin to oligonucleotides (oligos) and can be captured by oligo-dT primers. After cell lysis in droplets, both the cellular mRNAs and antibody-derived oligos can anneal via their 3′ poly-A tails to Drop-seq beads to be sequenced. Other novel methods such as STAMP [11] are able to identify the binding sites of full-length RBPs per cell by using C-to-U RNA editing via fusing APOBEC to the protein of interest. When combining this with single-cell sequencing one can capture the RBP binding sites via interrogating the editing signal per cell type. When APOBEC is also fused to the Ribosomal proteins then this method can be used for single cell Ribosome profiling. This plethora of methods can be multiple integrated to re-cluster cells based not only on the levels of RNA per cell but also according to the accessible chromatin regions or proteome. This has been demonstrated in [38], where a weighted nearest neighbor analysis is used to integrate multimodal single-cell data. A weighted nearest neighbor graph can be generated which denotes the most similar cells based on a combination of protein and RNA cell similarities which can be used for downstream analysis and visualization with tSNA or UMAP. Recent methodologies involve modeling of the RNA kinetics per cell using metabolic labeling techniques. A popular biochemistry involves 4sU treatment and thiol(SH)-linked alkylation of the RNA [40]. 4sU gets incorporated into novel transcripts when followed by IAA treatment this creates T>C mutations of the transcripts at the specific time point upon treatment, thus this can be used to estimate the RNA kinetics. When this

is coupled with a technique called velocito [56] then one can determine a multitude of rates such as RNA transcription, splicing, translation and decay per cell. Velocito is a computational technique which determines the spliced to non-spliced ratio of genes per cell based on the read counts that span exon-introns where the introns can be captured due to miss-priming. When combining this with the trajectory methodologies this is a powerful technique for deciphering the cell dynamics of a system on determining the various cell lineages. An interesting application is using vector field functions, to calculate an RNA Jacobian which is a cell by gene tensor, describing a gene regulatory network in each cell. An interesting approach is to use such metrics to build regulatory networks across different cell types to cluster different cell types within a manifold. Such computational frameworks have been widely adopted in state-of-the-art techniques such as scNT-seq [85] a UMI-based scRNA-seq method that combines metabolic RNA labeling with a chemically induced recoding of 4sU to a cytosine analog to simultaneously measure new and old transcriptomes from the same cell or scEU-seq [5] which is a method to sequence mRNA labeled transcripts with 5-ethynyl-uridine (EU) in single cells. When combining a multitude of single-cell experiments such as scRNA-seq with CITE-seq [102], REAP-seq [81], ECCITE-seq [99] one can model the cell states via extending the application of RNA velocity. The RNA and protein velocity fields enable single cell RNA and protein quantification techniques [34]. In this method a Gaussian kernel is used to determine the net velocities at regular grid-points thus the RNA and protein velocity fields are associated with each cell type. Figure 9 represents single cell RNA-seq velocities while modeling the velocities with vector fields using a Jacobian matrix a method adopted mainly in Dynamo [7]. Figure 10 depicts images from the human cell heart

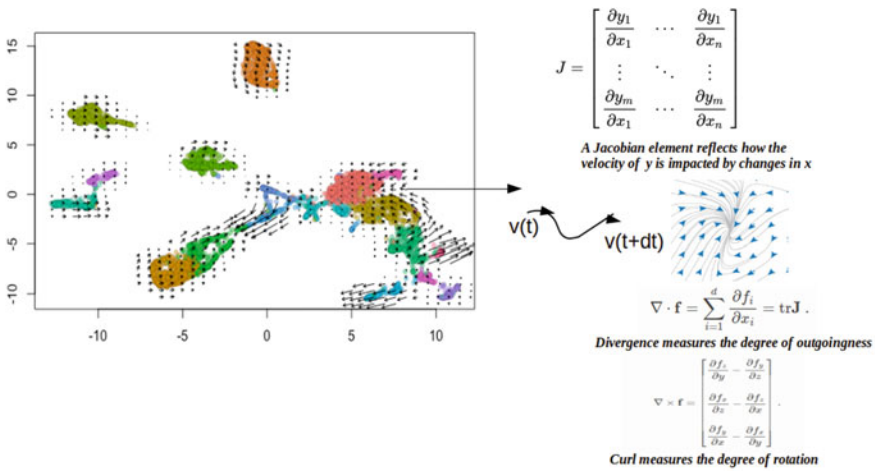

Fig. 9 Single cell RNA-seq velocities while modeling the velocities with vector fields using a Jacobian matrix a method adopted mainly in Dynamo [7]. The velocities are generated from rerunning the tutorials in https://ucdavis-bioinformatics-training.github.io/2020-Advanced_Single_Cell_RNA_Seq

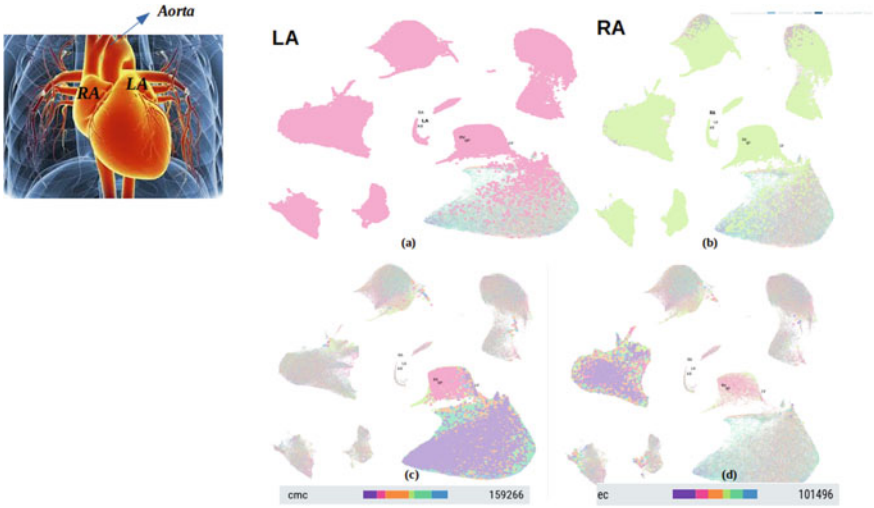

Fig. 10 Images from the human cell heart atlas [65], where samples are obtained from dissections of the right and left atrium (pink and green dots), while demonstrating the composition of these cells in cardiomyocites and endothelial cell sub-populations. Cell counts can be seen in the colored bar-plots

atlas, where samples are obtained from dissections of the right and left atrium (pink and green dots), while demonstrating the composition of these cells in cardiomyocites and endothelial cell sub-populations.

4.2 (ML) Applications for Predicting the RNA Properties and RNA-Protein Interactions

RNA molecules have a multitude of interactions with RBPs, enzymes, ncRNAs and even metabolites. Throughout the life of an RNA, it adopts a different structure thus facilitating the specific factors to bind at specific time points that are part of the post transcriptional regulation control. In the last decade, we have the ability to capture the binding of RBPs genome wide, and the RNA structure via novel RNA probing techniques and novel more robust immunoprecipitation of RNA-protein fragments. In brief to capture the RBP binding on any RNA, methods such HITS-CliP [22], iCLIP [44], PAR-CliP [21] introduce at the binding position of the RNA either mutations-deletions, or RT-stops during cDNA synthesis or specific mutations such as T>C, while introducing photosensitive analogs of uridines where upon cDNA synthesis they have the ability to introduce these mutations only at the binding positions. Furthermore, determining *in vivo* the RNA structure can be performed via utilizing specific enzymes that can cleave either single stranded (ss-RNAs) or double-stranded (ds-RNAs) as in PARS-seq [48] or chemical reagents can be incorporated which can

either cause mutations on specific residues of RNAs [101], such as in icSHAPE [29], SHAPE-seq [69], DMS-seq [93] or cross-linking of RNA-RNA to determine ds-RNA regions such as in PARIS-seq [70] or LIGR-seq [97].

The prediction of the exact RNA structure via combining the information of probing the RNA needs novel ML approaches to enhance the performance. The previous applications and ML schemes can be applied for the prediction of RNA. A graph-based ML method is E2Efold [18] which can predict an RNA base-pairing matrix. In this model, the sequence information is used for structure prediction using a transformer encoder which also shows the dependency between nucleotides, the second part uses a multilayer network based on an unrolled algorithm [76] used for solving constrained optimization problems and a 2D convolutional network is used after concatenating the output information from the two deep learning modules. The unrolled algorithm architecture cascades a neuron with parameters θ where each θ depends on the previous one. Zhang et al. in their recent study, describe a novel RNA secondary structure prediction algorithm which uses a convolutional neural network model combined with a dynamic programming to improve the accuracy of structure prediction regarding large RNA molecules [121]. The training happens while using experimental RNA sequences and structure data in deep convolutional networks, which are then used to extract implicit features from large-scale data with goal to predict the RNA base pairing probabilities. RNA-As-Graphs (RAG) approach [74] reduces a 2D structure RNA complexity using graph theory and offers new tools to study RNA structure. More precisely in RAG-sampler candidate tree graph topologies are generated via a random forest ML approach. Monte Carlo simulated annealing methods are embedded to convert a 2D tree to a scaled 3D tree graph [3, 109]. Furthermore, these are scored via potential functions from already known RNA structures. More advance approaches make use of coarse-grained molecular dynamics [63] to address the prediction of the 3D RNA structure. These methods are mainly based on a simplified form for solving the Poisson Boltzmann equation (Eq. 19) such as in SimRNA [10]. Representative examples of Monte Carlo simulations regarding sub-domains of an RNA 3D sub-structure are presented in Fig. 11.

RNAcontact [104] predicts RNA inter-nucleotide 3D closeness with deep residual neural networks. The input features are the covariance information from multiple sequence alignments and the predicted secondary structure. More precisely, this ML scheme uses three 2D convolution layers with five residual blocks, each containing two convolution layers where also a shortcut connection is established in each residual block. The output layer is a 2D convolution layer. The final output of the network is a 2D probability matrix which describes the 3D RNA contacts. Another method involves a deep Generative Model, Monte Carlo Tree Search [87]. A sequence is used as input and the output consists of the predicted 3D structure. The models are trained with PDB structures represented by Euclidian distances between nucleotide pairs. The position of each nucleotide is determined by five selected atoms thus forming 5 $\times$ 5 Euclidean distances for each nucleotide pair. These are encoded into K discrete distance classes by a VQ-VAE where a Deep Neural Network (DNN) is used to predict probability values for the distance classes. Based on the sequence composition

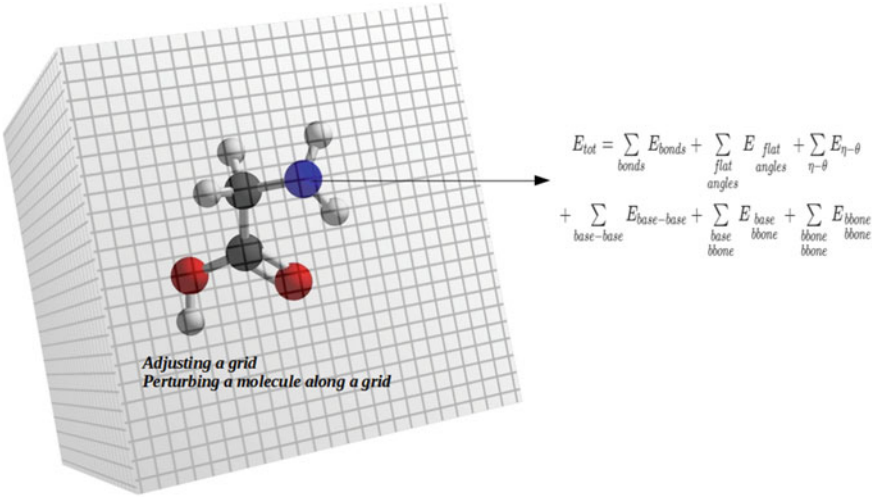

Fig. 11 Examples of Monte Carlo simulations regarding sub-domains of an RNA 3D sub-structure. The molecules of the RNA are placed along a grid where per grid-point the total energies per molecule are estimated while perturbing the molecules. The molecules perturbations will converge to a lowest total energy or most stable state

a score model selects the most promising structures. MXfold2 [95] is a deep neural network architecture that calculates folding scores while these are combined with Turner's nearest-neighbor free energy parameters. Training of the model while taking into account thermodynamic regularization ensures that the folding scores and the calculated free energy are as close as possible thus maximizing precision.

Apart from RNA structure prediction many models have been developed to determine the exact 3D protein folding or the residues of importance as this requires immense computational power (Antoniou et al. 2020). The latest most important achievement is AlphaFold [47] which makes use of ML to accurately predict with high computational speed the 3D protein structure in great detail. AlphaFold employs a "spatial graph", to map the proteins where residues are the nodes and edges connect the residues in close proximity. A neural network system attempts to interpret the structure of the graph while using evolutionary related sequences, multiple sequence alignment (MSA), as well with representation of amino-acid pairs to refine the graph. DeepFRI [33] is a Graph Convolutional Network that can predict the protein functions by combining sequence features and protein structures in terms of contact matrices while assembling a graph. It uses a two-stage architecture having as input the protein structure and sequence and outputs probabilities which identify specific residues via using a gradient-weighted Class Activation Map (grad-CAM). For the sequence input a recurrent neural network architecture with long short-term memory (LSTM-LM) is designed. For the structure the authors use graph convolutional networks.

The previous paragraphs demonstrated the latest ML techniques for predicting the RNA and protein structures in detail thus this information can be used to

develop ML approaches towards prediction of RBP-RNA interactions. PrismNet [105] is such an example where it employs deep learning that integrates experimental *in vivo* RNA structure data from probing experiments and RBP binding data (RNA bound sequence context) from CLiP methods for matched cells to accurately predict RBP-RNA binding. More precisely convolutional layers and two-dimensional residual blocks are connected by max-pooling to define useful sequence and structural elements. The convolutional channels can be adaptively re-calibrated via a squeeze and excite module. NucleicNet [58] predicts RBP-RNA interactions using a multitude of physicochemical characteristics for the RBPs (hydrophobicity, molecular charges, and accessibility surfaces) as well with the sequence content as determined from CliP experiments. A deep residual network is trained using the previous characteristics and the networks is optimized via back-propagations of the categorical cross entropy loss. iDeep [77] is a deep learning-based framework which is composed of a hybrid convolutional neural network and a deep belief network to predict the RBP interaction sites and motifs on RNA based on sequence of the RBP binding sites while using the RNA structure and region as input information. On the other hand, aPRBind [67] is trained to decipher interaction points with the RNA on the protein surface using the amino acid sequence and structure features of the RBPs including residue dynamics information and residue–nucleotide propensity as extracted from the predicted 3D proteins structures by I-TASSER [119]. In general ML potentials have become an important tool in modeling RNA-protein complexes. Ko et al. presented a fourth-generation high-dimensional neural network potential that combines cartesian coordinates, symmetry functions, atomic electronegativities with accurate atomic energies and atomic charges [50]. DFT calculations are used to decipher electrostatic potentials and the positions of the atoms. The atomic electronegativities are local properties of atom nuclei described as a function of the atomic environments using atomic neural networks. Atomic charges, are represented by Gaussian charge densities of width σ_i taken from the covalent radii of the respective elements surrounding, taking into account all the conformations. Deepnet-rbp [122] uses RNA tertiary structure motifs as predicted by JAR3D (a computational framework that extracts probable structural motifs in the hairpins and internal loop regions using RNA 3D Motifs Atlas (R3DMA)) with the sequence information into a deep learning scheme which uses restricted Boltzmann machines (RBMs) based on Markov random fields module to predict RBP binding sites [55].

4.3 Drug Repurposing a Novel Tool in Medicine

Predicting the outcome of a drug clinical trial based on prior information from already known approved drugs seems to be a promising method of choice for pharmaceutical R&D and a valuable resource for clinical trials that reduces the risk from unsatisfactory side-effect [94, 112, 114]. Deep-Learning methods have gained a wide range of applications towards prediction of cytotoxicity and efficacy of drugs. These methods are based on protein ligand docking approaches, while also using novel encoding

techniques for the 3D structure and physiochemical properties of drugs as well as using cytotoxicity testing, efficacy measurements (IC 50) [14], pathway analysis and gene-gene interactions together with genomics (mutation profiles, RNA abundance, methylation profiles etc.). Drug-ligand docking predictions are important tasks for measuring the efficiency of a specific drug while predicting docking scores using quantitative structure–activity relationship. QSAR models use a 3D "inductive" descriptor [31], support vector machines or random forests along with the conformal predictors [1, 107]. QSAR descriptors (such as in the description of a 2D molecular fingerprint), and iterative fast random sampling of docking databases can be used as input to a Deep Learning module to predict docking scores of yet unprocessed database entries at each iteration step such as in [31]. PaccMann [73] predicts cancer compound sensitivity with the use of neural networks. This model uses as input a SMILE structure encoding for drug molecules, gene expression profile of a cancer cell and predicts IC50 and sensitivity value. For extracting the most useful information from the SMILE embedding an attention-based encoder is used. PrOCTOR [30] from the other hand evaluates cytotoxicity using as input smile embeddings of drugs and the drug target smile structure information using a structured embedding as derived from a designed platform named TAPE [89]. Precompiled gene-gene interactions together with the median expression of the gene targets in 30 different tissues, from Genotype-Tissue Expression (GTEx) project [68] as well with the mutation frequency of the target gene as extracted from the Exome Aggregation Consortium (ExAC) [60] database and the molecular properties of the drug are used in a Random Forest application. In addition, the similarity of drugs in the database with the introduced drugs are evaluated using Lipinski's Rule 5 [6]. DRIAD Drug Repurposing in Alzheimer's Disease (AD) [91] is a ML approach used for drug repurposing of drugs in Alzheimer's disease. DRIAD was applied to lists of differential genes arising from perturbations of 80 FDA-approved drugs in various human neural cell cultures and produced a ranked list of possible repurposing candidates. The input of the framework uses mRNA expression profiles from human brains at various stages of AD progression as well with a dataset of differential expressed genes upon small molecules tested in different brain cell types such as neurons, astrocytes, and oligodendrocytes. PREDICT [35] is used for large-scale prediction of drug indications as it uses as input data, features from drug repositories, drug targets to GO terms, extracts disease-disease similarities drug to disease associations and also extracts drug to drug similarities. All these are combined to predict a score.

4.4 Prediction of Neo-Antigens

The role of neo-antigens as a method for immunotherapy in certain tumor types has been recently gaining the attention of scientists. RNA vaccination with the RNA of appropriate neo-epitopes will enable T cell therapy via re-engineering T-cells to target such epitopes on the membrane of tumor cells. T cells have the ability to recognize peptides presented on the major histocompatibility complex (MHC) of tumor

cells. T cells are primed by antigen presenting cells (APC). The APCs take up tumor antigens and process them into smaller peptides via the proteasome. They usually consist of peptides of 9-12 amino acids. These are transported from the cytosol to the endoplasmic reticulum to be potentially recognized and bound on MHC-I. Antigens can also arise from extracellular sources within a tumor environment such as from necrotic or apoptotic cells and other vesicles. Any protein produced in a tumor cell that has an abnormal structure due to mutations or transcriptome aberrations can act as a tumor antigen. Transcriptome aberrations such as alternative exon splicing events, intron retention, premature transcription ending, translation read-through on CDS and open reading frames (uORF) or introduction of stop codons via mutations that might cause altered proteins can cause such events. To identify such mechanisms bioinformatics and ML algorithms have been employed. NeoPredPipe (Neoantigen Prediction Pipeline) [96] is such a framework. The pipeline starts with determining non-synonymous variants with ANNOVAR [116] which processes the loci via prioritizing the exonic variants. The program determines if HLA haplotypes have been provided by the user. The user is also able to specify the epitope lengths to conduct predictions but also cross-reference with normal peptides utilizing PeptideMatch [16]. For the predictions, the method employs NetMHCpanII [46] an ML framework which uses artificial neural networks to predict the binding of peptides to any MHC molecule of known sequence. The network has been trained on 180,000 quantitative binding data and MS derived MHC eluted ligands and covers 172 MHC. NetMHCIIpan which is used from the pipeline can predict the binding of peptides to MHC class II molecules. It uses neural networks that can predict peptide binding affinities for all MHC molecules of known protein sequences. The training set includes a data set of more than 100.000 quantitative peptide binding measurements from IEDB (https://www.iedb.org/) covering 36 HLA-DR, 27 HLA-DQ, 9 HLA-DP, as well as 8 mouse MHC-II molecules. Similarly, in ProGeo-neo [62] tumor specific antigens can be captured from mRNA expression studies while using netMHCpan (v.4.0) to predict binding of peptides to MHC-I class of molecules. This is combined with a cross reference of MHC-peptides with mass spectrometry proteomics while checking for T-cell recognition. SVRMHC [115] is a support vector machine regression (SVR)-based method for modeling peptide-MHC binding. Furthermore, this ML scheme can use three different kernel functions (linear, polynomial and RBF) in combination with "sparse encoding", and "11-factor encoding [66]" for the sequence encodings. Neopepsee [49] is an ML neoantigen prediction program for next-generation sequencing data. It uses as input RNA-seq data and a list of somatic mutations to extract mutated peptide sequences and gene expression levels. In brief four different classification methods have been adopted such as Gaussian naïve Bayes (GNB), locally weighted naïve Bayes (LNB), random forest (RF), and support vector machine (SVM) while training using 500 iterations with 10-fold cross-validation. The input data use 2k-1-mer amino acid sequences as derived after isoform quantification of the RNA while in parallel identifying mutations. Initially HLA alleles from the RNA-seq data are separated and used in an MHC binding affinity prediction module with NetCTPpan [103] *NetCTLpan* derives a prediction using a weighted sum of three individual prediction values for MHC class I affinity, TAP transport efficiency, and

C-terminal proteasomal cleavage. Moreover, with the constant evolution of scRNA-seq technologies we can capture T-cells in a tumor microenviroment via using a TCR-barcode such as in [15] to identify the RNAs per T-cells and thus identify T cell dysfunction programs in tumor-infiltrating lymphocytes.

5 Discussion

The aim of this chapter was to describe and analyze applications of ML along with the fundamental mathematic principles that constantly aid biology and medicine and have become a fundamental aspect for understanding the mechanisms of life. These ML applications seem to be important as the constant evolution of biochemistry and NGS high throughput techniques are constantly increasing. Thus, we have focused on some state-of-the-art applications addressing current biological questions such as predicting cell states and estimating cell trajectories from large high throughput state of the art single cell technologies thus empowering the modeling of the human cell atlas. Moreover, we have investigated how ML applications can be used towards predicting the exact 3D dimensional structures of RNA and proteins and thus unveiling and modeling their interactions in 3D with high accuracy. Furthermore, we have examined how the same algorithms can be applied to clinical oncology in problems such as drug repurposing tasks for deriving appropriate treatments per specific cases while estimating the outcome of clinical trials and improving prediction of drug effectiveness and safety. Moving along the same lines, we have seen how ML applications can be used to predict neo-antigens and neo-epitopes describing the basics behind the promising new era of immunotherapy and RNA vaccination. We believe that ML applications combined with the constant advances of biochemistry and NGS will govern the new line of treatments moving along the road of personalized medicine with higher efficacy against diseases while understanding the mechanisms that employ them.

Acknowledgements VGG and his colleagues received financial support from the following grants: National Public Investment Program of the Ministry of Development and Investment/General Secretariat for Research and Technology, in the framework of the Flagship Initiative to address SARS-CoV-2 (2020ΣΕ01300001); Horizon 2020 Marie Sklodowska-Curie training program no. 722729 (SYNTRAIN); Welfare Foundation for Social & Cultural Sciences, Athens, Greece (KIKPE); H. Pappas donation; Hellenic Foundation for Research and Innovation (HFRI) grants no. 775 and 3782, NKUA-SARG grant 70/3/8916 and H. Pappas donation. This study was also co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T2EDK-02939).

References

1. Ammad-Ud-Din, M., Khan, S.A., Malani, D., Murumägi, A., Kallioniemi, O., Aittokallio, T., Kaski, S.: Drug response prediction by inferring pathway-response associations with kernelized Bayesian matrix factorization. Bioinformatics **32**, i455–i463 (2016)
2. Antoniou, N., Lagopati, N., Balourdas, D.I., Nikolaou, M., Papalampros, A., Vasileiou, P., Myrianthopoulos, V., Kotsinas, A., Shiloh, Y., Liontos, M., Gorgoulis, V.G.: The role of E3, E4 ubiquitin ligase (UBE4B) in human pathologies. Cancers **12**, 62 (2019)
3. Argyrou, M., Andreou, M., Lagopati, N., Baka, I., Vamvakas, I., Lyra, M.: Patient specific dosimetric calculations obtained by planar images and Monte Carlo simulation in 111In octreotide therapy. Case Rep. Images Surg. **1**, 1–5 (2018)
4. Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X., Garmire, L.X.: DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. Genome Biol. **20**, 1–14 (2019)
5. Battich, N., Beumer, J., de Barbanson, B., Krenning, L., Baron, C.S., Tanenbaum, M.E., Clevers, H., van Oudenaarden, A.: Sequencing metabolically labeled transcripts in single cells reveals mRNA turnover strategies. Science **367**(6482), 1151–1156 (2020)
6. Benet, Leslie Z., Hosey, Chelsea M., Ursu, Oleg, Oprea, Tudor I.: BDDCS, the rule of 5 and drugability. Adv. Drug Del. Rev. **101**(2016), 89–98 (2016)
7. Bergen, V., Lange, M., Peidli, S., Wolf, F.A., Theis, F.J.: Generalizing RNA velocity to transient cell states through dynamical modeling. Nat. Biotechnol. **38**, 1408–1414 (2020)
8. Biau, G.: Analysis of a random forests model. J. Mach. Learn. Res. **13**, 1063–1095 (2012)
9. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
10. Boniecki, M.J., Lach, G., Dawson, W.K., Tomala, K., Lukasz, P., Soltysinski, T., Rother, K.M., Bujnicki, J.M.: SimRNA: a coarse-grained method for RNA folding simulations and 3D structure prediction. Nucl. Acids Res. **44**, e63–e63 (2016)
11. Brannan, K.W., Chaim, I.A., Marina, R.J., Yee, B.A., Kofman, E.R., Lorenz, D.A., Jagannatha, P., Dong, K.D., Madrigal, A.A., Underwood, J.G., Yeo, G.W.: Robust single-cell discovery of RNA targets of RNA-binding proteins and ribosomes. Nat. Methods **18**, 507–519 (2021)
12. Brier, G.W.: Verification of forecasts expressed in terms of probability. Mon. Weather Rev. **78**, 1–3 (1950)
13. Buenrostro, J.D., Wu, B., Litzenburger, U.M., Ruff, D., Gonzales, M.L., Snyder, M.P., Chang, H.Y., Greenleaf, W.J.: Single-cell chromatin accessibility reveals principles of regulatory variation. Nature **523**(7561), 486–490 (2015)
14. Caldwell, G.W., Yan, Z., Lang, W., Masucci, A.J.: The IC50 concept revisited. Curr. Top. Med. Chem. **12**, 1282–1290 (2012)
15. Caushi, J.X., Zhang, J., Ji, Z., Vaghasia, A., Zhang, B., Hsiue, E.H.C., Smith, K.N.: Transcriptional programs of neoantigen-specific TIL in anti-PD-1-treated lung cancers. Nature **596**(7870), 126–132 (2021)
16. Chen, C., Li, Z., Huang, H., Suzek, B. E., Wu, C. H., and UniProt Consortium: A fast peptide match service for UniProt knowledgebase. Bioinformatics **29**, 2808–2809 (2013)
17. Chen, G., Ning, B., Shi, T.: Single-cell RNA-seq technologies and related computational data analysis. Front. Genet. **10**, 317 (2019)
18. Chen, X., Li, Y., Umarov, R., Gao, X., Song, L.: RNA secondary structure prediction by learning unrolled algorithms (2020). arXiv preprint. arXiv:2002.05810
19. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. Proc. Natl. Acad. Sci. **102**, 7426–7431 (2005)
20. Costa, F., Grün, D., Backofen, R.: GraphDDP: a graph-embedding approach to detect differentiation pathways in single-cell-data using prior class knowledge. Nat. Commun. **9**, 1–8 (2018)
21. Danan, C., Manickavel, S., Hafner, M.: PAR-CLIP: a method for transcriptome-wide identification of RNA binding protein interaction sites. In: Post-Transcriptional Gene Regulation, pp. 153–173. Humana Press, New York, NY (2016)

22. Darnell, R.B.: HITS-CLIP: panoramic views of protein-RNA regulation in living cells. Wiley Interdiscip. Rev. RNA **1**, 266–286 (2010)
23. Dence, T.: Cubics, chaos and Newton's method. Math. Gaz. **81**, 403–408 (1997)
24. Dou, L., Li, X., Ding, H., Xu, L., Xiang, H.: iRNA-m5C_NB: a novel predictor to identify RNA 5-Methylcytosine sites based on the Naive Bayes classifier. IEEE Access **8**, 84906–84917 (2020)
25. Ellis, P., Moore, L., Sanders, M.A., Butler, T.M., Brunner, S.F., Lee-Six, H., Osborne, R., Farr, B., Coorens, T.H.H., Lawson, A.R.J., Cagan, A., Stratton, M.R., Martincorena, I., Campbell, P.J.: Reliable detection of somatic mutations in solid tissues by laser-capture microdissection and low-input DNA sequencing. Nat. Protoc. **16**, 841–871 (2021)
26. Eraslan, G., Simon, L.M., Mircea, M., Mueller, N.S., Theis, F.J.: Single-cell RNA-seq denoising using a deep count autoencoder. Nat. Commun. **10**, 1–14 (2019)
27. Fan, X., Zhang, X., Wu, X., Guo, H., Hu, Y., Tang, F., Huang, Y.: Single-cell RNA-seq transcriptome analysis of linear and circular RNAs in mouse preimplantation embryos. Genome Biol. **16**, 1–17 (2015)
28. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. **20**, 1–81 (2019)
29. Flynn, R.A., Zhang, Q.C., Spitale, R.C., Lee, B., Mumbach, M.R., Chang, H.Y.: Transcriptome-wide interrogation of RNA secondary structure in living cells with icSHAPE. Nat. Protoc. **11**, 273–290 (2016)
30. Gayvert, K.M., Madhukar, N.S., Elemento, O.: A data-driven approach to predicting successes and failures of clinical trials. Cell Chem. Biol. **23**, 1294–1301 (2016)
31. Gentile, F., Agrawal, V., Hsing, M., Ton, A.T., Ban, F., Norinder, U., Gleave, M.E., Cherkasov, A.: Deep docking: a deep learning platform for augmentation of structure based drug discovery. ACS Central Sci. **6**, 939–949 (2020)
32. Gierahn, T.M., Wadsworth, M.H., Hughes, T.K., Bryson, B.D., Butler, A., Satija, R., Fortune, S., Love, J.C., Shalek, A.K.: Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. Nat. Methods **14**, 395–398 (2017)
33. Gligorijević, V., Renfrew, P.D., Kosciolek, T., Leman, J.K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B.C., Fisk, I.M., Vlamakis, H., Xavier, R.J., Knight, R., Cho, K., Bonneau, R.: Structure-based protein function prediction using graph convolutional networks. Nat. Commun. **12**, 1–14 (2021)
34. Gorin, G., Svensson, V., Pachter, L.: Protein velocity and acceleration from single-cell multiomics experiments. Genome Biol. **21**, 1–6 (2020)
35. Gottlieb, A., Stein, G.Y., Ruppin, E., Sharan, R.: PREDICT: a method for inferring novel drug indications with application to personalized medicine. Mol. Syst. Biol. **7**, 496 (2011)
36. Habib, N., Avraham-Davidi, I., Basu, A., Burks, T., Shekhar, K., Hofree, M., Choudhury, S.R., Aguet, F., Gelfand, E., Ardlie, K., Weitz, D.A., Rozenblatt-Rosen, O., Zhang, F., Regev, A.: Massively parallel single-nucleus RNA-seq with DroNc-seq. Nat. Methods **14**, 955–958 (2017)
37. Haghverdi, L., Buettner, F., Theis, F.J.: Diffusion maps for high-dimensional single-cell analysis of differentiation data. Bioinformatics **31**, 2989–2998 (2015)
38. Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W.M., III., Zheng, S., Butler, A., Lee, M.J., Wilk, A.J., Darby, C., Zager, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E.P., Jain, J., Srivastava, A., Stuart, T., Fleming, L.M., Yeung, B., Rogers, A.J., McElrath, J.M., Blish, C.A., Gottardo, R., Smibert, P., Satija, R.: Integrated analysis of multimodal single-cell data. Cell **184**, 3573-3587.e29 (2021)
39. He, Y., Yuan, H., Wu, C., Xie, Z.: DISC: a highly scalable and accurate inference of gene expression and structure for single-cell transcriptomes using semi-supervised deep learning. Genome Biol. **21**, 1–28 (2020)
40. Herzog, V.A., Reichholf, B., Neumann, T., Rescheneder, P., Bhat, P., Burkard, T.R., Wlotzka, W., von Haeseler, A., Zuber, J., Ameres, S.L.: Thiol-linked alkylation of RNA to assess expression dynamics. Nat. Methods **14**, 1198–1204 (2017)

41. Hinton, G., Roweis, S.T.: Stochastic neighbor embedding. In: Proceedings of NIPS, vol. 15, pp. 833-840 (2002)
42. Hochgerner, H., Lönnerberg, P., Hodge, R., Mikes, J., Heskol, A., Hubschle, H., Lin, P., Picelli, S., La Manno, G., Ratz, M., Dunne, J., Husain, S., Lein, E., Srinivasan, M., Zeisel, A., Linnarsson, S.: STRT-seq-2i: dual-index 5 single cell and nucleus RNA-seq on an addressable microwell array. Sci. Rep. **7**, 1–8 (2017)
43. Hua, J., Liu, H., Zhang, B., Jin, S.: LAK: Lasso and K-means based single-cell RNA-seq data clustering analysis. IEEE Access **8**, 129679–129688 (2021)
44. Huppertz, I., Attig, J., D'Ambrogio, A., Easton, L.E., Sibley, C.R., Sugimoto, Y., Tajnik, M., König, J., Ule, J.: iCLIP: protein-RNA interactions at nucleotide resolution. Methods **65**, 274–287 (2014)
45. Igashov, I., Olechnovič, K., Kadukova, M., Venclovas, Č, Grudinin, S.: VoroCNN: deep convolutional neural network built on 3D Voronoi tessellation of protein structures. Bioinformatics **37**, 2332–2339 (2021)
46. Jensen, K.K., Andreatta, M., Marcatili, P., Buus, S., Greenbaum, J.A., Yan, Z., Sette, A., Nielsen, M.: Improved methods for predicting peptide binding affinity to MHC class II molecules. Immunology **154**, 394–406 (2018)
47. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P., Hassabis, D.: Highly accurate protein structure prediction with AlphaFold. Nature **596**(7873), 583–589 (2021)
48. Kertesz, M., Wan, Y., Mazor, E., Rinn, J.L., Nutter, R.C., Chang, H.Y., Segal, E.: Genome-wide measurement of RNA secondary structure in yeast. Nature **467**(7311), 103–107 (2010)
49. Kim, S., Kim, H.S., Kim, E., Lee, M.G., Shin, E.C., Paik, S.: Neopepsee: accurate genome-level prediction of neoantigens by harnessing sequence and amino acid immunogenicity information. Ann. Oncol. **29**, 1030–1036 (2018)
50. Ko, T.W., Finkler, J.A., Goedecker, S., Behler, J.: A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. Nat. Commun. **12**, 1–11 (2021)
51. Kohonen, T.: The self-organizing map. Proc. IEEE **78**, 1464–1480 (1990)
52. Kokiopoulou, E., Chen, J., Saad, Y.: Trace optimization and eigenproblems in dimension reduction methods. Numer. Linear Algebr. Appl. **18**, 565–602 (2011)
53. Kravvaritis, D.C., Yannacopoulos, A.N.: Variational Methods in Nonlinear Analysis. De Gruyter (2020)
54. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**, 79–86 (1951)
55. Künsch, H.: Gaussian Markov random fields. J. Fac. Sci. Univ. Tokyo Sect. IA Math **26**, 53–73 (1979)
56. La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., Lidschreiber, K., Kastriti, M.E., Lönnerberg, P., Furlan, A., Fan, J., Borm, L.E., Liu, Z., van Bruggen, D., Guo, J., He, X., Barker, R., Sundström, E., Castelo-Branco, G., Cramer, P., Adameyko, I., Linnarsson, S., Kharchenko, P.V.: RNA velocity of single cells. Nature **560**(7719), 494–498 (2018)
57. Lagopati, N., Belogiannis, K., Angelopoulou, A., Papaspyropoulos, A., Gorgoulis, V.G.: Non-Canonical functions of the ARF tumor suppressor in development and tumorigenesis. Biomolecules **11**, 86 (2021)
58. Lam, J.H., Li, Y., Zhu, L., Umarov, R., Jiang, H., Héliou, A., Sheong, F.K., Liu, T., Long, Y., Li, Y., Fang, L., Altman, R.B., Chen, W., Huang, X., Gao, X.: A deep learning framework to predict binding preference of RNA constituents on protein surface. Nat. Commun. **10**, 1–13 (2019)
59. Lee, J.M.: Introduction to Riemannian Manifolds. Springer International Publishing (2018)
60. Lek, M., Karczewski, K.J., Minikel, E.V., Samocha, K.E., Banks, E., Fennell, T., MacArthur, D.G.: Analysis of protein-coding genetic variation in 60,706 humans. Nature **536**(7616), 285–291 (2016)

61. Leondes, C.T.: The Maximum A Posteriori (MAP) rule. Computer Techniques and Algorithms in Digital Signal Processing: Advances in Theory and Applications, vol. 75. Academic Press, Elsevier, USA (1996)
62. Li, H., Brouwer, C. R., and Luo, W.: A universal deep neural network for in-depth cleaning of single-cell RNA-seq data. BioRxiv (2020)
63. Li, J., Chen, S.J.: RNA 3D structure prediction using coarse-grained models. Front. Mol. Biosci. **8** (2021)
64. Li, Y., Wang, G., Tan, X., Ouyang, J., Zhang, M., Song, X., Liu, Q., Leng, Q., Chen, L., Xie, L.: ProGeo-neo: a customized proteogenomic workflow for neoantigen prediction and selection. BMC Med. Genomics **13**, 1–11 (2020)
65. Litviňuková, M., Talavera-López, C., Maatz, H., Reichart, D., Worth, C.L., Lindberg, E.L., Kanda, M., Polanski, K., Heinig, M., Lee, M., Nadelmann, E.R., Roberts, K., Tuck, L., Fasouli, E.S., DeLaughter, D.M., McDonough, B., Wakimoto, H., Gorham, J.M., Samari, S., Mahbubani, K.T., Saeb-Parsy, K., Patone, G., Boyle, J.J., Zhang, H., Zhang, H., Viveiros, A., Oudit, G.Y., Bayraktar, O.A., Seidman, J.G., Seidman, C.E., Noseda, M., Hubner, N., Teichmann, S.A.: Cells of the adult human heart. Nature **588**(7838), 466–472 (2020)
66. Liu, W., Meng, X., Xu, Q., Flower, D.R., Li, T.: Quantitative prediction of mouse class I MHC peptide binding affinity using support vector machine regression (SVR) models. BMC Bioinform. **7**, 182 (2006)
67. Liu, Y., Gong, W., Zhao, Y., Deng, X., Zhang, S., Li, C.: aPRBind: protein-RNA interface prediction by combining sequence and I-TASSER model-based structural features learned with convolutional neural networks. Bioinformatics **37**, 937–942 (2021)
68. Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., Moore, H.F.: The genotype-tissue expression (GTEx) project. Nat. Genet. **45**, 580–585 (2013)
69. Loughrey, D., Watters, K.E., Settle, A.H., Lucks, J.B.: SHAPE-Seq 2.0: systematic optimization and extension of high-throughput chemical probing of RNA secondary structure with next generation sequencing. Nucl. Acids Res. **42**, e165–e165 (2014)
70. Lu, Z., Zhang, Q.C., Lee, B., Flynn, R.A., Smith, M.A., Robinson, J.T., Davidovich, C., Gooding, A.R., Goodrich, K.J., Mattick, J.S., Mesirov, J.P., Cech, T.R., Chang, H.Y.: RNA duplex map in living cells reveals higher-order transcriptome structure. Cell **165**, 1267–1279 (2016)
71. Luecken, M.D., Theis, F.J.: Current best practices in single-cell RNA-seq analysis: a tutorial. Mol. Syst. Biol. **15**, e8746 (2019)
72. Macosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A.R., Kamitaki, N., Martersteck, E.M., Trombetta, J.J., Weitz, D.A., Sanes, J.R., Shalek, A.K., Regev, A., McCarroll, S.A.: Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell **161**, 1202–1214 (2015)
73. Martınez, R.: PaccMannRL: designing anticancer drugs from transcriptomic data via reinforcement learning (2019). arXiv preprint. arXiv:1909.05114
74. Meng, G., Tariq, M., Jain, S., Elmetwaly, S., Schlick, T.: RAG-Web: RNA structure prediction/design using RNA-As-graphs. Bioinformatics **36**, 647–648 (2020)
75. Moguerza, J.M., Muñoz, A.: Support vector machines with applications. Stat. Sci. **21**, 322–336 (2006)
76. Monga, V., Li, Y., Eldar, Y.C.: Algorithm unrolling: interpretable, efficient deep learning for signal and image processing. IEEE Signal Process. Mag. **38**, 18–44 (2021)
77. Pan, X., Shen, H.B.: RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. BMC Bioinform. **18**, 1–14 (2017)
78. Papayiannis, G.I., Domazakis, G.N., Drivaliaris, D., Koukoulas, S., Tsekrekos, A.E., Yannacopoulos, A.N.: On clustering uncertain and structured data with Wasserstein barycenters and a geodesic criterion for the number of clusters. J. Stat. Comput. Simul. 1–26 (2021)
79. Papaspyropoulos, A., Lagopati, N., Mourkioti, I., Angelopoulou, A., Kyriazis, S., Liontos, M., Gorgoulis, V.G., Kotsinas, A.: Regulatory and functional involvement of long non-coding RNAs in DNA double-strand break repair mechanisms. Cells **10**, 1506 (2021)

80. Pena, J.M., Lozano, J.A., Larranaga, P.: An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognit. Lett. **20**, 1027–1040 (1999)
81. Peterson, V.M., Zhang, K.X., Kumar, N., Wong, J., Li, L., Wilson, D.C., Moore, R., McClanahan, T.K., Sadekova, S., Klappenbach, J.A.: Multiplexed quantification of proteins and transcripts in single cells. Nat. Biotechnol. **35**, 936–939 (2017)
82. Pezoulas, V.C., Hazapis, O., Lagopati, N., Exarchos, T.P., Goules, A.V., Tzioufas, A.G., Fotiadis, D.I., Stratis, I.G., Yannacopoulos, A.N., Gorgoulis, V.G.: Machine learning approaches on high throughput NGS data to unveil mechanisms of function in biology and disease. Cancer Genomics Proteomics **18**, 605–626 (2021)
83. Picelli, S., Faridani, O.R., Björklund, Å.K., Winberg, G., Sagasser, S., Sandberg, R.: Full-length RNA-seq from single cells using Smart-seq2. Nat. Protoc. **9**, 171–181 (2014)
84. Pierson, E., Yau, C.: ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol. **16**, 1–10 (2015)
85. Qiu, Q., Hu, P., Qiu, X., Govek, K.W., Cámara, P.G., Wu, H.: Massively parallel and time-resolved RNA sequencing in single cells with scNT-seq. Nat. Methods **17**, 991–1001 (2020)
86. Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H.A., Trapnell, C.: Reversed graph embedding resolves complex single-cell trajectories. Nat. Methods **14**, 979–982 (2017)
87. Ramakers, J., Blum, C.F., König, S., Harmeling, S., Kollmann, M.: De Novo prediction of RNA 3D structures with deep learning. BioRxiv (2021)
88. Rao, J., Zhou, X., Lu, Y., Zhao, H., Yang, Y.: Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. Iscience **24**, 102393 (2021)
89. Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., Song, Y.S.: Evaluating protein transfer learning with TAPE. Adv. Neural Inf. Process. Syst. **32**, 9689 (2019)
90. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. 400–407 (1951)
91. Rodriguez, S., Hug, C., Todorov, P., Moret, N., Boswell, S.A., Evans, K., Zhou, G., Johnson, N.T., Hyman, B.T., Sorger, P.K., Albers, M.W., Sokolov, A.: Machine learning identifies candidates for drug repurposing in Alzheimer's disease. Nat. Commun. **12**, 1–13 (2021)
92. Rosenberg, A.B., Roco, C.M., Muscat, R.A., Kuchina, A., Sample, P., Yao, Z., Gray, L., Peeler, D.J., Mukherjee, S., Chen, W., Pun, S.H., Sellers, D.L., Tasic, B., Seelig, G.: SPLiT-seq reveals cell types and lineages in the developing brain and spinal cord. Science (New York, NY) **360**(6385), 176 (2018)
93. Rouskin, S., Zubradt, M., Washietl, S., Kellis, M., Weissman, J.S.: Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo. Nature **505**(7485), 701–705 (2014)
94. Sakellaropoulos, T., Vougas, K., Narang, S., Koinis, F., Kotsinas, A., Polyzos, A., Moss, T.J., Piha-Paul, S., Zhou, H., Kardala, E., Damianidou, E., Alexopoulos, L.G., Aifantis, I., Townsend, P.A., Panayiotidis, M.I., Sfikakis, P., Bartek, J., Fitzgerald, R.C., Thanos, D., Mills Shaw, K.R., Petty, R., Tsirigos, A., Gorgoulis, V.G.: A deep learning framework for predicting response to therapy in cancer. Cell Rep. **29**(11), 3367–3373 (2019)
95. Sato, K., Akiyama, M., Sakakibara, Y.: RNA secondary structure prediction using deep learning with thermodynamic integration. Nat. Commun. **12**, 1–9 (2021)
96. Schenck, R.O., Lakatos, E., Gatenbee, C., Graham, T.A., Anderson, A.R.: NeoPredPipe: high-throughput neoantigen prediction and recognition potential pipeline. BMC Bioinform. **20**, 1–6 (2019)
97. Sharma, E., Sterne-Weiler, T., O'Hanlon, D., Blencowe, B.J.: Global mapping of human RNA-RNA interactions. Mol. Cell **62**, 618–626 (2016)
98. Sheng, K., Cao, W., Niu, Y., Deng, Q., Zong, C.: Effective detection of variation in single-cell transcriptomes using MATQ-seq. Nat. Methods **14**, 267–270 (2017)
99. Smibert, P., Mimitou, E., Stoeckius, M.: ECCITE-seq (2019). https://protocolexchange.researchsquare.com/article/nprot-7393/v1. Last Accessed 24 Sept 2021
100. Specht, H., Emmott, E., Petelski, A.A., Huffman, R.G., Perlman, D.H., Serra, M., Kharchenko, P., Koller, A., Slavov, N.: Single-cell proteomic and transcriptomic analysis of macrophage heterogeneity using SCoPE2. Genome Biol. **22**, 1–27 (2021)

101. Spyropoulou, Z., Papaspyropoulos, A., Lagopati, N., Myrianthopoulos, V., Georgakilas, A.G., Fousteri, M., Kotsinas, A., Gorgoulis, V.G.: Cockayne syndrome group B (CSB): the regulatory framework governing the multifunctional protein and its plausible role in cancer. Cells **10**, 866 (2021)
102. Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P.K., Swerdlow, H., Satija, R., Smibert, P.: Simultaneous epitope and transcriptome measurement in single cells. Nat. Methods **14**, 865–868 (2017)
103. Stranzl, T., Larsen, M.V., Lundegaard, C., Nielsen, M.: NetCTLpan: pan-specific MHC class I pathway epitope predictions. Immunogenetics **62**, 357–368 (2010)
104. Sun, S., Wang, W., Peng, Z., Yang, J.: RNA inter-nucleotide 3D closeness prediction by deep residual neural networks. Bioinformatics **37**, 1093–1098 (2021)
105. Sun, L., Xu, K., Huang, W., Yang, Y.T., Li, P., Tang, L., Xiong, T., Zhang, Q.C.: Predicting dynamic cellular protein-RNA interactions by deep learning using in vivo RNA structures. Cell Res. **31**, 495–516 (2021)
106. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning, pp. 3319–3328. PMLR (2017)
107. Svensson, F., Norinder, U., Bender, A.: Improving screening efficiency through iterative screening using docking and conformal prediction. J. Chem. Inf. Model. **57**, 439–444 (2017)
108. Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N.J., Livak, K.J., Mikkelsen, T.S., Rinn, J.L.: The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat. Biotechnol. **32**, 381–386 (2014)
109. Vamvakas, I., Lagopati, N., Andreou, M., Sotiropoulos, M., Gatzis, A., Limouris, G., Antypas, C., Lyra, M.: Patient specific computer automated dosimetry calculations during therapy with 111In Octreotide. Eur. J. Radiogr. **1**, 180–183 (2009)
110. Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A.J., Burdziak, C., Moon, K.R., Chaffer, C.L., Pattabiraman, D., Bierie, B., Mazutis, L., Wolf, G., Krishnaswamy, S., Pe'er, D.: Recovering gene interactions from single-cell data using data diffusion. Cell **174**, 716–729 (2018)
111. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. 9 (2008)
112. Vergetis, V., Skaltsas, D., Gorgoulis, V.G., Tsirigos, A.: Assessing drug development risk using big data and machine learning. Cancer Res. **81**, 816–819 (2021)
113. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
114. Vougas, K., Sakellaropoulos, T., Kotsinas, A., Foukas, G.P., Ntargaras, A., Koinis, F., Polyzos, A., Myrianthopoulos, V., Zhou, H., Narang, S., Georgoulias, V., Alexopoulos, L., Aifantis, I., Townsend, P.A., Sfikakis, P., Fitzgerald, R., Thanos, D., Bartek, J., Petty, R., Tsirigos, A., Gorgoulis, V.G.: Machine learning and data mining frameworks for predicting drug response in cancer: an overview and a novel in silico screening process based on association rule mining. Pharmacol. Ther. **203**, 107395 (2019)
115. Wan, J., Liu, W., Xu, Q., Ren, Y., Flower, D.R., Li, T.: SVRMHC prediction server for MHC-binding peptides. BMC Bioinform. **7**, 1–5 (2006)
116. Wang, K., Li, M., Hakonarson, H.: ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucl. Acids Res. **38**, e164–e164 (2010)
117. Wang, L. (ed.).: Support Vector Machines: Theory and Applications, vol. 177. Springer Science & Business Media (2005)
118. Webb, G.I., Keogh, E., Miikkulainen, R.: Naïve Bayes. Encycl. Mach. Learn. **15**, 713–714 (2010)
119. Yang, J., Zhang, Y.: Protein structure and function prediction using I-TASSER. Curr. Protoc. Bioinform. **52**, 5–8 (2015)
120. Yu, W., Lee, H.K., Hariharan, S., Bu, W., Ahmed, S.: Evolving generalized Voronoi diagrams for accurate cellular image segmentation. Cytom. A **77**, 379–86 (2010)
121. Zhang, H., Zhang, C., Li, Z., Li, C., Wei, X., Zhang, B., Liu, Y.: A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming. Front. Genet. **10**, 467 (2019)

122. Zhang, S., Zhou, J., Hu, H., Gong, H., Chen, L., Cheng, C., Zeng, J.: A deep learning framework for modeling structural features of RNA-binding protein targets. Nucl. Acids Res. **44**, e32–e32 (2016)
123. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **67**, 301–320 (2005)