A Multi-Model Based Approach for Driver Missense Identification

Ahmed T. Soliman, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL, USA
a.soliman@umiami.edu, shyu@miami.edu

***Abstract*—The rapid growth in DNA and protein sequencing techniques over the last decade boosted the availability and scale of mutations data, and therefore the necessity of developing automated approaches to predict driver mutations arises. Identifying driver mutations is essential to better understand and measure cancer progression and thus enable proper diagnosis and targeted treatment of cancer. Here, we present a scalable machine learning based approach to identify driver missense mutations. The proposed approach builds on and expands our previously proposed framework. A group of independent parallel classifiers where each classifier handles a single set of features can be deployed. Then, a model fusion module combines the classifiers' outputs to produce a final mutation label. Each classifier is trained and validated independently with its corresponding feature set. Feature sets undergo a feature selection process to filter out low significance features. In this paper, four protein sequence-level feature sets are leveraged, namely two amino acid indices (AAIndex1 and AAIndex2) feature sets, one pseudo amino acid composition (PseAAC) feature set, and one feature set generated using wavelet analysis. The proposed approach is extensible to consume new additional features with the minimal impact on the computational complexity due to the parallel design of its components. Experiments were performed to assess the performance of the proposed approach and to compare it with other similar approaches.**

Keywords: Cancer genome; driver mutation; passenger mutation

I. INTRODUCTION

Cancer is one of the leading medical causes of mortality in both developing and economically developed countries [1]. Studies show that cancer is the second leading cause of death in the United States, after heart disease, and it accounted for 22.2% and 21.8% of all deaths in 2015 and 2016, respectively [2]. Cancer is mainly caused by the accumulation of somatic mutations, a genetic alternation that is acquired by a cell and is passed to cells resulting from the division of the mutated cells, that occur over a period of time and result in increasing the fitness of some cells over their neighbor cells where they start proliferating and developing cancer [3][4]. There are two classes of genes whose mutations play a role in the progression of cancer, tumor-suppressor genes and oncogenes. Mutations in tumor-suppressor genes deactivate them and compromise their ability to protect cells from cancer and mutations in oncogenes will overactive them and develop tumor. Cancer genome studies revealed that not all somatic mutations in cancer genes play the same role. There are driver mutations that confer a selective growth advantage to the cancer cell and thus drive cancer progression, and passenger mutations that do not provide growth advantage. Passenger mutations were either present in the ancestor of the cancer cell when it acquired a driver mutation or may occur in the cancer cell because of the mutational processes. Although there is a general belief that passenger mutations are neutral, some recent research studies suggested that they may have a damaging effect on the cancer progression [5]. Passenger mutations are more common compared to driver mutations and it is estimated that 90% of somatic mutations are passenger mutations [6], which makes the identification of driver mutations more challenging.

Current research shows a correlation between protein missense mutations, where a single nucleotide change results in a codon that codes for a different amino acid, and cancer. A study covering common solid tumors demonstrates that on average 33 to 66 genes with somatic mutations are expected to change their protein products with approximately 86% of these mutations leading to missense changes [4]. Another research work has demonstrated that missense mutations in the adenomatous polyposis coli (APC) are correlated with the tumorigenesis in the colon as well as extra-intestinal tissues [7].

In this paper, we present a machine learning-based framework to identify driver missense mutation in protein sequences. The proposed framework extends our previously proposed framework in [8] to support additional feature sets and to improve the feature selections process. Fig. 1 illustrates the proposed framework which consists of multiple sequential components with the support for the parallel execution of activities within a component, which reduces the computational complexity and improves the scalability to support additional feature sets. First, the somatic mutation protein sequence is downloaded from GenBank [9], and the mutation sample is extracted from the protein sequence. The mutation sample can be represented by numerical numbers according to a predefined mapping scheme. Then, multiple feature sets are extracted from the protein sequence and its numerical representation. A feature selection approach is applied to each of the extracted feature sets to identify the optimal features to be used. The optimal feature sets are presented to a group of independent parallel classifiers. Finally, a model fusion technique is used to combine the classifiers' outputs and label the mutation as a

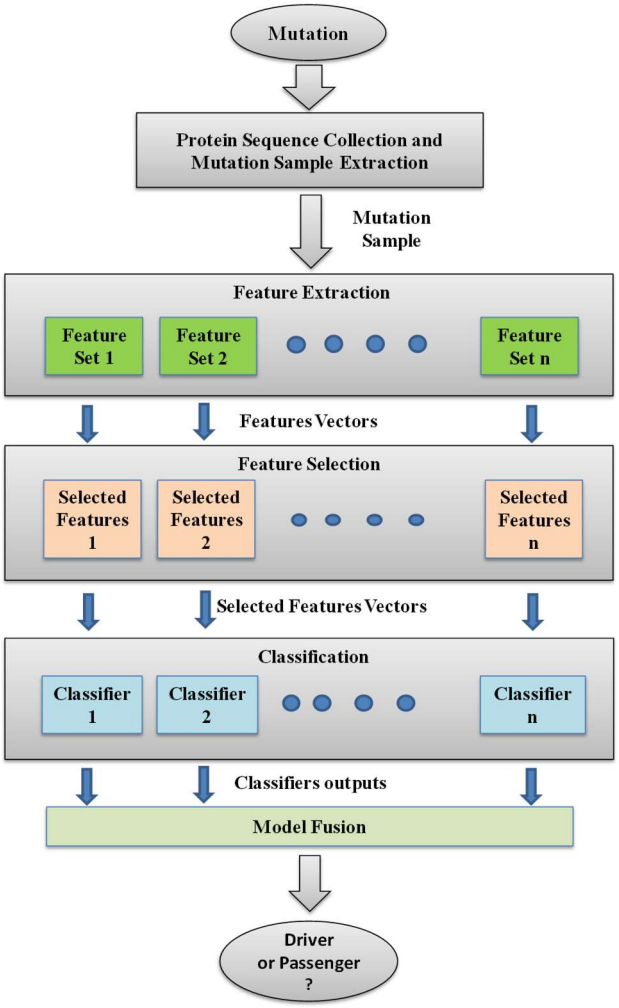

Figure 1. The proposed framework

driver or passenger mutation. The performance of the proposed framework is assessed and compared to existing techniques using the same mutation data set.

The paper is organized as follows. Section II summarizes some of the existing research to identify driver somatic mutations. Section III describes the proposed framework. The experimental results and conclusion are presented in sections IV and V.

II. Related Work

Several computational approaches have been proposed to automate the prediction of driver missense mutations. Some of these approaches rely on statistical methods to measure the frequency of mutation recurrence in a gene and compare it against a predefined statistical model to decide if they are driver mutations or not. MuSiC is an example of a tool that uses statistical tests to identify genes with driver mutations based on the mutation rate in a gene and it compares to an expected background mutation rate [10]. A survey of statistical approaches based on mutation recurrence rate can be found in [11].

There are approaches that leverage machine learning based techniques, where a set of features or attributes are extracted from the mutated sequence or gene to represent its characteristics, and then a machine learning model is trained to predict driver mutations based on the mutation characteristics. FATHMM is a mutation prediction approach that uses a group of Hidden Markov Models (HMMs) to capture the characteristics of protein sequences, then it can predict the impact of a mutation on the protein functionality and whether it is related to a specific disease or not [12][13]. Another approach based on support vector machines (SVMs) was proposed in [14], where a support vector machine is trained using a set of 126 features capturing several aspects of the mutations. The used features include amino acid residue changes, substitution scoring metrics, and annotated features retrieved from public databases. CHASM is an approach, based on Random Forest, to predict driver missense mutations [15]. It leverages the COSMIC somatic mutation database for its feature set. Orchid is another example of machine learning based approaches. It is a software package built using python and uses a random forest classifier to analyze cancer mutations [16].

III. Proposed Framework

A. Overview

The proposed framework consists of a series of sequential components as illustrated in Fig. 1. The first component is the protein sequence collection and mutation sample extraction, where the mutated protein sequences are downloaded from GenBank [9] and a mutation sample is extracted from the mutated protein sequence according to the mutation location and a predefined mutation window size. The extracted mutation sample is processed by the feature extraction component, which uses N parallel modules to generate N feature sets, where N is the number of feature sets used. The feature selection component reads the N extracted feature sets and filters out those low significant features in each set, resulting in N selected feature sets. The next component is the classification component which contains a group of N parallel SVM-based classifiers. Each classifier is trained to handle one feature set from the previous feature selection component. The group of classifiers can be trained, validated, and run in parallel. Finally, a model fusion approach is applied to combine the scores from the N classifiers and to generate a final label for the sample (either a driver or passenger mutation).

Four feature sets are used in this paper, setting N to 4. They are a wavelet analysis feature set, an amino acid index (AAindex1) feature set, an amino acid index (AAindex2) feature set, and a pseudo amino acid composition (PseAAC) feature set.

The details of each component are described as follows.

B. Input Data Format and Data Sets Used

1) Input Data Format: In this study, the same input data format as the one used in [8] is adopted. It reads two inputs

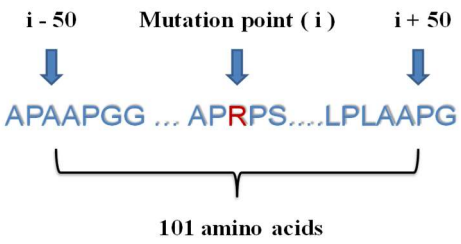

Figure 2. A mutation sample

for each somatic mutation input, namely the accession number of the mutated protein sequence using the NCBI RefSeq format and the mutation information which encodes the mutation location, old amino acid, and mutated amino acid (e.g., NP_001135977 R641W).

2) Data Set: The benchmark data set provided by the authors of the CHASM system is utilized [14]. This data set contains driver mutations and passenger mutations. The driver mutations were previously identified as playing a functional role in oncogenic transformation, while the passenger mutations were synthesized using the procedure described in the CHASM system. A total of 2534 driver mutations and 2894 synthesized passenger mutations were used in our experiemnts.

C. Protein Sequence Collection and Mutation Sample Extraction

This component generates the mutation sample corresponding to the input mutation. First, the protein sequence corresponding to the input mutation accession number is retrieved using the Matlab bioinformatics toolbox. Next, a mutation sample containing 101 amino acids is generated from the protein sequence with 50 amino acids flanking either side of the mutation point. Fig. 2 illustrates a mutation sample. Each mutation sample contains two sequences, one representing the protein sequence before mutation and one representing the protein sequence after mutation.

D. Feature Extraction

The feature extraction component contains N modules that can run in parallel, one module per feature set. Each module reads the mutation sample amino acids sequence and generates the feature vector for its feature set. The fact that all modules within the component can run in parallel limits the component computational complexity of the feature extraction component to be bounded by the maximum computational complexity of the N modules and therefore the component can scale to handle additional feature sets without much impact on the computational complexity. Four modules are defined and used in our experiments, since four feature sets are proposed in this paper. We adopt three feature sets that were proposed in our previous work [8]. A new feature set is introduced in this paper (i.e., AAIndex2). They are defined as follows.

1) Wavelet Features Module: Several approaches have been proposed to use wavelet analysis in DNA and protein sequence analysis [17]. This module utilizes wavelet analysis to capture that mutation sample charactrastics. First, it reads the amino acids sequence of the mutation sample before and after mutation, and represents it by a numerical sequence according to the mapping scheme defined in Table I [8]. Then, wavelet analysis using the Matlab wavelet toolbox with a continuous wavelet transform based on Gaussian wavelets function is applied to the resulting numerical sequence to generate the wavelet feature set. The scale vector has values from 1 to 101 with a step of 1, resulting a 101 by 101 wavelet coefficient matrix The wavelet power spectrum is calculated at each scale and thus a vector with 101 dimensions is generated, representing the wavelet feature set for the amino acid sequence. These steps are applied to the mutation sample before mutation and after mutation. A feature vector with 202 dimensions can then be obtained.

TABLE I. NUMERICAL REPRESENATION OF AMINO ACIDS

Amino Acid	Number Representation	Amino Acid	Number Representation
A	65	L	76
R	82	K	75
N	78	M	77
D	68	F	70
C	67	P	80
E	69	S	83
Q	81	T	84
G	71	W	87
H	72	Y	89
I	73	V	86

2) Amino Acid Index – AAIndex1 Features Module: AAIndex1 is a set of 566 amino acid indices, where each index has 20 numerical values representing some physicochemical and biological properties of the 20 amino acids [18]. The module reads the mutation amino acids sequence and computes the average sequence value for each index in the AAIndex1 set. The number of dimensiones in the resulting feature vector is equal to the number of indices in the AAIndex1 set. It can be forulmated as follows. Let S be a protein sequence with L amino acids, each amino acid is S^i where $1 \leq i \leq L$, and let $f_n(S^i)$ be the AAIndex1 value for amino acid S^i and index n. For an AAIndex set of N indicies, the feature vector is a vecotr of N values where each value F_n corresponds to the n^{th} index in AAIndex1 and can be represented as follows.

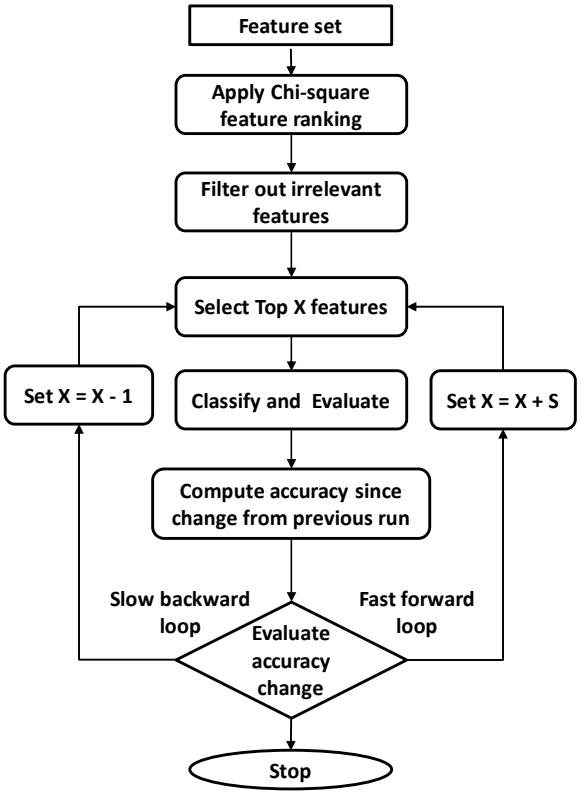

Figure 3. Feature selection process

$$F_n = \frac{1}{L}\sum_{i=1}^{L} f_n(S^i) \quad (1)$$

The above equation is applied to the amino acids sequence before mutation and after mutation. In our experiments, we used the first 544 indices only (N=544) and each protein sequence has 101 amino acids ($L = 101$), therefore the resulting feature vector has 1088 dimensions.

3) Amino Acid Index – AAIndex2 Features Module: AAIndex2 is a collection of 94 matrices, where each matrix contains 210 numerical values that capture the similarity between amino acids pairs. Each matrix represents a similarity or a mutation matrix. This module reads the mutation sample amino acids sequence before and after mutation and computes the average similarity per sequence between the sequence before mutation and after mutation, resulting in a single numerical value per each similarity matrix. The operation can be represented mathematically as follows. Let S_b and S_a be the two protein sequences before and after mutation where each sequence has L amino acids, S_b^i and S_a^i represent amino acids in S_b and S_a where $1 \leq i \leq L$, and let $f_n(S_b^i, S_a^i)$ be the AAIndex2 value representing the similarity between the two amino acids S_b^i and S_a^i at index n in AAIndex2. For a set AAIndex2 of N indicies, the feature vector is a vecotr of N values where each value F_n corresponds to the n^{th} index in AAIndex2 and can be represented as follows

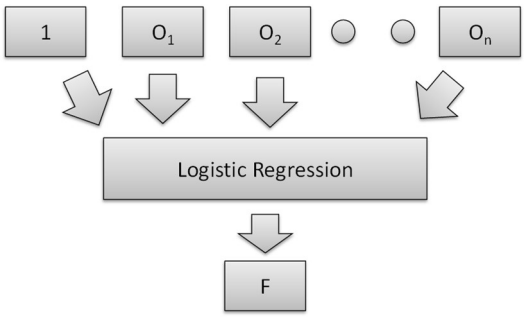

Figure 4. Model fusion component

$$F_n = \frac{1}{L}\sum_{l=1}^{L} f_n(S_b^l, S_a^l) \quad (2)$$

Since the above equation is applied once to both the mutation sequences before and after mutation and AAIndex2 contains 94 indices ($N = 9\,4$), therefore the resulting feature vector has 94 dimensions.

4) Pseudo Amino Acid Composition Features Module: This module is responsible for computing the PseAAC feature set from the mutation sample to capture the sequence order information. It was originally proposed in [19] for protein cellular attribute prediction. The following parameters are used in our experiment, i.e., λ and ω are set to 10 and 0.05. The PseAAC values are computed from the mutation sample before and after mutations, resulting in a feature vector with 60 dimensions. More details about the computation of PseAAC can be found in [19].

E. Feature Selection

The feature selection component consists of N identical modules, where each module is responsible for filtering out those low significant features in its corresponding feature set to reduce the dimensionality of the feature set and to increase the accuracy and performance of the framework. The steps applied to identify and filter out unnecessary features are identical for all modules. Four modules were employed in our framework as there are four feature sets. Since the approach proposed in our work in [8] was computationally expensive, here we propose an enhanced approach to improve the performance while not compromising the accuracy. Fig. 3 illustrates the proposed feature selection process. The process starts with applying chi-square on the feature vector to evaluate the correlation between the features and the class label, and then sort them in a descending order. Next, a fast-forward loop is used to iterate on the features by selecting the top X, initialized to one, ranked features with a step of S in each iteration, where S is set to an arbitrary value to control the speed of the loop. S was set to 5 in our experiments. After selecting the top X

features, a classifier is used to evaluate the features and measure the increase or decrease in accuracy between the current iteration and the previous iteration. If there is a decrease in accuracy, the backward loop is activated, where the module iterates on the features backward one by one till it reaches *X* with the maximum accuracy. Then it stops and reports the top *X* features to the following component in the framework. Another case where the fast-forward loop can be terminated is when the gain in accuracy over the previous iteration is too small or close to zero. It is worth noting that a drawback of this approach is the risk of being trapped in some local maxima.

F. Classification

The classification component employs a group of *N* SVM-based classifiers. Four classifiers are used in the current implementation to handle the four feature sets, where one classifier for the wavelet based features, one classifier for the AAIndex1 based features, one classifier for the AAIndex2 based features, and one for the PseAAC based features. Each feature vector generated by the feature selection component is passed to its corresponding classifier. Each classifier outputs a score to be processed by the model fusion component. Each classifier is trained and validated independently, and all classifiers run in parallel. The runtime for this component is not a function of the number of classifiers, and it is dependent on the classifier with the maximum runtime. The SVM-based classifiers are implemented using the LIBSVM library [20]. They use the radial basis function kernel, also known as the Gaussian kernel, as shown in Equation (3). Each SVM classifier can be tuned using two parameters, γ and C.

$$K(x_i, x_j) = e^{\gamma \|x_i - x_j\|^2} \quad (3)$$

G. Model Fusion

The model fusion component reads the output scores from the classifiers and integrates them into a single score to generate a final mutation label, namely either a driver or a passenger mutation. This process is called late fusion. It has been utilized in bioinformatics [8][21] and multimedia [22] [23][24][25][26][27][28]. Logistic regression was proposed in our work [8] to integrate the output of multiple classifiers into one output. In the current framework, logistic regression is used to implement the model fusion component. Fig. 4 illustrates the generalized implementation of the model fusion component implementation, where O_1, O_2, …, O_N are the *N* classifiers output scores and *1* is the a bias unit. Since four classifiers are used in our implementation, *N* is set to 4. Additional details about the implementation of the logistic regression is available at [8].

IV. EXPERIMENTS

Experiments were carried out to apply to filter out those low significant features in each feature set and to evaluate the performance of the proposed framework. The data set used in the experiments consists of 2534 driver mutations and 2894 synthesized passenger mutations. Table II illustrates the parameters used for the SVM-based classifiers during our experiments. The two experiments are described as follows.

The goal of the first experiment was to identify the minimum features that best represent each feature set. Four feature sets were used, wavelet features, AAIndex1 features, AAIndex2, and PseAAC features. First, the feature extraction component was applied to the driver and passenger mutations. Then, the feature selection component was used, and the selection process outlined in the previous section was applied to each of the extracted feature sets. The feature selection output summary is illustrated in Table III.

The results indicate that there is no reduction in PseAAC feature set, suggesting that all features are highly significant and important. This is consistent with the findings reported in our previous work [8]. There was a reduction in the other feature sets, AAIndex1, AAIndex2, and wavelet, with AAIndex1 features reporting the highest reduction percentage, followed by the wavelet features, and then AAIndex2. It is worth noting that the data set used for the feature selection experiment is different from the one used in our previous study [8], which may help explain why a higher reduction in the wavelet features was reported in [8], in addition to the fact that there is a difference between the current feature selection process and the one used in [8].

In the second experiment, the proposed framework performance is assessed by using the features selected in the first experiment to train and validate the framework. First, the selected features were passed to the classification component, where four SVM-based classifiers were employed. Each classifier was trained and validated using a 10-fold cross validation approach with one set of features. Then, the model fusion component reads the resulting four classifiers' output scores and splits the scores to 90% for training and 10% for testing randomly. Since the four classifiers generate output scores during test folds only, then the mutations presented to the model fusion component per fold, to train and test, represented 10% of the original mutations data set. As a result, the model fusion was tested using 54 mutations during each fold. Three metrics were used to evaluate the performance of the model fusion component output, namely accuracy, F1, and MCC as described in [8]. The process of training and testing the model fusion component was repeated for 12 runs, and then we excluded two runs, i.e., the run with the highest score and the run with the lowest score. The average of the remaining ten runs is reported in Table IV.

The results illustrated in Table IV indicate that the proposed framework slightly outperforms the results reported in [8]. Furthermore, the proposed enhancement to the feature selection process reduces the cost of feature selection without compromising the overall framework accuracy. Also, it is worth noting that since most activities in the proposed framework can run in parallel, it is possible to expand the implementation to consume additional features with the minimal runtime overhead.

TABLE II. SVM-BASED CLASSIFIERS PARAMETERS VALUES

Classifier	C value	γ value
Classifier 1 (Wavelet)	16	1
Classifier 2 (AAIndex1)	4	0.25
Classifier 3 (AAIndex2)	4	0.25
Classifier 4 (PseAAC)	8	0.125

TABLE III. FEATURE VECTOR SIZE BEFORE AND AFTER FEATURE SELECTION

Feature Set	Extracted Feature Vector Size	Selected Feature Vector Size
Wavelet	202	126
AAIndex1	1088	146
AAIndex2	94	61
PseAAC	60	60

TABLE IV. EXPERIMENTAL RESULTS

Framework	Accuracy	F1	MCC
Proposed Framework	0.9287	0.9165	0.8621
Framework in [8]	0.9258	0.9154	0.8548

V. CONCLUSION

The recent advancement in cancer genome sequencing techniques has boosted the availability and the scale of caner genomics data as well as the challenging diversity found in cancer genomic abnormalities. Hence, there is a need for scalable computational solutions that analyze the large scale data and identify useful patterns to improve the quality of cancer diagnosis and treatment. In this paper, a machine learning based framework to automatically detect driver mutations is proposed. The proposed framework builds on and enhances our previous framework to improve the driver mutation detection performance and to reduce the computational cost. It consists of a series of sequential components that perform the protein sequence data collection and mutation sample extraction, feature extraction, feature selection, classification, and model fusion. The current implementation of the framework leverages four types of features (wavelet, AAIndex1, AAIndex2, and PseAAC features). In addition, since the design of the framework supports running parallel operations within some of its components, it enables adding new features with the minimal impact on the computational complexity. Experimental results indicate that the proposed framework outperforms other approaches. In the future work, we will explore new feature sets and other model fusion techniques to further enhance the performance of the framework.

REFERENCES

[1] A. Jemal, F. Bray, M. M. Center, J. Ferlay, E. Ward, and D. Forman, "Global cancer statistics," CA Cancer J Clin, vol. 61, no. 2, pp. 69-90, 2011.

[2] K. Kochanek, S. Murphy, J. Xu, and E. Arias, "Mortality in the United States, 2016," NCHS Data Brief, no. 293, pp. 1-8, 2017.

[3] P. Futreal, L. Coin, M. Marshall, T. Down, T. Hubbard, R. Wooster, N. Rahman, and M. Stratton, "A census of human cancer genes," Nature Reviews Cancer, vol. 4, no. 3, pp. 177-183, 2004.

[4] B. Vogelstein, N. Papadopoulos, V. Velculescu, S. Zhou, L. Diaz, and K. Kinzler, "Cancer genome landscapes," Science, vol. 339, no. 6127, pp. 1546-1558, 2013.

[5] C. McFarland, J. Yaglom, J. Wojtkowiak, J. Scott, D. Morse, M. Sherman, and L. Mirny, "The damaging effect of passenger mutations on cancer progression," Cancer Research, 2017.

[6] C. Greenman, P. Stephens, R. Smith, G. L. Dalgliesh, C. Hunter, G. Bignell, H. Davies, J. Teague, A. Butler, C. Stevens, et al.,"Patterns of somatic mutation in human cancer genomes," Nature, vol. 446, no. 7132, pp. 153–158, 2007.

[7] M. R. Stratton, P. J. Campbell, and P. A. Futreal, "The cancer genome," Nature, vol. 458, no. 7239, pp. 719–724, 2009.

[8] A. Soliman, T. Meng, S.-C. Chen, S. Iyengar, P. Iyengar, J. Yordy, and M.-L. Shyu, "Driver missense mutation identification using feature selection and model fusion," Journal of Computational Biology, vol. 22, no. 12, pp. 1075-1085, 2015.

[9] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, "Genbank," Nucleic Acids Research, vol. 36, no. Supplement 1, pp. D25–D30, 2008.

[10] N. Dees, Q. Zhang, C. Kandoth, M. Wendl, W. Schierding, D. Koboldt, T. Mooney, M. Callaway, D. Dooling, E. Mardis, R. Wilson, and L. Ding, "MuSiC: Identifying mutational significance in cancer genomes," Genome Research, vol. 22, no. 8, pp. 1589-1598, 2012.

[11] B. Raphael, J. Dobson, L. Oesper, and F. Vandin, "Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine," Genome Medicine, vol. 6, no. 1, p. 5, 2014.

[12] H. Shihab, J. Gough, D. Cooper, P. Stenson, G. Barker, K. Edwards, I. Day, and T. Gaunt, "Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using Hidden Markov Models," Human Mutation, vol. 34, no. 1, pp. 57-65, 2012.

[13] H. Shihab, J. Gough, M. Mort, D. Cooper, I. Day, and T. Gaunt, "Ranking non-synonymous single nucleotide polymorphisms based on disease concepts," Human Genomics, vol. 8, no. 1, p. 11, 2014.

[14] H. Tan, J. Bao, and X. Zhou, "A novel missense-mutation-related feature extraction scheme for 'driver' mutation identification," Bioinformatics, vol. 28, no. 22, pp. 2948-2955, 2012.

[15] H. Carter, S. Chen, L. Isik, S. Tyekucheva, V. E. Velculescu, K. W. Kinzler, B. Vogelstein, and R. Karchin, "Cancer-specific high-throughput annotation of somatic mutations: computational prediction of driver missense mutations," Cancer Research, vol. 69, no. 16, pp. 6660–6667, 2009.

[16] C. Cario and J. Witte, "Orchid: a novel management, annotation and machine learning framework for analyzing cancer mutations," Bioinformatics, vol. 34, no. 6, pp. 936-942, 2017.

[17] T. Meng, A. Soliman, M.-L. Shyu, Y. Yang, S.-C. Chen, S. Iyengar, J. Yordy, and P. Iyengar, "Wavelet analysis in current cancer genome research: A survey,"" IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 10, no. 6, pp. 1442-14359, 2013.

[18] S. Kawashima and M. Kanehisa, "Aaindex: amino acid index database," Nucleic Acids Research, vol. 28, no. 1, pp. 374–374, 2000.

[19] K.-C. Chou, “Pseudo amino acid composition and its application in bioinformatics, proteomics and system biology,” Current Proteomics, vol. 6, no. 4, pp. 262–274, 2009.

[20] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, p. 27, 2011.

[21] M. Re, “Comparing early and late data fusion methods for gene expression prediction,” Soft Computing, vol. 15, no. 8, pp. 1497–1504, 2011.

[22] T. Meng, M.-L. Shyu, and L. Lin, “Multimodal information integration and fusion for histology image classification,” International Journal of Multimedia Data Engineering and Management, vol. 2, no. 2, pp. 54–70, 2011.

[23] T. Meng and M.-L. Shyu, “Concept-concept association infmation integration and multi-model collaboration for multimedia semantic concept detection,” Information Systems Frontiers, vol. 1, pp. 1–13, 2013.

[24] C. Chen, Q. Zhu, L. Lin, and M.-L. Shyu, “Web media semantic concept retrieval via tag removal and model fusion,” ACM Transactions on Intelligent Systems and Technology, vol. 4, no. 4, pp. 1-22, 2013.

[25] Y. Yan, M. Chen, S. Sadiq, and M.-L. Shyu, “Efficient imbalanced multimedia concept retrieval by deep learning on Spark clusters,” International Journal of Multimedia Data Engineering and Management, vol. 8, no. 1, pp. 1-20, 2017.

[26] Y. Yan, Q. Zhu, M.-L. Shyu, and S.-C. Chen, “A classifier ensemble framework for multimedia big data classification,” Proceedings of the 2016 IEEE 17th International Conference on Information Reuse and Integration, pp. 615-622, July 2016.

[27] T. Meng, L. Lin, M.-L. Shyu, and S.-C. Chen, “Histology image classification using supervised classification and multimodal fusion,” Proceesings of the IEEE International Symposium on Multimedia, pp. 145-152, December 2010.

[28] Q. Zhu, Z. Li, H. Wang, Y. Yang, and M.-L. Shyu, “Multimodal sparse linear integration for content-based item recommendation,” Proceedings of the IEEE International Symposium on Multimedia, pp. 187-194, December 2013.

`