Noname manuscript No.
(will be inserted by the editor)

A Modified NSGA-II with Silhouette Coefficient and K-means Clustering

Nadir Mahammed · Abdelghani Bekka · Yassine Kazi Tani · Souad Bennabi · Mahmoud Fahci · Badia Klouche · Zouaoui Guellil

Received: date / Accepted: date

Abstract This article is a proposition for enhancing the genetic algorithm NSGA-II by some form of hybridization. The later explores the K-means clustering algorithm and the Silhouette coefficient features. It implies two specific phases. First, the right number of clusters generated automatically by K-means clustering is verified by Silhouette coefficient according to a number of iterations. Thereafter, NSGA-II is executed, in turn, for a defined number of iterations within the proposed algorithm. Obtained results of the algorithm for some benchmark test functions are used to illustrate the validity of the article proposition.

Keywords evolutionary algorithm · NSGA-II · hybridization · K-means clustering · silhouette coefficient

Nadir Mahammed
PO 73, Post office El Wiam, Sidi Bel Abbés 22016, Algeria
Tel.: +213-48-74-94-52
E-mail: n.mahammed@esi-sba.dz

Abdelghani Bekka
E-mail: a.Bekka@univ-chlef.dz

Yassine Kazi Tani
E-mail: y.kazitani@esi-sba.dz

Souad Bennabi
E-mail: E-mail: s.bennabi@univ-chlef.dz

Mahmoud Fahci
E-mail: m.fahci@univ-sba.dz

Badia Klouche
E-mail: b.klouche@esi-sba.dz

Zouaoui Guellil
E-mail: z.guellil@esi-sba.dz

1 Introduction

Everyday, stakeholders and company managers are confronted with problems of increasing complexity in various technical fields. The problem to be solved can often be expressed as an optimization problem. To solve these problems as well as possible, a group of methods, called evolutionary algorithms (EA), has been available from the 80s. EA are a subset of evolutionary computation which implies generally techniques implementing mechanisms inspired by biological evolution. The genetic algorithms (GA) are one of the most well-known algorithms.

GA are population-based oriented which leads to the obvious deduction that the quality and quantity of the launch population (i.e. initial population) are significant. To confirm this proposal, our interest is focused in a specific genetic algorithm, the NSGA-II. In this paper, we present a revised version of NSGA-II on which we applied Silhouette coefficient combined with K-means clustering method to influence and manipulate the population of individuals during NSGA-II iterations. The experiments show a promising results regarding to solutions number or time execution.

The paper is organized as follows. We start by a literate review of the different work implying NSGA-II and K-means in an optimization matter, in Section 2. Thereafter, we briefly mention the concepts used in the present research in Section 3. In Section 4, we describe the proposed SK-NSGA-II algorithm in details. Section 5 presents results of the proposed algorithm and compare them with the canonical NSGA-II using different test functions to evaluate characteristics of our algorithm. Finally, we outline the conclusions of this paper.

2 Related work

The authors of [9] proposed a novel technique for cooling of an array of vertical printed circuit boards by providing fins on the backside of the PCBs, forming a high density fin region. NSGA-II was employed to obtain the global Pareto optimal set containing non-dominated solutions. Thereafter, K-means was used to In order to obtain heat transfer and friction loss oriented designs. To perform interval forecasting of the future wind speed, the authors in [26] used a two steps method to radial basis function neural networks. First, K-means is applied to determine the centers of RDF. Second, NSGA-II is used to adjust the hidden output weights.

The authors of [22] proposed a hybrid heuristic algorithm which combines K-means and NSGA-II to minimize operating cost and the total number of vehicles in the network in a collaborative multiple centers vehicle routing problem with simultaneous delivery and pickup. no details were provided on hybridization. In [24], the authors dealt with the two-echelon collaborative multiple centers vehicle routing problem (collaboration mechanism and the vehicle routing). They combined K-means with an improved NSGA-II to reduce car-

bon dioxide emission and minimize operation cost. In this paper, a method based on the K-means clustering algorithm is utilized to detect and delete outliers. Then, in order to select significant and effective features, NSGA-II and other meta-heuristic algorithms are employed to choose the least number of significant features with the highest classification accuracy using support vector machines. The authors [1] were looking to find the main causes of diabetes mellitus, based on high volume of data involved in therapeutic contexts and disease diagnosis.

The authors in [10] proposed to model a single allocation multi-commodity hub and-spoke network problem, considering the congestion in both hubs and connection links. The model used a novel learning based metaheuristic based on NSGA-II, K-Means clustering method, and an iterated local search algorithm. The paper [14] studied a green meal delivery routing problem (meal delivery and vehicle routing problem). The authors proposed aa multi-objective scheduling model to maximize customer satisfaction and rider balance utilization, and minimize carbon footprint. Then, NSGA-II is adopted to find an initial rider number at the first stage. Principal component analysis and K-means are used to merge the customers' orders and generate the initial delivery routing. The paper of [18] presented an integrated framework for management of aquifers threatened by saltwater intrusion. K-means is used to decrease the number of training parameters; the aquifer area is divided into different zones. Additionally, it is then integrated with NSGA-II with the aim of maximize pumping rates and minimize saltwater intrusion length. In [25], the authors established a bi-objective programming model that optimize the total operating cost and the total number of delivery vehicles. The authors look to solve a collaborative multi-depot vehicle routing problem with time window assignment. They used a hybrid heuristic algorithm consisting of K-means, clarke–wright saving algorithm and an extended NSGA-II.

The authors proposed a work [19] to solve the energy demand planning in Smart Homes, in the form of a methodology that combines three (03) artificial intelligence techniques: elitist NSGA-II, the support vector regression technique and K-means to determine the user comfort levels. In [23], this study tackled a collaborative multi-center vehicle routing problem with resource sharing and temperature control constraints, by minimizing the total cost and the number of refrigerated vehicles. To find the solution, the authors developed a hybrid heuristic algorithm that combines the extended K-means and tabu search NSGA-II to search a large solution space. The authors of [21] considered the stability of collaboration (between companies) by comparing different profit allocation strategies in the collaborative logistics pickup and delivery problem with eco-packages solving process. So, they proposed a methodology that combines multi-objective mixed integer programming, multidimensional K-means, reference point based NSGA-II, forward dynamic programming and improved shapley value method.

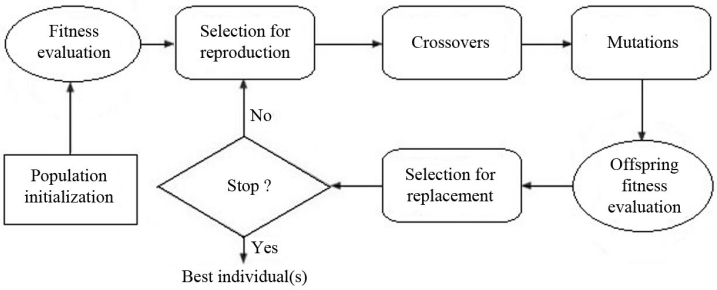


Fig. 1 Principle of an evolutionary algorithm.

3 Basic Concepts

3.1 Evolutionary algorithms

Evolutionary algorithms are search techniques inspired by the biological evolution of species, from the end of the 1950s [6]. The principle of a simple EA can be summarized as follows (Fig. 1):

1. An initial population is constituted of a set of N points in a search space, selected randomly.
2. Each individual x of the population has a given performance, which able to measure its degree of adaptation to the aimed objective.
3. It consists in evolving progressively, in successive generations, the composition of the population (with a constant size).
4. During generations, the overall performance of the individuals are improved.
5. Such a result can be obtain by imitating the two principal mechanisms which govern the evolution of living beings by Darwin's theory:
   (a) Selection: In which is favored the reproduction and survival of the fittest individuals..
   (b) Reproduction: Where recombination (e.g. crossover) and variation (e.g. mutation) of the hereditary features of the parents, to form descendants with new capabilities are allowed.

3.2 NSGA-II

Among several approaches in evolutionary algorithms [8, 13, 16], genetic algorithms (GA) constitute surely the most well-known example [8]. Non-dominated Sorting Genetic Algorithm or NSGA-II [3] is one of the most popular GA and multi objective optimization algorithm with interesting features:

– A fast non-dominated sorting approach.
– A fast crowded distance estimation procedure.

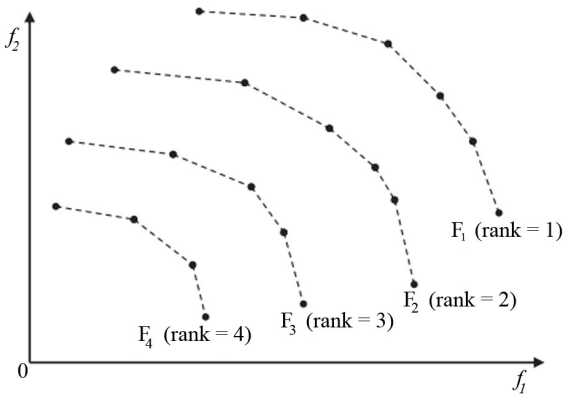


Fig. 2 Rank assignment by non-dominated sorting.

Generally, NSGA-II can be summarized as following steps:

1. Population initialization : Initialize the population based on the problem definition and its constraints.
2. Non dominated sort : A sorting process based on non-domination criteria of the initialized population (Fig. 2).
3. Crowding distance : After the sorting process, the crowding distance value is assign to each individual. Afterward, the individuals are selected based on both rank and crowding distance.
4. Genetic Operators : A binary tournament selection, a simulated binary crossover and polynomial mutation are applied.
5. Recombination and selection : The individuals of the next generation are determined through selection, once the offspring population and current population are combined.

3.3 K-means clustering

K-means clustering algorithm [5] [15] is seen as a Swiss knife for metric data. It force field lies in the simplicity of use, and the local-minimum convergence properties.

K-means or naive K-means algorithm (Fig. 3) is used to produce a clustering of the points in the input into k clusters . The algorithm segments or partitions the data-points into k subsets in a way that all points in a given subset pertain to a center. The algorithm keeps track of the centroids of the subsets, and operates in iterations. The centroids are noted $C^{(i)}$ after the i^{th} iteration i.e. the centroids locations stay fixed. First, The centroids are initialized randomly. The algorithm stops when $C^{(i)}$ and $C^{(i-1)}$ are identical. In each iteration, the following is performed:

1. For each point x, find the center in $C^{(i)}$ which is closest to x. Associate x with this center.

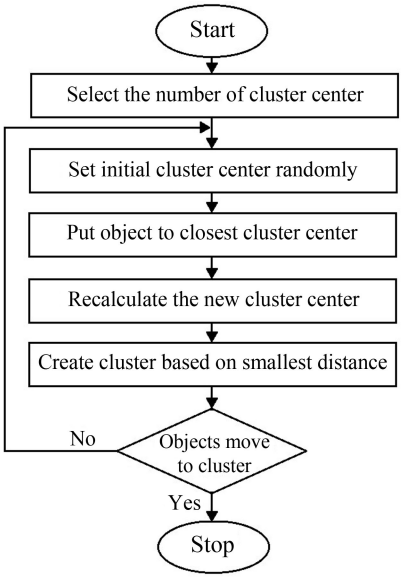


Fig. 3 K-means clustering flowchart.

2. Recalculate $C^{(i+1)}$ by taking, for each center, the center of mass of points associated with this center.

3.4 Silhouette coefficient

The silhouette coefficient or silhouette coefficient is a cluster validity measure [11] [2]. It is a metric used to calculate the goodness of a clustering technique (Fig. 4).

- 1: Clusters are well apart from each other and clearly distinguished.
- 0: Clusters are indifferent or the distance between clusters is not significant.
- -1: Clusters are assigned in the wrong way.

More formally:

$$s = (b - a)/max(b, a) \tag{1}$$

- $s \in [-1, 1]$.
- a: Average intra-cluster distance i.e. the average distance between each point within a cluster.
- b: Average inter-cluster distance i.e. the average distance between all clusters.

To summarize, s measures how well an object matches the clustering at hand i.e. how well it has been classified.

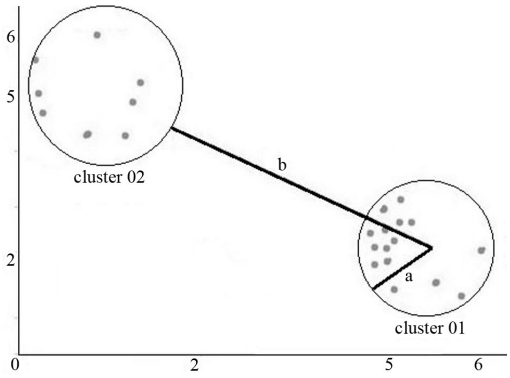


Fig. 4 Silhouette coefficient principal.

4 Our proposition

NSGA-II is a popular multi objective optimization algorithm. Compared with the first generation of NSGA, it has the following three (03) improvements:

1. The concept of fast non-dominated sorting reduces the computational complexity $O(MN^3)$ from to $O(MN^2)$.
2. The elite mechanism pushes the parent and the offspring populations to compete to generate a new population, to ensures that the obtained optimal solutions are kept.
3. The crowding distance operator is used to overcome the shortcoming of the shared Fitness function.

Although the traditional NSGA-II has the above advantages, there are still some interrogations to be set and enhancement that need to be done. Indeed, we aim to address the following questions:

1. How do various initial population within NSGA-II influence its performance ?
2. How does the algorithm deal with a large population size ?
3. Can we develop practical implementation for enhancing these features within NSGA-II ?

By addressing the above questions, this work contributes to gain more knowledge of NSGA-II operation. So, The modified NSGA-II, named Silhouette for K-means NSGA-II (SK-NSGA-II) (see Fig. 5). It is designed as an hybrid or combination of the Silhouette coefficient applied on K-means clustering algorithm during traditional NSGA-II execution (Fig. 5).

K-means has been added to NSGA-II to promote the use of multiple subpopulations instead of a unique population [16]. But when clustering a dataset, the right number k of clusters to use is often not obvious, and choosing k automatically is a hard algorithmic problem. To over come this matter, Silhouette coefficient is used to find optimal clusters in K-means. Silhouette coefficient

has the particularity to relate to a standard concept of clustering when the its score is higher (near +1) which leads to clusters well separated and dense.

The procedure of SK-NSGA-II can be summed up as followed:

1. P_t : parent population, Q_t : offspring population,
   $R_t = P_t \cup Q_t$, t : generation index.
2. Generate k clusters $\{C_1, C_2, \ldots, C_k\}$ from R_t.
3. Calculate S_k : silhouette coefficient for each cluster.
4. Compare S_k.
5. Repeat (2 to 4).
6. Keep k with highest S_k.
7. Apply non-dominated sorting on each cluster $C`_k$ to obtain n fronts F.
8. Sort each front F_i based on crowding distance.
9. $F = \{F_1, F_2, \ldots, F_n\}$ are in descending order.
10. Generate new population $P_t + 1$ by mustering each top (best) front from $C`_k$.

To calculate the overall complexity of the proposed algorithm, we consider the complexity of one iteration of the entire algorithm. The basic operations and their worst-case complexities are as follows:

- The k-means clustering is $O(ktn)$.
  - n is the number of objects.
  - k is the number of clusters.
  - t is the number of iterations.
- The Silhouette coefficient is $O(n)$.
- The non-dominated sorting is $O(m(2n)^2)$.
- The crowding-distance sorting is $O(2nlog(2n))$.

The overall complexity of the algorithm is $O(mn)^2$.

5 Experiments and Results

In this section, we first present the test problems used to compare the performance of the proposed algorithm with canonical NSGA-II. For the comparison, we have taken identical parameter settings as suggested in the original study by [3].

So, we start by shortly describe the test problems used in the comparison process of our study. From different studies in the multi objective evolutionary algorithms (MOEA), four (04) test problems are chosen:

1. Schaffer's study (SCH) [20]
2. Kursawe's study (KUR) [13].
3. Poloni's study (POL) [17].
4. Fonseca–Fleming's study (FON) [4].

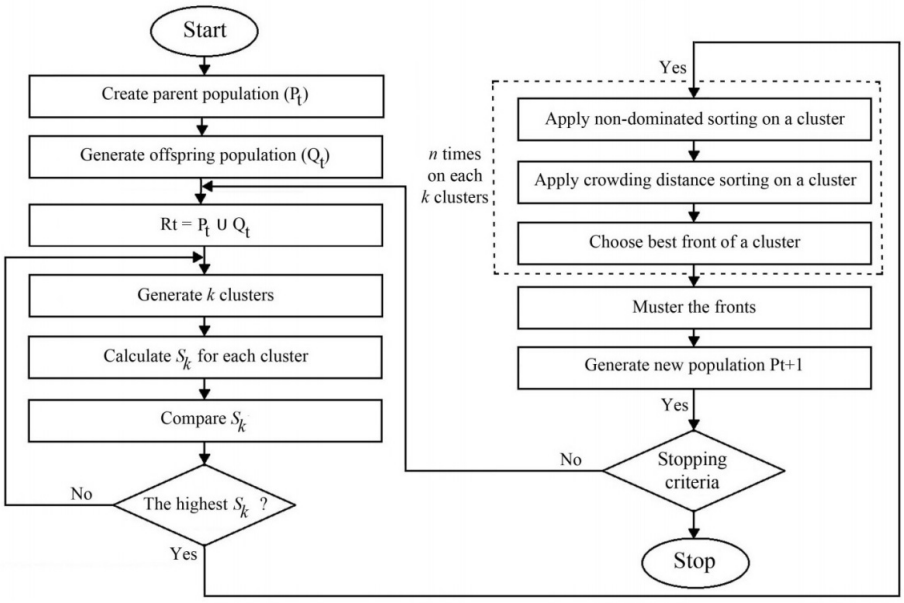


Fig. 5 SK-NSGA-II steps.

During the experimentation and to demonstrate the efficiency of SK-NSGA-II, we compared it results with those obtained by NSGA-II and a proposition of the NSGA-II coupled with the K-means clustering algorithm without Silhouette coefficient, called K-NSGA-II. We chose to implement the algorithms real-coded because real coding is recommended for optimization problems where the parameter space is continuous [7] and it is the best way to tackle the real-world problem [12]. All the algorithms are run with the following parameters:

- A maximum of 500 function evaluations.
- A population size of 100 individuals.
- A number of 40 generation each iteration.
- Each algorithm is run until 250 generations.
- Number of clusters $K = 3$ with K-NSGA-II.

For each tested algorithm, we use the genetic operators as mentioned in [3] for real-coded:

1. Tournament selection.
2. Single-point crossover (0.9).
3. Bitwise mutation ($1/l$: l number of decision variables).
4. Crossover distribution index is 20.
5. Mutation distribution index is 20.

The implementation of SK-NSGA-II can be found at https://cutt.ly/akkNqXs. Table 1 shows the best number of non-dominated solutions obtained using

Table 1 Results regarding non-dominated solutions

Test function	NSGA-II	K-NSGA-II	SK-NSGA-II
SCH	46	73	88
KUR	43	50	67
POL	35	71	80
FON	40	56	69

Table 2 Results regarding execution time (seconds)

Test function	NSGA-II	K-NSGA-II	SK-NSGA-II
SCH	3.72	2.48	1.43
KUR	2.83	1.23	1.19
POL	1.7	1	0.34
FON	4.44	2.81	0.99

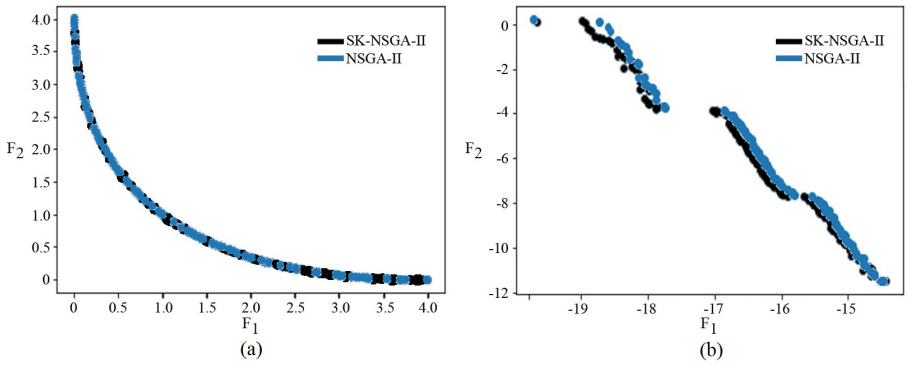


Fig. 6 Non-dominated solutions on (a) SCH, (b) KUR

three (03) algorithms SK-NSGA-II, K-NSGA-II and NSGA-II. SK-NSGA-II performs the best in all chosen test functions. The worst performance is observed with NSGA-II. We note that the number of non-dominated solutions are quite close. K-NSGA-II has done a good job too, but SK-NSGA-II always outperform the obtained values. We take it for granted that that Silhouette coefficient finds the right K number of clusters each iteration.

Table 2 shows the best calculated execution time using the three (03) previous algorithms. SK-NSGA-II performs the best in all chosen test functions. We find that the obtained results support those of Table 1 and the statement that K-means clustering combined with Silhouette coefficient helps NGSA-II to surpass his capacities.

Figures 6 and 7 show all non-dominated solutions obtained after 250 generations with SK-NSGA-II and NSGA-II. The Pareto-optimal region is also shown in each figure. They clearly display that SK-NSGA-II finds a better converged set of non-dominated solutions in any test function compared to the other algorithms.

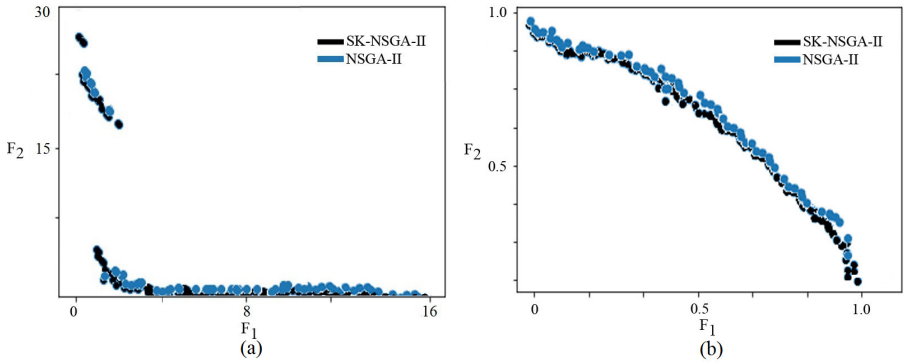


Fig. 7 Non-dominated solutions on (a) POL, (b) FON

The distribution in solutions is better with the proposed algorithm compared to NSGA-II. Even if, as NSGA-II, figures demonstrate the abilities of SK-NSGA-II in converging to the true front and in finding diverse solutions in the front.

6 Conclusion

In the present work, We proposed an attempt to enhance a computational fast genetic algorithm based on clustering method and non-dominated sorting approach. Using different test functions taken from the literature, the proposed algorithm named SK-NSGA-II was able to converge better in the obtained non-dominated front and keeping a better spread of solutions compared to a canonical NSGA-II. The experiments have clearly shown that SK-NSGA-II has been able to come close to the Pareto front faster than the other algorithm. We strongly believe that the proposed algorithm with the clustering mashup property should find more and divers applications in the future. Even though SK-NSGA-II has proven his efficiency, we have to do more experimentation with different constrained test problems and by modifying parameters setting using more explicit performance measures.

References

1. Alirezaei, M., Niaki, S.T.A., Niaki, S.A.A.: A bi-objective hybrid optimization algorithm to reduce noise and data dimension in diabetes diagnosis using support vector machines. Expert Systems with Applications **127**, 47–57 (2019)
2. Berkhin, P.: A survey of clustering data mining techniques. In: Grouping multidimensional data, pp. 25–71. Springer (2006)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
4. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary computation **3**(1), 1–16 (1995)

5. Forgy, E.W.: Cluster analysis of multivariate data: efficiency versus interpretability of classifications. biometrics **21**, 768–769 (1965)
6. Fraser, A.S.: Simulation of genetic systems by automatic digital computers i. introduction. Australian Journal of Biological Sciences **10**(4), 484–491 (1957)
7. Gaffney, J., Pearce, C., Green, D.: Binary versus real coding for genetic algorithms: A false dichotomy? Anziam journal **51**, C347–C359 (2009)
8. Goldenberg, D.E.: Genetic algorithms in search, optimization and machine learning (1989)
9. Jadhav, R.S., Balaji, C.: Fluid flow and heat transfer characteristics of a vertical channel with detached pin-fin arrays arranged in staggered manner on two opposite endwalls. International Journal of Thermal Sciences **105**, 57–74 (2016)
10. Karimi-Mamaghan, M., Mohammadi, M., Pirayesh, A., Karimi-Mamaghan, A.M., Irani, H.: Hub-and-spoke network design under congestion: A learning based metaheuristic. Transportation Research Part E: Logistics and Transportation Review **142**, 102069 (2020)
11. Kaufman, L., Rousseeuw, P.: Finding groups in data; an introduction to cluster analysis. Tech. rep., J. Wiley (1990)
12. Kim, J.W., Kim, S.W., Park, P., Park, T.J.: On the similarities between binary-coded ga and real-coded ga in wide search space. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), vol. 1, pp. 681–686. IEEE (2002)
13. Kursawe, F.: A variant of evolution strategies for vector optimization. In: International Conference on Parallel Problem Solving from Nature, pp. 193–197. Springer (1990)
14. Liao, W., Zhang, L., Wei, Z.: Multi-objective green meal delivery routing problem based on a two-stage solution strategy. Journal of Cleaner Production **258**, 120627 (2020)
15. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
16. Mahammed, N., Bennabi, S., Fahsi, M.: Optimizing business process designs with a multiple population genetic algorithm. In: Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics, pp. 252–254 (2020)
17. Poloni, C.: Hybrid ga for multi objective aerodynamic shape optimisation (1995)
18. Ranjbar, A., Mahjouri, N., Cherubini, C.: Development of an efficient conjunctive meta-model-based decision-making framework for saltwater intrusion management in coastal aquifers. Journal of Hydro-environment Research **29**, 45–58 (2020)
19. Rocha, H.R., Honorato, I.H., Fiorotti, R., Celeste, W.C., Silvestre, L.J., Silva, J.A.: An artificial intelligence based scheduling algorithm for demand-side energy management in smart homes. Applied Energy **282**, 116145 (2021)
20. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the first international conference on genetic algorithms and their applications, 1985. Lawrence Erlbaum Associates. Inc., Publishers (1985)
21. Wang, Y., Peng, S., Guan, X., Fan, J., Wang, Z., Liu, Y., Wang, H.: Collaborative logistics pickup and delivery problem with eco-packages based on time-space network. Expert Systems with Applications p. 114561 (2021)
22. Wang, Y., Zhang, J., Assogba, K., Liu, Y., Xu, M., Wang, Y.: Collaboration and transportation resource sharing in multiple centers vehicle routing optimization with delivery and pickup. Knowledge-Based Systems **160**, 296–310 (2018)
23. Wang, Y., Zhang, J., Guan, X., Xu, M., Wang, Z., Wang, H.: Collaborative multiple centers fresh logistics distribution network optimization with resource sharing and temperature control constraints. Expert Systems with Applications **165**, 113838 (2021)
24. Wang, Y., Zhang, S., Assogba, K., Fan, J., Xu, M., Wang, Y.: Economic and environmental evaluations in the two-echelon collaborative multiple centers vehicle routing optimization. Journal of Cleaner Production **197**, 443–461 (2018)
25. Wang, Y., Zhang, S., Guan, X., Peng, S., Wang, H., Liu, Y., Xu, M.: Collaborative multi-depot logistics network design with time window assignment. Expert Systems with Applications **140**, 112910 (2020)
26. Zhang, C., Wei, H., Xie, L., Shen, Y., Zhang, K.: Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework. Neurocomputing **205**, 53–63 (2016)

Noname manuscript No.
(will be inserted by the editor)

Prediction of Cancer Clinical Endpoints Using Deep learning and RPPA data

Imene Zenbout · Abdelkrim Bouramoul · Souham meshoul

Received: date / Accepted: date

Abstract Whithin the advances in highthrouput technologies, handling vast and various cancer omic data requires more accurate and felexible models to either achieve a precise clinical decision or to discover new and relevant diagnostic, prognostic and theurapeutic genes. Reverse protein phase array (RPPA) data are considered to be more stable than gene expression data and contain less noisy inputs. The analysis of these type of data may help in accurately classify cancer types or predict more precise clinical and pathological endpoints. In this paper we construct a computational framework that combines deep learning and biological knowledge, to predict clinical cancer related outcomes and extract a set of discriminative features in cancer classification. The framework have been experimented on different cancer data sets and the results shows promising performance of deep learning and RPPA data in classifying predicting cancer pathological stage, progression free interval, and overall survival.

Keywords Cancer classification · Reverse Protein Phase · Deep learning, Protein-Protein interaction network

I. Zenbout
Fundamental Informatics and its Applications Department, misc laboratory, Faculty NTIC, University Abdelhamid Mehri - Constantine 2, Constantine, Algeria
National Center for Biotechnology Research, Constantine, Algeria.
Research Center for Scientific and Technical Information, Algiers, Algeria.
E-mail: imene.zenbout@constantine2.dz

A. Bouramoul
Fundamental Informatics and its Applications Department, misc laboratory, Faculty NTIC, University Abdelhamid Mehri - Constantine 2, Constantine, Algeria.
E-mail: abdelkrim.bouramoul@univ-constantine2.dz

S.Meshoul
Princess Nourah bint Abdulrahman University IT dept.CCIS-RC Riyadh, KSA
E-mail: sbmeshoul@pnu.edu.sa

1 Introduction

The remarkable advances in high throughput technologies also came with a tremendous advances in measuring protein expression using the Reverse Phase Protein Array, or called RPPA. RPPA is a sensitive and powerful functional proteomic approach that captures cancer related molecular mechanism and aid in the development of therapies [4]. It allows the monitoring of hundreds to thousands of samples simultaneously, in order to the quantitative measuring of protein expression[10]. Previous research explored genomic and transcriptomic impact on cancer diagnosis yet these research mainly aid on a research level and on the monitoring of patients yet the clinical impact was weak and limited in comparison with the needs of targeted therapy [7], also the transcriptomic measurement of next generation sequencing came with the curse of small patient sample and huge genomic expression, which complicated the analysis and exploration of these data [2]. In contrast proteomic measuring is more efficient in capturing biological process [7] as well as handling RPPA data is easier because of the low dimensionaliy in comparison with transcriptomic data. The research on the impact of RPPA on cancer was mainly explored by medical community like the work of Mari et al [7], where the authors presented a detailed research on the impact of RPPA on precision oncology. Also Mari et al [6], explained the signal pathway profiling using RPPA and its clinical application. In cancer classification RPPA have been used to classify breast cancer in the paper of Negm O et al [9].Zhang et al [6] used RPPA data set to classify ten most known cacncer type, where the authors selected the most relevant 23 proteins in cancer type classification.
Our work falls in the range of the first papers that explores the impact of RPPA data on targeting diagnostic endpoints in association with biological protein-protein interaction network(PPI), using a deep learning model. Where we used autoencoders for there relevancy in cancer research , mainly in omic data integration, gene expression analysis, and cancer type classification [3],[11],[5], As well as in cancer stage prediction [12]. The used autoencoders were trained by a set of proteins extracted from the PPI and, in order to map the RPPA features space into a reduced representation, that further used in training classifiers in predicting cancer clinical and pathological endpoints. The architecture was experimented on the PanCancer Atlas data on cancer pathological stage, progression free interval(PFI), and Overall Survival(OS).
The rest of the paper is structured as follows: Section 2 explains the architecture and details its steps. The experimental results were conducted and explained in section3. Finally we concluded our paper in 4 with overall overview and perspectives.

2 Proposed architecture

Our predictive model consists of four phases, the first phase is the data collection and preparation then we applied a proteins' filtration, where we select

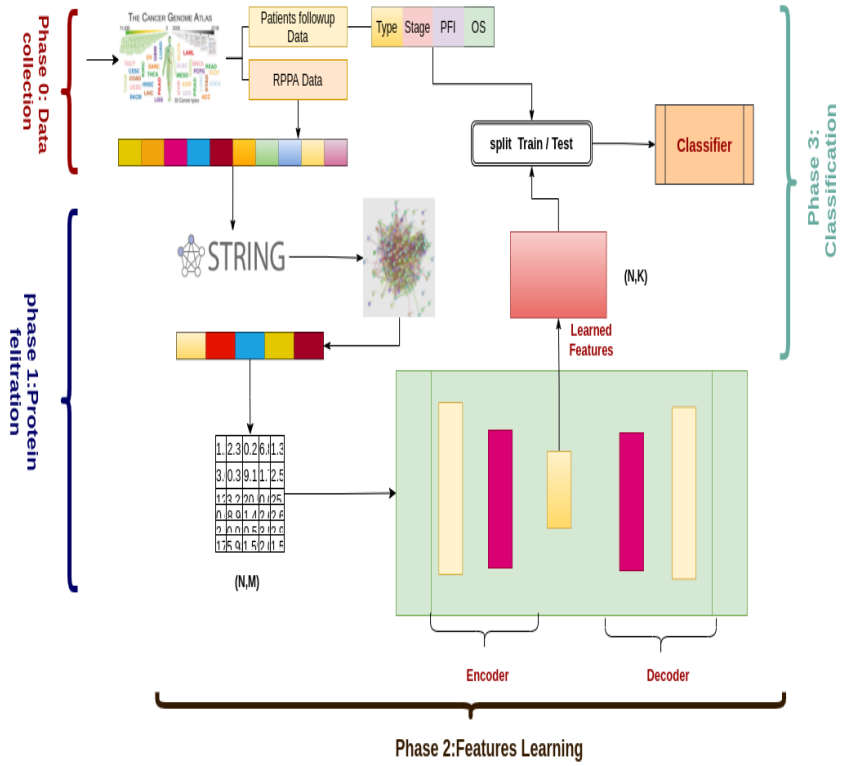


Fig. 1 Illustration of the proposed architecture

only proteins that appears in the PPI network from the string database. The third phase is a feature learning phase, in which we train a deep autoencoder to map the expression of the filtered set of proteins into a reduced new features space, After training the DAE, we pass to the last phase in order to train a classifier based on the learned feature space along with the corresponding endpoint.

2.1 Data Collection

From The Cancer Genome Atlas, we have collected the TCGA-RPPA pancancer data set along with the patients clinical outcomes. The pancancer project represent a set of cancer samples collected from different patients all around the US hospitals for more than 30 cancer type including some rare types[8]. Reverse phase protein array was used to measure the RPPA data set, were the experiments was applied on 7790 patient's sample and, and 199 protein. As for the patients followup data set it contains the clinical, pathological and all the follow up information of 11160 patient.

Table 1 Data Description -1-

	Protein Expression				Follow up data			
Data:	initial Data	Stage	PFI	OS	initial Data	Stage	PFI	OS
Patients:	7790	4954	7769	7677	11160	4954	7769	7677
Proteins:	199				/			

From the followup data set we defined three endpoints as classification targets namely:

- pathological type, which contain four stages and each stage is divided into sub-stages, in our case we adressed all substages as the global stage i.e all sub-stages of stage 1 are considered as stage 1 cancer patient.
- Progression free interval(PFI) and overall survival(OS), where both were addressed as binary endpoints(0/1)

For each endpoint we extracted the set of patients barcode that have an available endpoint value, then , we concatenated the list of patients' barcode with the RPPA matrix in order to construct three expression matrix $S(N\_1 \times M)$, for stage prediction, $P(N\_2 \times M)$ for PFI prediction and $O(N\_3 \times M)$ for OS prediction. Where M is the list of expressed proteins which is initially 199, and N_* is the number of patients with available target value. Table2.1 exhibits the datasets in numbers.

2.2 PPI and features filtration

From the RPPA data sets we collected the set of proteins and constructed the protein-protein network(figure2) using String tools, We downloaded the mapping matrix and the interaction-score matrix. Then, we merged the two matrices by the protein identifier and extracted the list $IM(K)$ of the interacted proteins i.e proteins that posses an interaction with other proteins. This phase reduced the set of proteins from 199 to 97 interacted protein. After defining the list of most interacted proteins we used the matrices S, P, O, to drop the proteins expression that does not appear in IM, which led to re-scale the matrices dimension from (N_*, M) to (N_*, K). After the Construction of the matrices using in training our models, we moved to a preprocessing phase, where we replaced the missing protein expressions by using a KNN imputation and then we normalized the data set using log transformation. After we split the data sets 80% for training and 20% for testing. Table 2.2 shows the data statistics.

2.3 Deep features learning

The tackled features learning problem can be formulated as follows: given a matrix $P(N \times M)$, where vector p_{ij} of P represents the value of an expressed

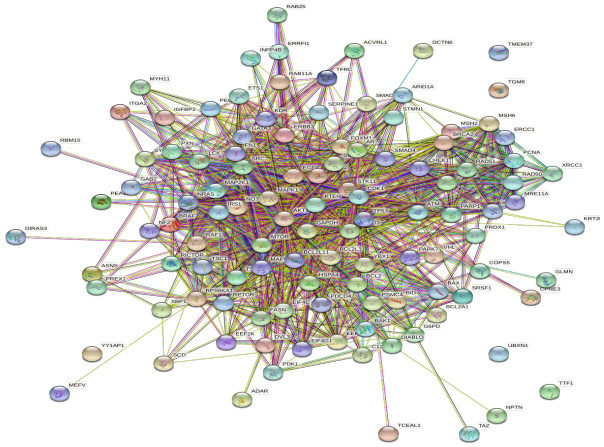


Fig. 2 Initial protein-protein network

Table 2 Data Description -2-

	Stage		PFI		OS	
Train	3963		6143		6141	
Test	991		1536		1536	
endpoints	**Stage 1**	1409	**Class 0**	4994	**Class 0**	5295
	Stage 2	1638	**Class 1**	2685	**Class 1**	2382
	Stage 3	1333				
	Stage 4	574				

protein j for a sample i. Since deep features learning falls in the category of unsupervised learning, at this phase the samples' classes were ignored. Taking in count the matrix P dimensions , the deep autoencoder takes $P(X \times M)$ as an input and transform it successively through the encoder E (eq1) layers into a reduced features representation $P1$ of range K, where $K < M$ at the bottleneck layer.

$$Encoder(P, \Phi_E) = P_1; P \in \mathbf{R}^{N \times M}, P_1 \in \mathbf{R}^{K \times M} \tag{1}$$

Φ_E is the encoder parameters responsible of transforming P into $P1$, the latter represents the linear transformation that contains all the necessary information of P . Following the autoencoder architecture The input P is later reconstructed into P' using the decoder$D$ output (eq:decoder)

$$Decoder(P_1, \Phi_D) = P'; P' \in \mathbf{R}^{N \times M}, P' \approx P. \tag{2}$$

Although, the decoder output is a primordial part in autouencoder, yet in our case it is not the output P' that matters but the learned features $P1$ space by the encoder. So, in order to assist the consistency of the $P1$ and the whole auto encoder performance in mapping P to a reduced features space, we evaluated the performance of $P1$ in reconstructing P into P' through the

reconstruction loss error.In which we calculate the degree of similarity between the input P and the decoder D output P'. In our case we used the $mean\_absolute\_error(mae)$ (eq3)

$$mae = 1/n \sum_{i=1}^{n} |y_i - y'_i| \tag{3}$$

As a result the smaller the loss is the more the AE architecture is capable to generate consistent reduced features space $P1$. So, the objective of training this autoencoder is to minimize the loss (eq5) between P and P':

$$Loss(P, \Phi_D(\Phi_E(P))) = 1/n \sum_{i=1}^{n} |P - P'| \tag{4}$$
$$P' = \Phi_D(\Phi_E(P).$$

2.4 Classification

After training the autoencoder to map the inpt P into an output P' with a low loss score, we used the trained model to map our input data P into the reduced space data P'. Then using cross validation we split the data set into training and testing set (80%,20% respectively). After we built a support vector machine classifier and used the train and test data set to train and evaluate the svm perfomance as well as the performance of the features learned from the previous phases (PPI filtration and dimensionality reduction).

3 Results and discussion

In order to evaluate the performance of the cancer related end points prediction we built three different instances of the proposed architecture *AE+SVM*, one for predicting stage of cancer, one for PFI score and the other for the OS score (figure3). The experiments were conducted on a hp laptop with Intel Core i7-7500U CPU @ 2.70GHz 4, with ubuntu 18.04 operating system. We used the Keras [1] package to implement the autoencoder architucture.

3.1 Implement and Train the features learning model

We built three features learning model each for a specific end point the encoder E is constructed of an input layer(96 nodes), two hidden layers(40,30 nodes) and the bottleneck (20 nodes) layer that represent the output of the encoder D that will be in charge of transforming the data into a reduced features representation $P1$, the decoder D takes as input $P1$ and is built symmetrically to the encoder with two hidden layers and an output layer responsible of reconstructing the input P into P'. We used $softplus$ activation function to setup the layers weights and we trained the models using *adamax* optimizer

Fig. 3 Features learning and classification Architecture

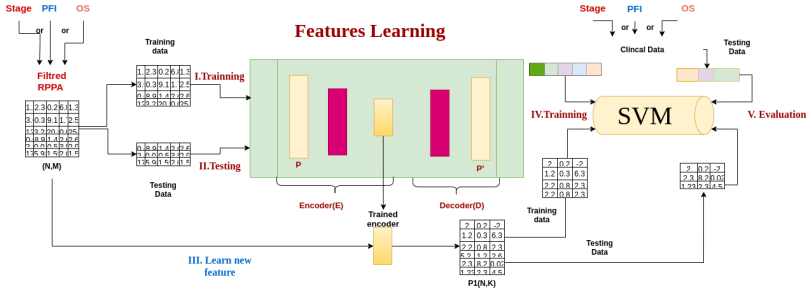


Table 3 Architecture Parameters setting

	Stage	PFI	OS
Architecture	(96,40,30,20,40,96)		
Activation function	Softplus		
Optimizer	adamax		
Batch size	120		
Learning rate	**round 1:** 5^{-3} / **round 2:** 1^{-4}	**round 1:** 5^{-4} / **round 2:** 1^{-4}	**round 1:** 5^{-3} / **round 2:** 1^{-4}
Epochs	**round 1:** 400 / **round 2:** 100		
Loss	**Train:** 0.5864 / **Test:** 0.5992	**Train:** 0.636 / **Test:** 0.6371	**Train:** 0.5580 / **Test:** 0.5741

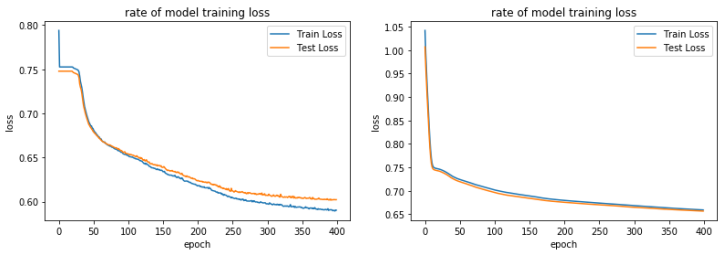
and batch training. After a set of training we set the architecture's parameters as illustrated in table3.1

We trained the three instances in the follow scenarios:

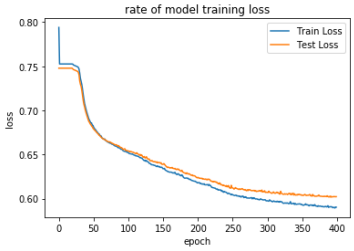
- **Stage:** We trained the autoencoder on two rounds the first round we trained the model for 400 epochs using a batch size of 120 with a learning rate(lr) of 5^{-5}. As shown in figure 4, we notice that the model is training without overfitting. Then we reset the optimizer' lr to 1^{-4} and we trained the model again for 100 epochs, which dropped the loss value to 0.5.
- **PFI/OS:** In the same way we trained the autoencoder in the first round for 400 on batch size equals to 120 with a lr equals to 5^{-4} for PFI and 5^{-3} for OS. The loss training values shows that the model is training without an unnoticeable overfitting. After we reset the optimizer' lr to 1^{-4}, and we scored a loss score of 0.63,0.57 for PFI and OS respectively.

3.2 Classification Results

After training the models for each endpoint we construct the new reduced data set and train/test split the data set in order to train and evaluate an SVM classifier to associate each sample to its corresponding endpoint. To ass the performance of the predictive model and the consistency of the learned features

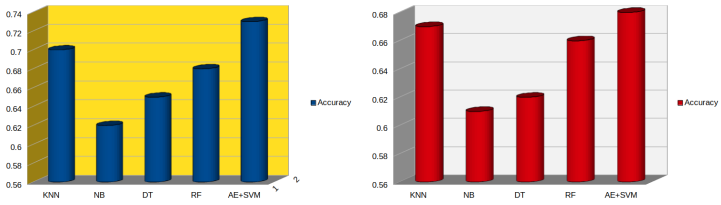


(a) a (b) b

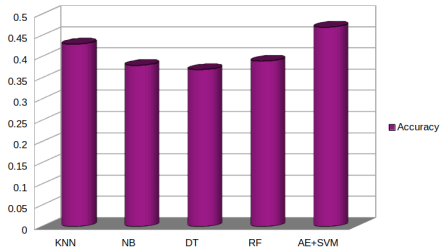


(c) c

Fig. 4 Training performance based on reconstruction loss, (a):Stage,(b):PFI,(c):OS



(a) a (b) b



(c) c

Fig. 5 Classification performance based on Accuracy (a):OS,(b):PFI,(c):Stage

we performed a comparison with some classic classification models namely, K-nearest-neighbors(KNN), Decision Trees(DT), Random Forest (RF), and Gaussian Naive Bayes(NB). The Comparison was established using the following metrics: Accuracy, Precision, Recall,f1_score The histogram plot on figure 5, shows that our proposal outperforms all the other models where we

Table 4 Performance in predicting pathological stage for RPPA Data

	Precision	*Recall*	*f1_score*
KNN	0.42	0.4	0.39
NB	0.38	0.36	0.38
DT	0.35	0.35	0.35
RF	0.21	0.32	0.24
AESVM	**0.5**	**0.41**	**0.39**

Table 5 Performance in predicting PFI score for RPPA Data

	Precision	*Recall*	*f1_score*
KNN	0.62	0.62	**0.62**
NB	0.61	**0.63**	0.6
DT	0.58	0.58	0.58
RF	0.33	0.50	0.40
AESVM	**0.66**	0.54	0.5

was able to score approximately 0.5%,0.68%,0.74% accuracy rate for Stage, PFI,and OS. Whereas, all the other models scored noticeably lower results.
In addiction to accuracy we collected the macro average classification report of the other metrics to capture, where our model's ups and falls. The overall performance of our proposed model in front of the comparison models on predicting pathological stage(Table3.2) was positive along all metrics, citing that KNN had a fair closeness to our results. As for PFI and OS (table3.2,table3.2) our model was able to outperform the other model only in precision yet in PFI NB was able to score the best results with Recall measure and KNN with f1_score. While in OS, KNN and NB scored the best recall and KNN scored the best f1_score .
As general discussion we adress three points, the first one is the performance of data set in training the autoencoder, where we notice the absence of overfitting between training and testing data, with the note we have applied neither regularization, nor penalty dropouts on layers' input. We assume that this phenomena is due to the protein filtration based on the biological knowledge(PPI), that was responsible of eliminating nose input and dropping outliers that may leads to a misleading learning.
The second and third points resumes in the low prediction rate and the weak performance of the proposed model in PFI and OS on Recall and f1_score metrics, where we address this falls to the lack of data for stage prediction, where there is not enough data set for each stage, which leads to poor learning. As well as for the high unbalance between the two classes, which also leads to poor learning and weak discrimination between the samples of each classes, especially for cancers that have high correlation.

4 conclusion

The most crucial phase when dealing with cancer related biological data, whether its is transcriptomic or proteomic is the selection of a representa-

Table 6 Performance in predicting OS score for RPPA Data

	Precision	*Recall*	*f1_score*
KNN	0.65	**0.63**	**0.64**
NB	0.62	**0.63**	0.61
DT	0.6	0.6	0.6
RF	0.34	0.50	0.40
AESVM	**0.72**	0.6	0.6

tive, not noisy feature space. in this paper we tried to curate our RPPA data following two steps. The first was to filter the dataset based on biological background then to extract a small features set using unsupervised deep learning in order to make the classifier learns from data that have a high discriminative ratio and my play the role of in silico molecular signatures. Despite the curse of the unbalanced data sets in terms of endpoits classes, we were able to notice a interesting performances of our proposal that may further helps us on improving those results by data collection or using other biological background such as signaling pathways.

References

1. Chollet, F.: keras. https://github.com/fchollet/keras (2015)
2. Fakoor, R., Ladhak, F., Nazi, A., Huber, M.: Using deep learning to enhance cancer diagnosis and classification. In: Proceedings of the International Conference on Machine Learning, vol. 28. ACM New York, USA (2013)
3. Franco, E.F., Rana, P., Cruz, A., Calderón, V.V., Azevedo, V., Ramos, R.T., Ghosh, P.: Performance comparison of deep learning autoencoders for cancer subtype detection using multi-omics data. Cancers **13**(9), 2013 (2021)
4. Li, J., Akbani, R., Zhao Wand Lu, Y., Weinstein, J., Mills, G., Liang, H.: Explore, visualize, and analyze functional cancer proteomic data using the cancer proteome atlas. Cancer research, **21**(77), 51–54 (2017)
5. Macías-García, L., Luna-Romera, J.M., García-Gutiérrez, J., Martínez-Ballesteros, M., Riquelme-Santos, J.C., González-Cámpora, R.: A study of the suitability of autoencoders for preprocessing data in breast cancer experimentation. Journal of biomedical informatics **72**, 33–44 (2017)
6. Mari, M., Tesshi, Y.: Signaling pathway profiling using reverse-phase protein array and its clinical applications. Expert Review of Proteomics **14**(7), 607– (2017)
7. Masuda, M., Yamada, T.: Utility of Reverse-Phase Protein Array for Refining Precision Oncology, pp. 239–249 (2019)
8. Nawy T, A.: pan-cancer atlas. Nat Methods **15**(407) (2018)
9. Negm, O., al: Clinical utility of reverse phase protein array for molecular classification of breast cancer. Breast cancer research and treatment **155**(1), 25–35 (2016)
10. Spurrier, B., Ramalingam, S., Nishizuka, S.: Reverse-phase protein lysate microarrays for cell signaling analysis (2008)
11. Way, G.P., Greene, C.S.: Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. In: PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium, pp. 80–91. World Scientific (2018)
12. Zenbout, I., Bouramoul, A., Meshoul, S.: Targeted unsupervised features learning for gene expression data analysis to predict cancer stage. In: Proceedings of the Tenth International Conference on Computational Systems-Biology and Bioinformatics, pp. 1–7 (2019)