DEPARTMENT OF BIOTECHNOLOGY
INDIAN INSTITUTE OF TECHNOLOGY
MADRAS
CHENNAI-600 036

Sequence neighborhoods enable reliable prediction of pathogenic mutations in cancer genomes

A Thesis

Submitted by

SHAYANTAN BANERJEE

For the award of the degree

Of

MASTER OF SCIENCE by

Research

June, 2021

THESIS CERTIFICATE

This is to undertake that the Thesis titled, "**SEQUENCE NEIGHBORHOODS ENABLE RELIABLE PREDICTION OF PATHOGENIC MUTATIONS IN CANCER GENOMES"** submitted by me to the Indian Institute of Technology Madras, for the award of **Master of Science (by Research),** is a bonafide record of the research work done by me under the supervision of Dr. Karthik Raman and Dr. Balaraman Ravindran. The contents of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Chennai 600 036
Date: 14th June, 2021

Shayantan Banerjee
Research Scholar

Dr. Karthik Raman
Research Guide
Associate Professor
Dept. of Biotechnology
IIT Madras

Dr. Balaraman Ravindran
Research Co-guide
Professor
Dept. of Computer Science
and Engineering
IIT Madras

ACKNOWLEDGEMENTS

I am forever indebted to my supervisors Dr. Karthik Raman and Dr. Balaraman Ravindran, for their invaluable guidance and continued support. I joined IIT Madras with very little to no knowledge of computational biology and machine learning. Thanks to their supervisory skills and the knowledge that I have gained from attending other courses at the institute, I feel motivated enough to continue my role as a researcher in this exciting research area. Also, I would like to thank all the GTC members, Dr. Himanshu Sinha, Dr. Swagatika Sahoo, and Dr. Amal Kanti Bera, for their valuable inputs, discussions, and efforts to generate the best version of my work.

I am also fortunate enough to be a part of the Initiative for Biological Systems Engineering (IBSE) and the Robert Bosch Centre for Data Science and Artificial Intelligence (RBCDSAI). Besides the top infrastructure and work environment, I have gained valuable experience and skills from the numerous talks, seminars, symposiums, and workshops organized by these research groups. Also, the weekly interactions with these groups' members have proved to be extremely beneficial for my research. I thank all the administrative staff from the Biotechnology department and the RBCDSAI for their continued support and cooperation.

I am also thankful to all members (former and current) of the Computational Systems Biology lab, especially Aarthi, Gayathri, Malvika, Priyanka, Devika, Lavanya, Samyugdha, Karthik, Philge, Keerthika, Anjana, Prem, and Debomita for their continued support as lab mates and for making my stay memorable. I could not have asked for a better experience in a laboratory setting. I would also like to acknowledge the help and support of some of my very close and dear friends, Vedant, Priyan, and Bhaswar. They made every single day of my stay at the Mahanadhi hostel entertaining, memorable, and restful.

Finally, I would like to thank the two very important persons in my life, my

parents, for their unwavering moral support and understanding. They are the ones who have motivated me to stay on track and attain my life goals. When I was a kid, my mother explained that doing research is very similar to writing a new chapter to an existing book to help make this world a better place. I hope that someday I will be competent enough to write a new chapter to this ever-growing body of knowledge and make her proud.

ABSTRACT

KEYWORDS: Sequence neighborhoods; Driver and passenger mutations; Machine Learning; Cancer.

Identifying cancer-causing mutations from sequenced cancer genomes hold much promise for targeted therapy and precision medicine. “Driver” mutations are primarily responsible for cancer progression, while “passengers” are functionally neutral. Although several computational approaches have been developed for distinguishing between driver and passenger mutations, very few have actually concentrated on utilizing the raw nucleotide sequences surrounding a particular mutation as potential features for building predictive models. Using experimentally validated cancer mutation data in this study, we explored various string-based feature representation techniques to incorporate information on the neighborhood bases immediately 5’ and 3’ from each mutated position. Density estimation methods showed significant distributional differences between the neighborhood bases surrounding driver and passenger mutations. Binary classification models derived using repeated cross-validation experiments gave comparable performances across all window sizes. Integrating sequence features derived from raw nucleotide sequences with other genomic, structural and evolutionary features improved the model’s overall predictive power in identifying pathogenic variants from five independent validation datasets. An ensemble predictor obtained by combining the predictions from two other commonly used driver prediction tools (CONDEL and Mutation Taster) outperformed existing pan-cancer models in prioritizing a literature-curated list of driver and passenger mutations. Using the list of true positive mutation predictions derived from our model, we identified a list of 138 known driver genes with functional evidence from various sources. Overall, our study underscores the efficacy of utilizing raw nucleotide sequences as features to distinguish between pathogenic and neutral variants from sequenced cancer genomes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AUROC	Area Under the Receiver Operating Characteristic (ROC) Curve
CGC	Cancer Gene Census
CI	Confidence Interval
CMC	Cancer Mutation Census
COSMIC	Catalogue Of Somatic Mutations In Cancer
GBM	Glioblastoma Multiforme
ICGC	International Cancer Genomics Consortium
JS	Jensen–Shannon
KDE	Kernel Density Estimation
MCC	Matthews Correlation Coefficient
NPV	Negative Predictive Value
OVC	Ovarian Cancer
PPV	Positive Predictive Value
SNV	Single Nucleotide Variant
TCGA	The Cancer Genome Atlas
TF-IDF	Term Frequency – Inverse Document Frequency

CHAPTER 1

Introduction

Cancer is caused due to accumulation of somatic mutations during the lifetime of an individual (Stratton *et al.*, 2009). These mutations can be due to both endogenous factors such as mistakes during DNA replication or exogenous factors such as substantial exposure to mutagens (tobacco smoking, UV light, etc) (Samet, 1989; Drake, 1969; Zhu *et al.*, 2017*a*). These somatic mutations can be of different types ranging from single-nucleotide variants (SNVs) to insertions and deletions (INDELs), copy-number aberrations (CNAs), and large genomic alterations known as structural variants (SVs) (Raphael *et al.*, 2014). Due to the advent of high-throughput sequencing, the identification of somatic mutations from sequenced cancer genomes has become easier. Whole-genome sequencing can comprehensively reveal all types of somatic mutations in the genome. In contrast, whole-exome sequencing concentrates only on the coding region of the genome and is cost-effective. Identifying cancer-causing mutations from large sequencing data is usually considered the first step towards precision oncology, where a cancer treatment can be adjusted based on the patient's mutational landscape (Garraway, 2013).

1.1 Somatic mutations in cancer

The clonal theory of cancer (Nowell, 1976) states that all cancerous cells start from a single cell where the first driver mutation had occurred. This is followed by subsequent cell divisions and positive selection, leading to a tumor containing a majority of cancer cells containing driver mutations (Figure 1.1). But not all somatic mutations are equally responsible for the development of cancer. Each somatic mutation in the cancer cell can be classified according to its contribution to the disease's overall progression. Driver mutations are the ones that are positively selected and confer a growth advantage on the cancer cells (Stratton

et al., 2009). On the other hand, passenger mutations neither offer any growth advantage nor contribute to overall cancer development. They are the "biologically inert" group of mutations that are present in the final tumor (Stratton *et al.*, 2009).

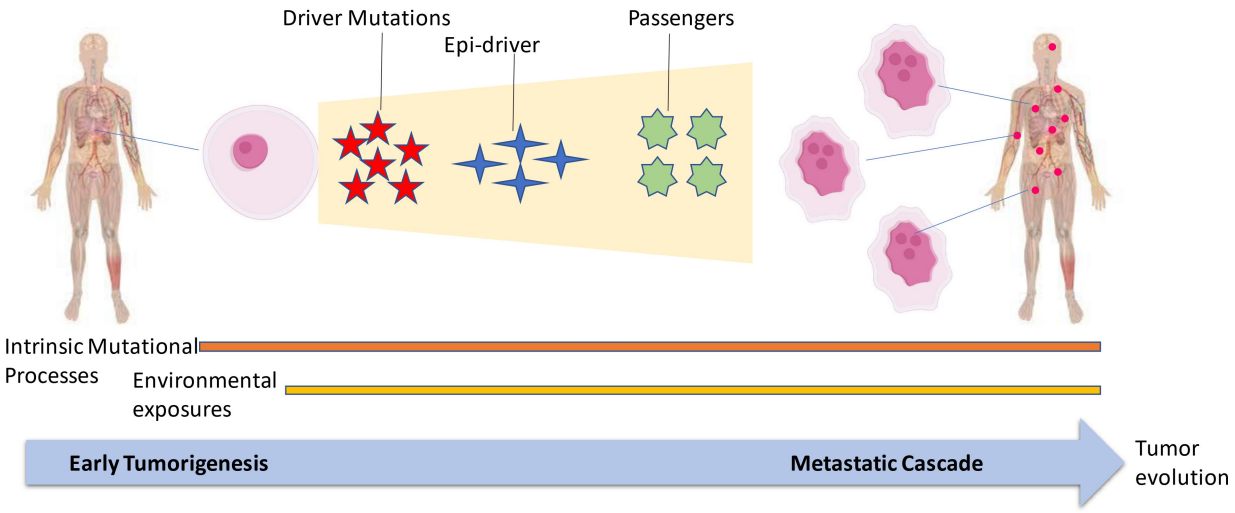

Figure 1.1: Contribution of both endogenous and exogenous factors to the overall mutational burden.

A cancer gene, by definition, carries a mixture of both driver and passenger mutations. A central aim in cancer genomics research is to distinguish between the two. With the advent of whole-genome sequencing studies, it has become increasingly easier to screen tens of thousands of samples and search for functional elements in the protein-coding genes, intergenic and intronic regions, etc. This task is challenging because of the substantially higher number of passenger mutations as compared to drivers.

1.2 Detection of somatic mutations in a cancer genome

Large-scale cancer genome sequencing projects led by consortia such as the International Cancer Genome Consortium (ICGC) (Zhang *et al.*, 2011) and The Cancer Genome Atlas (TCGA) (Weinstein *et al.*, 2013) employ whole-genome

and whole-exome sequencing thousands of samples. Whole-exome sequencing provides a more targeted approach and provides more in-depth coverage of the individual genomes but at the cost of ignoring the non-protein-coding regions. Despite dramatic advances, DNA sequencing technologies still face significant limitations in identifying and prioritizing somatic mutations from a heterogeneous collection of cancer and normal cells. *Tumor purity* is defined as the proportion of cancer cells in the given tumor. An accurate estimation of the tumor purity is, therefore, essential in the detection of somatic mutations.

1.2.1 Tumor purity estimation

Most of the major DNA-sequencing technologies produced by Illumina, Ion Torrent, 454, Pacific Biosciences, and others produce millions of short sequences known as reads mostly 50-100 bp in length. Detection of somatic mutations involves aligning the reads to the reference genome and identifying the differences. A matched normal sample from the same individual is also aligned with the reference genome to differentiate between somatic and germline mutations. Sequencing artifacts are the most common source of noise that is introduced in the experiments described above. These include GC-bias, strand bias, PCR duplicates, skipped bases resulting in apparent insertions and deletions among others (Ding *et al.* (2012)). This results in many false-positives (incorrectly identified variants) and false-negatives (missing variants). Bulk tumor-sequencing studies contain a mixture of both normal and cancer cells. Contamination of normal cells results in a reduced signal (low sensitivity and specificity) for detecting high-confidence somatic mutations. This phenomenon is also referred to as intra-tumor heterogeneity. Some of the most commonly used tools for somatic mutation detection and tumor purity estimation of sequenced whole-genomes/exomes are shown in Table 1.1.

1.2.2 Computational prioritization of somatic mutations

Once we have determined the set of somatic mutations in a sequenced cancer genome, the next step is to identify driver mutations responsible for the disease

Commonly used tools for somatic mutation detection			
Data	**Tool name**	**Description**	**Reference**
SNV	MuTect	Works for both whole genome and whole exome sequencing data and can detect low-frequency variants	Cibulskis *et al.* (2013)
SNV	VarScan 2	A platform-independent tool that works for both whole-genome and whole-exome sequencing data and only calls variants that passes a predefined threshold for base quality, read depth and allele frequency.	Koboldt *et al.* (2012)
SNV	Strelka	A Bayesian approach for identifying somatic variants not requiring purity estimates beforehand. Maintains a high frequency even for impure samples.	Saunders *et al.* (2012)
CNA	BreakDancer	Detects structural variants by clustering paired-end reads.	Chen *et al.* (2009)
Commonly used tools for tumor purity estimation			
SNV	ABSOLUTE	Uses an outlier detection method to detect subclonal heteregenity.	Carter *et al.* (2012)
CNA	THetA	Able to detect multiple tumor subpopulations if they differ by copy number aberrations.	Oesper *et al.* (2013)

Table 1.1: Commonly used tools for somatic mutation detection and tumor purity estimation

progression. Distinguishing between the deleterious driver mutations and the neutral passengers solely based on the type of mutation (C>A, T>G, CTG>CGA, and others) is very difficult since the effect of most mutations is not well understood, (Raphael *et al.*, 2014) and this holds even in the simplest case of single nucleotide variants present in the coding regions of well-studied protein-coding genes. There are primarily three approaches for *in silico* prioritization of driver mutations from sequenced cancer genomes; recurrence, functional impact, and commonly affected pathways (Figure 1.2).

1.2.3 Recurrent single point mutations

One common approach to prioritize driver mutations is to identify recurrent mutations. By definition, recurrent mutations are those variations in the cancer genome (at the level of a single nucleotide, a codon, a single gene, or a pathway)

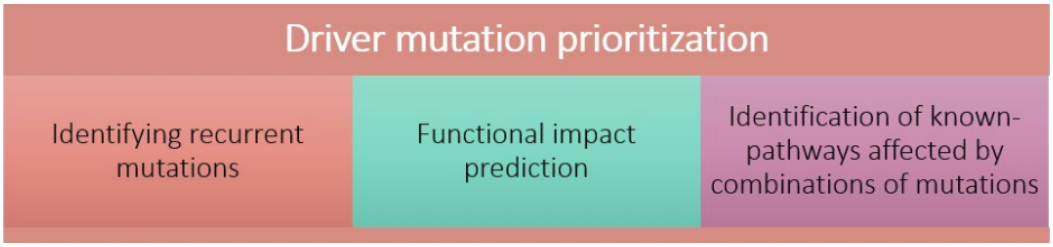

Figure 1.2: The three main approaches for identifying driver mutations

that occur more than what is expected by chance, according to a Background Mutation Rate (BMR) (Vogelstein *et al.*, 2013). The calculation of the BMR is central to exploring the recurrence concept. It is also defined as the probability of observing a passenger mutation at a particular genomic location. The BMR calculation is non-trivial and is often complicated by a variety of factors, some of which are listed below:

- The BMR is not constant across samples, (Cancer Genome Atlas Research Network *et al.*, 2008) and calculation are often difficult for hypermutated samples.
- Certain genomic regions are more prone to localized hypermutation, also known as *kataegis* (Nik-Zainal *et al.*, 2012).
- The BMR varies across the genome and depends on the nucleotide context (Sjöblom *et al.*, 2006) and the mutation type (Cancer Genome Atlas Research Network *et al.*, 2011).

Inaccurate estimation of the BMR eventually leads to a large number of false positives and false negatives. This issue is somewhat non-existent for well-known cancer genes such as *TP53*, which are recurrently mutated in many samples and are easily identified by most driver gene detection algorithms. However, the real challenge lies in identifying rarely mutated genes that do contribute to cancer progression. Hence, either an accurate estimation of the BMR or the large sample size is often necessary to identify rare driver genes.

1.2.4 Functional impact prediction

In addition to the recurrence approach, another method of distinguishing between driver and passenger mutations involves predicting the functional impact of a mutation using additional information about the sequence context, protein

structures, etc. These methods are applied mainly on non-synonymous SNVs that affect the corresponding amino acid and consequently the final protein product. Tools that predict germline mutations' functional impact include SIFT (Kumar *et al.*, 2009), PROVEAN (Choi *et al.*, 2012) and Polyphen-2, (Adzhubei *et al.*, 2010). Some of recent methods such as CHASM (Carter *et al.*, 2009), Oncodrive-FM (Gonzalez-Perez and Lopez-Bigas, 2012) and MutationAssesor (Reva *et al.*, 2011) also focuses on somatic mutations. Clustering of known missense mutations in particular genomic positions, such as the V600E mutation in *BRAF*, (Davies *et al.*, 2002) can also be used to predict the functional impact of cancer variants.

1.2.5 Identification of altered pathways

Interaction between genes and proteins in the form of signaling networks are often affected by somatic mutations (Hanahan and Weinberg, 2011). Identification of patterns of mutational recurrence from these complex networks is often not straightforward. Some genes might be mutated with very low frequency than others to show any statistical significance. One popular method known as the Gene Set Enrichment Analysis (GSEA) (Subramanian *et al.*, 2005) measures whether the given set of genes has more high-ranking genes than random. More sophisticated methods such as PathScan (Wendl *et al.*, 2011) evaluates the patient-wise mutational enrichment in a given gene set. Examining mutations in known gene sets across large-scale protein-protein interaction network databases is also a popular method to prioritize driver mutations. The STRING database (Franceschini *et al.*, 2012) is one such database that is often used for this purpose. HotNet (Vandin *et al.*, 2011) identifies subnetworks from large interaction networks with significantly more mutations than random. A list of some of the most commonly used driver prioritization tools and their respective categories (recurrence, functional impact, and pathways) is shown in Table 1.2.

Approach	Tool name	Description	Reference
Recurrence	MutSigCV	Calculates the BMR using a variety of mutational and genomic features	Lawrence *et al.* (2013)
Recurrence	MuSiC	Uses a statistical method to find significantly mutated genes and altered pathways. User-specified genomic regions of interest are also accepted	Dees *et al.* (2012)
Recurrence	DrGaP	Uses Bayesian modelling to estimate the BMR	Hua *et al.* (2013)
Functional impact (Non-cancer specific)	SIFT	Uses conserved amino acids to predict the impact of a non-synonymous SNV on the protein product	Kumar *et al.* (2009)
Functional impact (Non-cancer specific)	Polyphen-2	Machine learning-based method to predict the effect of non-synonymous SNVs on the final protein product	Adzhubei *et al.* (2010)
Functional impact (Non-cancer specific)	MutationAssessor	Uses evolutionary conservation based features to assess the functional impact of amino acid changes	Reva *et al.* (2011)
Functional impact (Cancer-specific)	CHASM	Uses machine learning models trained on evolutionary and protein-based features to differentiate between drivers and passengers	Carter *et al.* (2009)
Altered pathways	GSEA	Measures whether a given set of genes has more high-ranking genes than random	Subramanian *et al.* (2005)
Altered pathways	PathScan	Evaluates the patient-wise mutational enrichment within a given gene set	Wendl *et al.* (2011)
Altered pathways	HotNet	Identifies subnetworks from large interaction networks with significantly more mutations than random	Vandin *et al.* (2011)
Altered pathways	Dendrix	Identifies set of genes with mutually independent groups of mutations	Vandin *et al.* (2011)

Table 1.2: Commonly used tools for identifying driver mutations or genes classified on the basis of three main approaches (recurrence, functional impact and altered pathways)

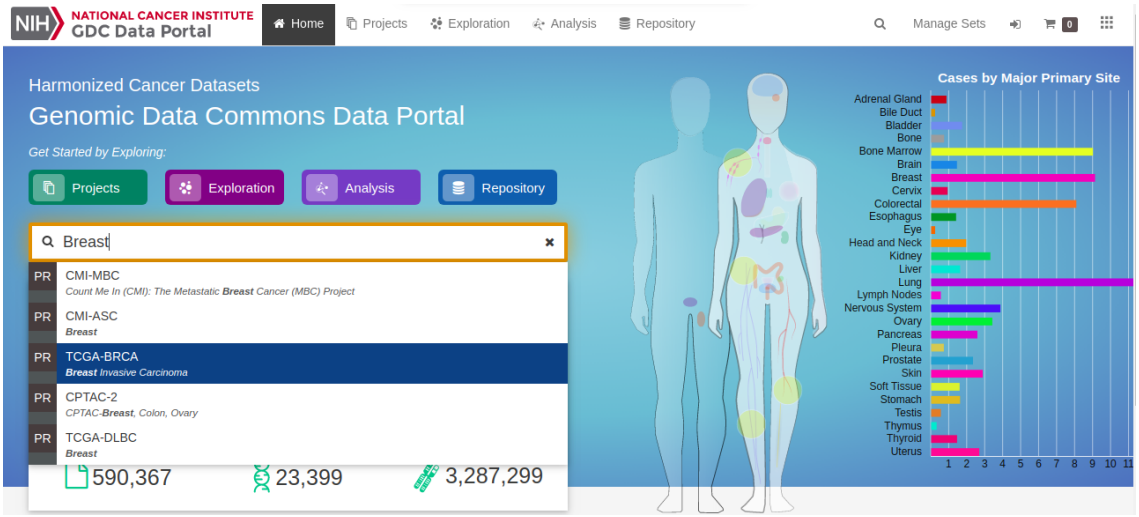

Figure 1.3: TCGA database containing list of cancer-causing mutations from large-scale sequencing studies (https://portal.gdc.cancer.gov/)

1.3 Cancer mutation databases

To build our predictive models, we use two of the most commonly used cancer mutation databases to build our training set. A summary explaining the structure and content of both the databases is given below.

1.3.1 The Cancer Genome Atlas (TCGA)

The Cancer Genome Atlas is a publicly available database containing normal-tumor matched sequencing data from over 84,000 patients spanning 68 different cancer types. For our study, we are mainly interested in single nucleotide variants (SNVs). It is fairly straightforward to find SNVs for a particular tissue type in TCGA. The list of SNVs has annotations obtained using tools like the Polyphen-2, SIFT, and VEP. This database also contains information regarding the consequence type, such as "missense variant", "upstream gene variant" etc. The variant files are available mainly in the "vcf" or "maf" format. The VCF or the "Variant Calling Format" is used to store the variation in the gene sequence in text-based format. The MAF or the "Mutation Annotation Format" is similar to the VCF format but also contains the mutation annotation information.

1.3.2 Catalog of Somatic Mutations In Cancer (COSMIC)

The Catalogue of Somatic Mutations in Cancer (COSMIC) is the single largest curated source of somatic mutations belonging to different cancer types. Data in COSMIC are regularly collected from peer-reviewed large scale sequencing studies, other databases such as the TCGA (Weinstein *et al.*, 2013) and the ICGC, (Zhang *et al.*, 2011) and the Cell Lines Project, based at the Wellcome Sanger Institute (Iorio *et al.*, 2016). This list is regularly updated, and newer cancer genes and their associated variants are added. Some of the most commonly used dedicated tools such as the Genome Browser (https://cancer.sanger.ac.uk/cosmic/browse/genome), Cancer Browser (https://cancer.sanger.ac.uk/cosmic/browse/tissue), Hallmarks of Cancer (https://cancer.sanger.ac.uk/cosmic/census-page/pten), Cancer Gene Census (https://cancer.sanger.ac.uk/census), COSMIC-3D (https://cancer.sanger.ac.uk/cosmic3d), and Mutational Signatures (https://cancer.sanger.ac.uk/cosmic/signatures) help us explore the database. For our purposes, however, we focused on the dataset titled "COSMIC Mutation Data" that contain a tab-separated list of coding variants for >30 different cancer types. Mutations are available for both the GRCh37 and GRCh38 genome build. Figure 1.4 above summarizes the data curation process for COSMIC from different sources and the various available tools for users.

1.4 Commonly used features for distinguishing between driver and passenger mutations

Efficient feature representation is central to building any predictive models. In the following section, we discuss some of the commonly used features to distinguish between driver and passenger mutations from large-scale cancer sequencing studies.

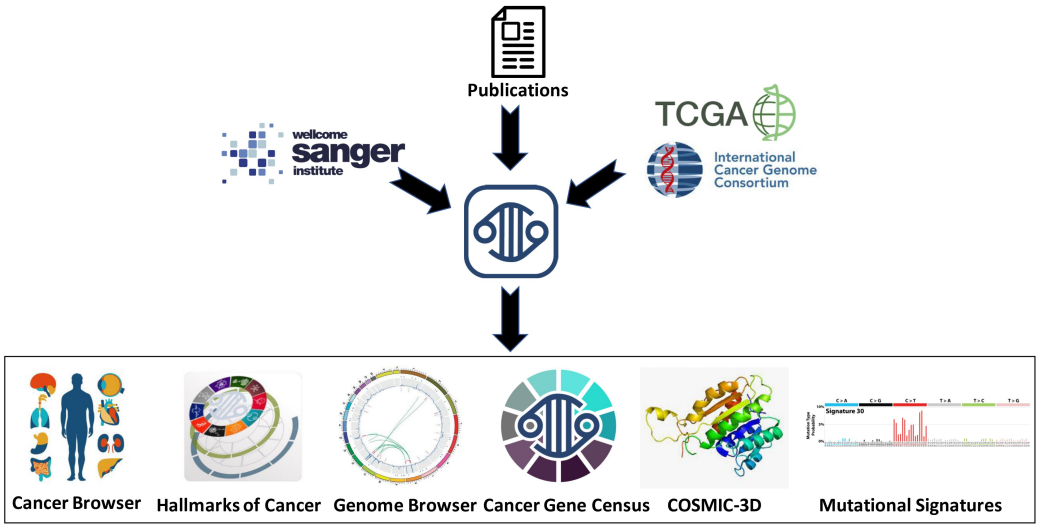

Figure 1.4: COSMIC data curation process and the available tools for exploring the database

1.4.1 Genomic and conservation features

The following gene-based features have been used to understand the effect of missense variants on cancer progression.

- Gene size: The size of the coding region of a gene and the mutation rate often co-vary, which can be estimated from the NCBI Consensus CDS project. Theoretically, a gene with a larger coding region will be more susceptible to mutations, but other factors such as location and GC content also influence the mutation rate.
- Potential sites for mutations: The number of possible sites for a single nucleotide substitution (C>T) to occur can be used to study the impact of mutations. There are three point substitutions possible for each nucleotide position.
- Nucleotide composition: The proportion of the four nucleotides in the coding region of a gene.
- CpG dinucleotides: Genes having a higher proportion of CpGs have a higher mutation rate and consequently a higher proportion of missense mutations (Millar *et al.*, 2002).
- Evolutionary conservation: Some studies have indicated a direct correlation between mutation rate and evolutionary conservation of a gene (Michaelson *et al.*, 2012). Conservation scores such as the phyloP scores (Pollard *et al.*, 2010) are derived from multiple sequence alignments of vertebrate genomes. Mutations at conserved sites lead to more deleterious outcomes than in non-conserved regions.

- Gene expression levels: Mutation density is defined as the proportion of the total number of mutations in the coding region of a gene. Some studies have shown a negative correlation between the gene expression level and mutation density (Hodis *et al.*, 2012; Park *et al.*, 2012*a*).
- Chromatin accessibility: Some studies have established the relationship between mutability of a region in the genome and chromatin accessibility (Thurman *et al.*, 2012). Formaldehyde-Assisted Isolation of Regulatory Elements sequencing (FAIRE-seq) data uses a peak-calling algorithm to analyze open chromatin regions in the genome.

1.4.2 Consequence-based features (for coding regions only)

The ENSEMBLE Variant Effect Predictor (VEP) can be used to extract characteristic features for specific genomic locations. These include transcript features, changes in amino acid, allele frequencies, and scores from variant pathogenicity prediction tools such as SIFT and PolyPhen-2.

1.4.3 Sequence-based features

Several studies have indicated a relationship between the local DNA-sequence context and the mutation rate (Rogozin *et al.*, 2018; Hodgkinson and Eyre-Walker, 2011; Stratton *et al.*, 2009). The maximum variation in mutation rate has been shown to be dependent on the sequence context (Michaelson *et al.*, 2012). A similar observation has been made for germline mutations as well (Carlson *et al.*, 2018). To sum up, mutation rate can be affected by the local DNA context, histone modifications, GC content, CpG dinucleotides etc. In the later chapters, we will explore how to utilize the adjacent nucleotide context to predict the pathogenicity of cancer-causing mutations.

1.5 Objectives

In this study, we aim to utilize the neighborhood sequence context to predict the pathogenicity of cancer-causing mutations. Broadly, our objectives were as follows:

- To explore different nucleotide sequence-based feature representations and use the same to study the underlying distributional differences between driver and passenger mutations.
- To develop a novel machine learning-based driver mutation prediction algorithm, **NBDriver**, that utilizes the sequence context and other structural, genomic and evolutionary features to differentiate between drivers and passengers.
- To study the effect of the combination of various mutation effect predictors on the overall predictive power of the model in differentiating between driver and passenger mutations.

1.6 Organization of the thesis

The rest of the thesis is organized as follows: Chapter two provides a review of the literature and a brief background of the current work. Chapter three discusses our approach for studying sequence-based features for identification of driver mutations and integration of the same with other genomic features to build a robust machine learning model. Finally, chapter four concludes the thesis and discusses some possible future directions.

CHAPTER 2

Background and Related Work

This chapter serves as a brief introduction to the different machine learning (ML) models used in Chapter 3 and a broad overview of a few recently published sequence-context-based driver prioritization models.

2.1 Machine Learning methodology

ML is a branch of computer science that deals with developing algorithms that have the ability to learn and improve from experience. The input data for ML algorithms is usually of the form (X, y), where X denotes the set of variables or features and y denotes the output. The output can be either discrete (for classification problems) or continuous (for regression problems). The main goal of machine learning algorithms is to find a mapping/function that best maps the input features (X) to the output variables y. Broadly, ML can be divided into two categories: *supervised* and *unsupervised* learning.

2.1.1 Supervised learning

Supervised learning refers to a subtype of ML algorithms that require labeled training data to derive a mapping function from a set of training examples. The inferred function can be used to map new test instances. These algorithms are judged based on the average squared error (for regression) or fraction of misclassified instances (for classification). In this scenario, a possible supervised learning approach will derive a function that uses a set of mutational features to predict whether a given mutation is driver/passenger. This approach can be implemented for tissue-specific (e.g., lung, breast, kidney, etc.) cancer samples or a pan-cancer framework.

2.1.2 Unsupervised learning

Unsupervised learning does not require labeled training data for learning purposes. They are mainly used to find patterns in the training data (such as clusters) and reduce redundant features in a high-dimensional training set. Some studies have explored the concept of clustering of driver mutations in specific regions of cancer genomes using unsupervised learning techniques (Tamborero *et al.*, 2013; Arnedo-Pac *et al.*, 2019).

2.2 Feature representation

ML algorithms cannot take raw sequence-based data directly as input. We have to map them to some numerical format using different feature transformation techniques. Since we are dealing with raw sequence data consisting of four letters (A, T, G, and C), we used three of the most commonly used feature mappings for string variables.

2.2.1 One-hot encoding

In one-hot encoding, we represent each categorical variable as a binary vector. The number of levels (or types) of the categorical variable is used to decide the length of the vector. For instance, a single DNA nucleotide can be represented as a binary vector of size 4 containing all zero values except the index of that nucleotide, which can be marked as 1. Thus "*A*" can be encoded as [1, 0, 0, 0], "*G*" as [0, 1, 0, 0], "*C*" as [0, 0, 1, 0] and "*T*" as [0, 0, 0, 1]. So, the nucleotide sequence "*ATT*" can be represented as the binary vector (100000010001). It must be noted that if there exists an ordinal relationship between the categories, then one-hot encoding must be avoided and integer encoding must be adopted.

2.2.2 Bag-of-Words model

As discussed in the previous section, we cannot feed text as input directly to ML algorithms. A simple yet effective model to represent text-based features in machine learning is called the **Bag-of-Words** model, or **BoW**. In this model, the order of words in a given document is not considered at all, and instead the focus is placed on the occurrences of words. A word matrix is then constructed, where sentences are represented as rows and all possible words in the document (or vocabulary) are represented as columns. Each entry in the matrix is then populated with the count or frequency of words in the document. The two most popular ways to calculate each word's count or frequency in the word matrix are discussed below.

2.2.3 Bag-of-Words for DNA nucleotide sequences: Overlapping *k*-mers

A sequence of k consecutive nucleotides composed of DNA nucleotides or amino acids is called a k-mer. The frequency of a set of k-mers of a given size in a particular species' genome can be used as a signature of the underlying genomic sequence. There are different factors affecting the frequency of a k-mer of a given size and some of them are listed below.

$k=1$

In its most basic form, when $k=1$, there are four nucleotides - A, T, G and C. Due to the extra hydrogen bond between G and C, GC bonds are more thermally stable than the AT bonds. As a result, there is a higher proportion of G and C bases as compared to A and T (GC content) in a mammalian genome.

$k=2$

Unlike GC content, which displays a significant variation, dinucleotide biases are more or less constant throughout the genome. CpG sites are regions in the

genome where a cytosine is followed by a guanine in the 5'-3' direction. CpG islands have a high concentration of CpG sites. Methylation of CpG dinucleotides can cause a change in the gene expression. Studies have shown that the loss in expression of cancer genes is sometimes due to methylation of CpG islands (Illingworth *et al.*, 2010).

k=3

There are 64 distinct 3-mers that represents each amino acid. These are also known as *codons*. A particular amino acid can be represented by multiple codons. The differences in the frequency of codons in the coding DNA is also known as the Codon Usage Bias (or CUB) (Hershberg and Petrov, 2008). Proper distinction must be made between the frequency of 3-mers and the codon usage bias.

k=4

It has been shown that phylogenetically similar organisms share similar tetranucleotide frequencies to maintain genomic stability (Perry and Beiko, 2010).

Overlapping k-mers

The overlapping list of k-mers from a DNA sequence is obtained by extracting the first k characters followed by shifting one nucleotide at a time before the start of the next k-mer. Any sequence of length n will contain $n-k+1$ k-mers. A diagram showing all possible k-mers (k=1, 2, 3, 4) from an arbitrary sequence of length eight is shown in Figure 2.1.

2.2.4 Count vectorizer

The simplest way to tokenize a set of documents, construct a vocabulary and encode a new document is through a vectorization technique called **Count vectorizer**. The encoded vector output is of the same size as the vocabulary and contains the integer frequency of each word in the document. Figure 2.2 shows

k=1	*k*=2	*k*=3	*k*=4
Sequence: ATCGATCAC	Sequence: ATCGATCAC	Sequence: ATCGATCAC	Sequence: ATCGATCAC
1-mer #0: A	2-mer #0: AT	3-mer #0: ATC	4-mer #0: ATCG
1-mer #1: T	2-mer #1: TC	3-mer #1: TCG	4-mer #1: TCGA
1-mer #2: C	2-mer #2: CG	3-mer #2: CGA	4-mer #2: CGAT
1-mer #3: G	2-mer #3: GA	3-mer #3: GAT	4-mer #3: GATC
1-mer #4: A	2-mer #4: AT	3-mer #4: ATC	4-mer #4: ATCA
1-mer #5: T	2-mer #5: TC	3-mer #5: TCA	4-mer #5: TCAC
1-mer #6: C	2-mer #6: CA	3-mer #6: CAC	
1-mer #7: A	2-mer #7: AC		
1-mer #8: C			

Figure 2.1: All possible overlapping 1-mers, 2-mers, 3-mers and 4-mers from the given sequence

Data= ['the','quick','brown','fox','jumps','over','the','lazy','dog']

the	quick	brown	fox	jumps	over	lazy	dog
2	1	1	1	1	1	1	1

Figure 2.2: Construction of a word matrix using a Count vectorizer

the construction of a word matrix using the Count vectorizer.

2.2.5 TF-IDF vectorizer

The TF-IDF vectorizer stands for **T**erm **F**requency- **I**nverse **D**ocument **F**requency Vectorizer. This algorithm is mostly used to represent how important a particular word is for the entire corpus. It has two essential parts.

- Term Frequency (TF): It is essentially the raw count of a particular word in a document.
- Inverse Document Frequency (IDF): This quantity measures whether a particular word is common/rare in the given corpus. It is the logarithmic ratio between the total number of documents in the corpus and the number of documents containing the given term. Thus, rarer words are given a higher IDF score as compared to common words.

Finally, the TF-IDF score of a particular term is the product of the Term Frequency (TF) and the Inverse Document Frequency (IDF). Figure 2.3 shows the construction of a word matrix using the TF-IDF vectorizer.

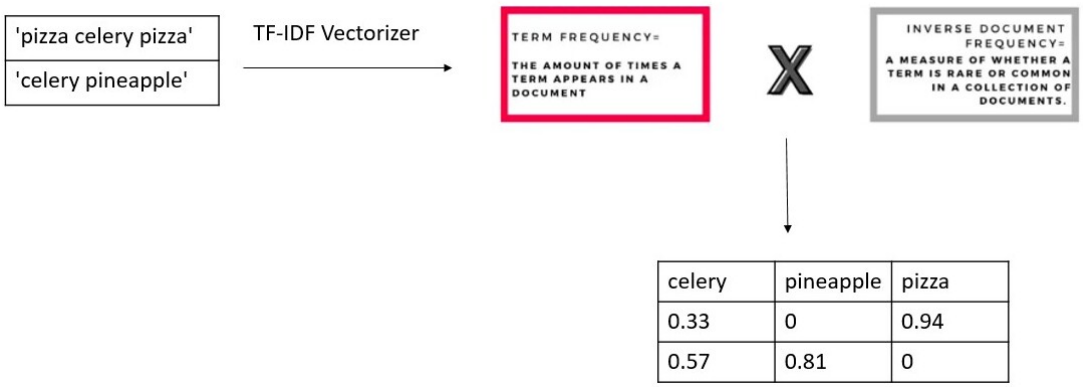

Figure 2.3: Construction of a word matrix using TF-IDF vectorizer

2.3 Density estimation

Density Estimation is a technique to construct an unknown underlying probability distribution from a given data set (Figure 2.4). Before going into the details of how the estimation of an unknown probability distribution is derived, let us look at a few basic underlying concepts.

2.3.1 Probability density

Probability density is the relationship that exists between the outcome of a random variable X and its probability. For a continuous random variable X, the shape of the density function across the entire domain of X is known as its probability distribution. We can calculate the mean and variance of a random variable from the probability distribution. However, the underlying distribution is often not available to us since we don't have access to all possible outcomes of a random variable X. Only a sample of observations is given to us. Hence, we need to estimate the *density function* from the sample of observations available.

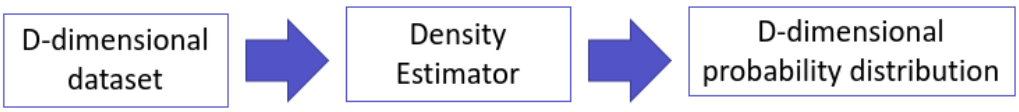

Figure 2.4: A density estimator inputs a D-dimensional data set and outputs the estimated D-dimensional probability distribution

2.3.2 Histograms

In the simplest case, for 1-D data, we use a 1-D density estimator or a histogram. The data are divided into discrete bins and the frequency of the data points in each bin is calculated. An intuitive visualization process is also adopted to interpret the results. Figure 2.5 shows a histogram representing data drawn from two Gaussians, $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$.

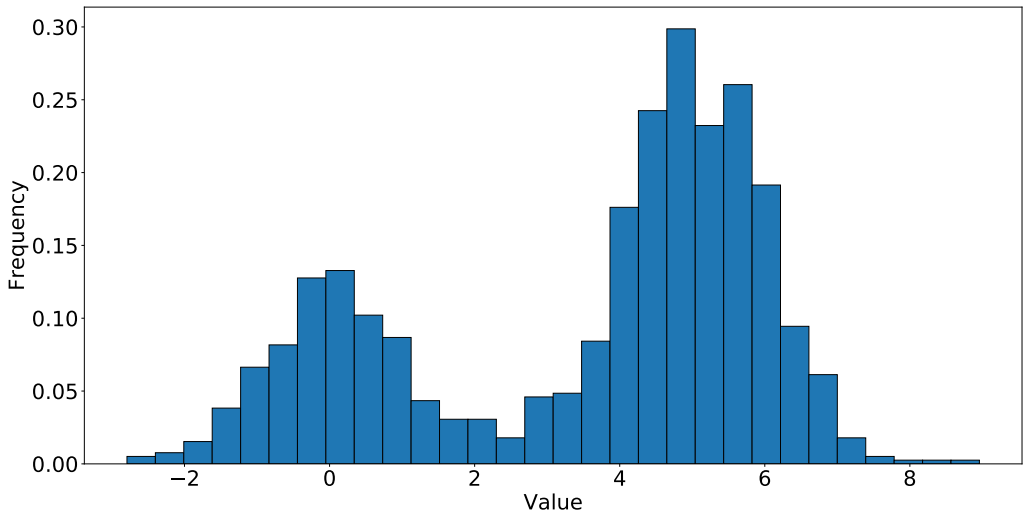

Figure 2.5: A histogram representing data drawn from two Gaussians (VanderPlas, 2016)

One of the major issues with histograms is that if we choose a different size of bins then we will get qualitatively different representations of the data. For instance, with just 20 points from the same data set and with a different choice of binning can lead to a different interpretation of the data as shown in Figure 2.6. On the left, we can see that the histogram represents a bimodal distribution but on the right we observe an unimodal distribution. The surprising fact is that these two histograms were built on the same data.

2.3.3 Parametric density estimation

Certain standard distributions repeat across domains, and the shape of the histograms will often match a well-known probability distribution. If we can identify the distribution from its shape, we can estimate the distribution parameters

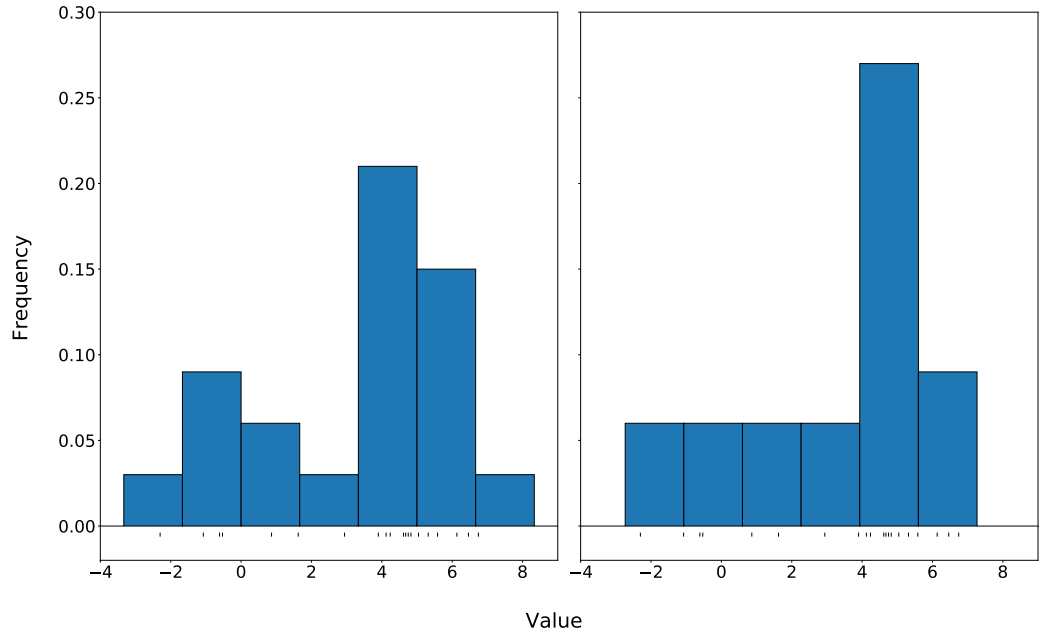

Figure 2.6: Choice of binning may lead to an entirely different qualitative representation of the same data (VanderPlas, 2016)

from the given sample of data points. This process is also known as *parametric density estimation*. Figure 2.7 displays an example of the parametric density estimation where histogram of the original data sample is plotted along with the estimated PDF.

2.3.4 Non-parametric density estimation

In some cases, we might not be able to represent our data sample with common probability distributions. Pre-defined distributions are often absent in such cases, and alternative methods also referred to as *non-parametric* methods are required to estimate the PDF. The most common non-parametric approach is known as ***kernel density estimation***. One important point to note is that non-parametric methods do not mean an absence of parameters but that the number of parameters is not fixed *a priori*. A kernel density estimator has two parameters: the *kernel*, which is used to specify the shape of the PDF, and the *bandwidth*, which acts as a smoothing parameter controlling the kernel size at each point. Figure 2.8 shows an example of a kernel density estimation plot for a bimodal data sample.

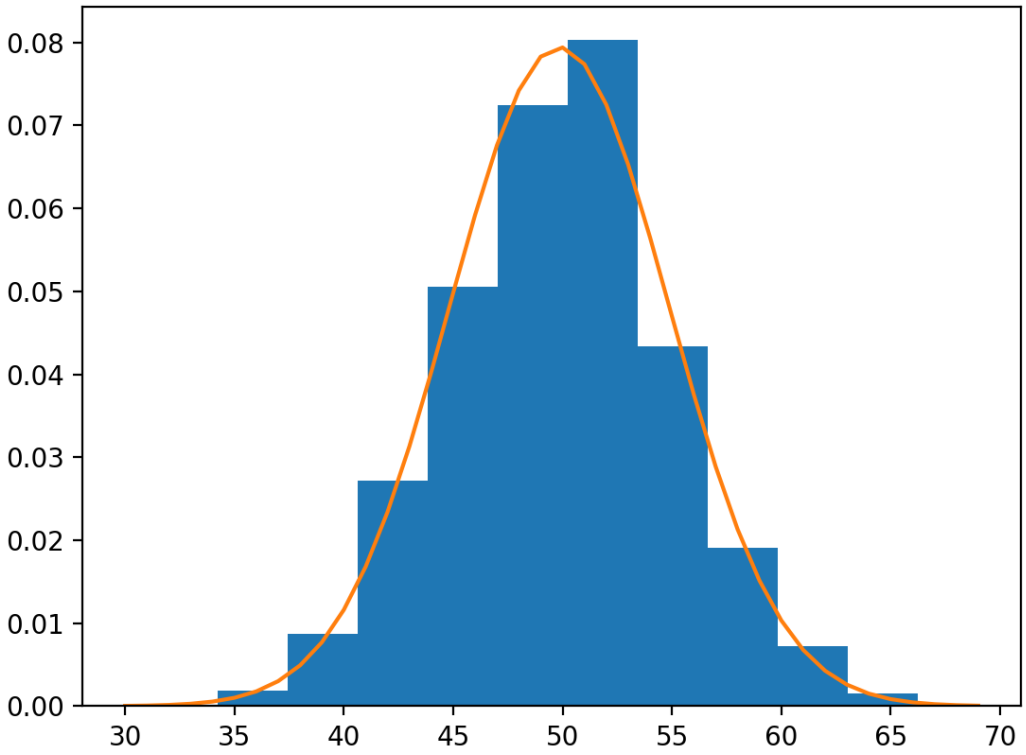

Figure 2.7: Parametric density estimation where histogram of the original data sample is plotted along with the estimated PDF (represented by the line plot)

2.4 Brief overview of the binary classifiers used in this study

2.4.1 KDE-based classifier

Generative classifiers model the joint probability distribution $P(X, Y)$ for a given set of features X and the observed variable Y. The main steps for generative classification are given as follows:

1. Split the data set into train and test.
2. For the training set, split the data by the two-class labels (for binary classification).
3. For each of the split, derive a generative model by fitting a density estimation algorithm like KDE. Thus, for any observation x and label y, calculate the likelihood $P(x|y)$.

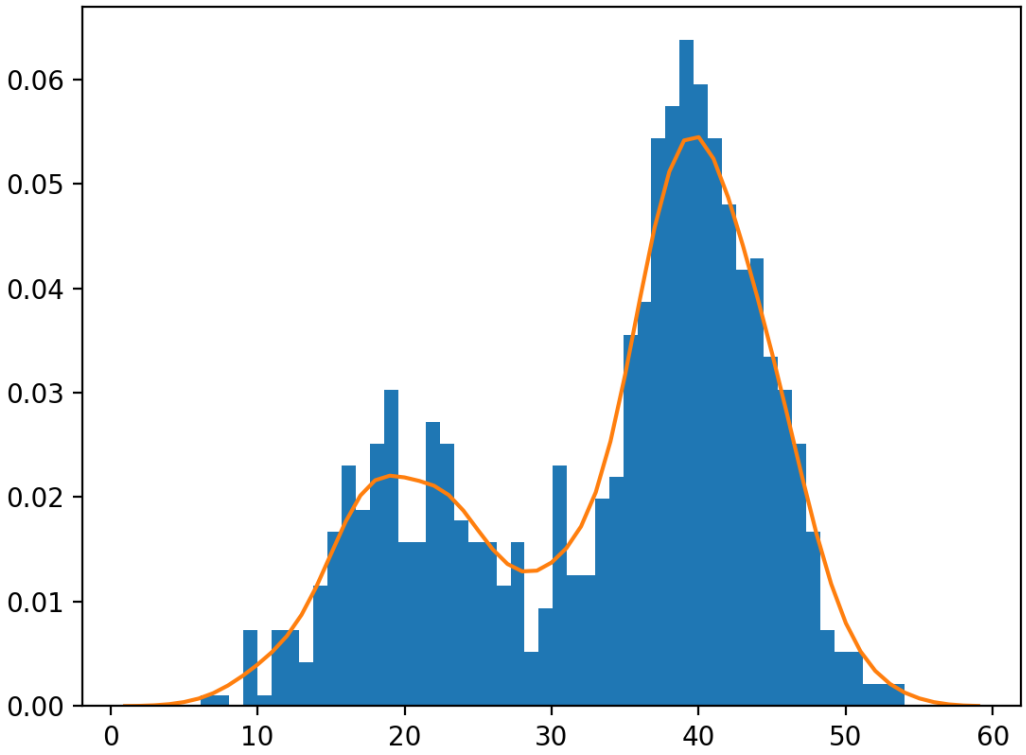

Figure 2.8: Kernel density estimation for a bimodal data sample displaying both the histogram and the estimated density function plot

4. Now calculate the class prior $P(y)$ which denotes the fraction of samples of each class in the training set.

5. Take an unknown point x from the test set and calculate the posterior probability of each class $P(y|x) \propto P(x|y)P(y)$. Now assign that class as the label which maximizes this posterior.

It is also important to note that the Kernel Density Estimator's best parameters, i.e. *bandwidth* and *kernel* are selected through a cross-validation based grid search technique using the training data only.

2.4.2 Balanced random forest classifier

Random Forests (Breiman, 2001) are one of the most powerful ensemble techniques in machine learning. ***Bagging*** or ***B**ootstrap **Agg**regat**ing*** works by combining the predictions from different models fitted on randomly sampled subsets of the training data. Random Forests apply the bagging technique on the feature

space to introduce randomness (Figure 2.9). This random sampling of features, also known as *feature bagging*, from the feature space reduces the overall variance of the model. An extension of the random forest algorithm for imbalanced data sets involves a scenario where a balanced bootstrap sample is provided to each tree of the forest. This is also known as the *Balanced Random Forest Classifier* (Chen *et al.*, 2004).

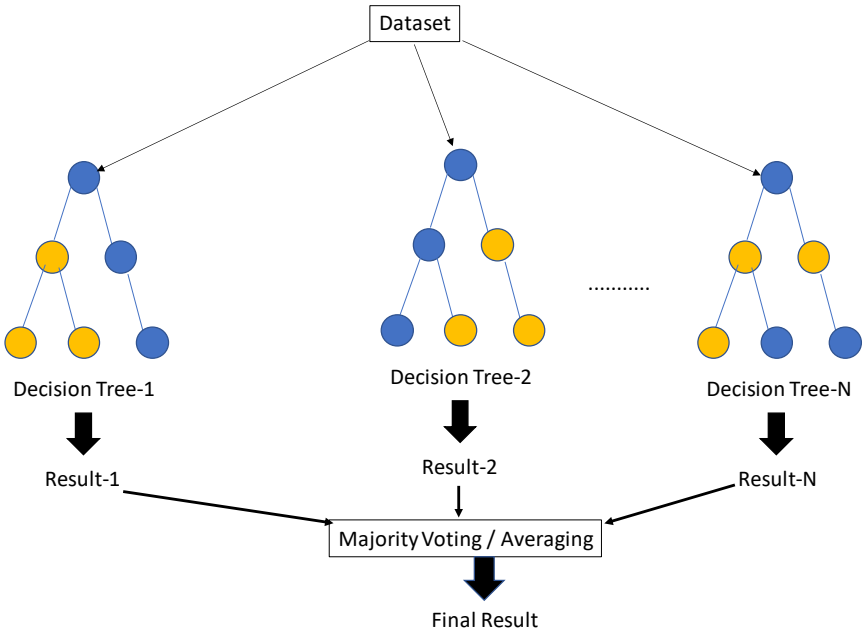

Figure 2.9: A Random forest classifier made up of an ensemble of decision trees

2.4.3 Extremely randomized trees classifier

As mentioned above, random forests use bootstrap subsets of both training data and feature space to introduce randomness. Extremely Randomized Trees (or Extra-Trees) goes a step further by randomly picking the thresholds for splitting the different features in addition to considering random features. Although it introduces more variance at the cost of a higher bias, it does not look for an optimum like “traditional” random forests, and is therefore faster to train.

2.4.4 Support vector machine

An ***SVM*** is a separating hyperplane based classification technique that tries to maximize the distance between the nearest data points (from either class) by solving an optimization function (Figure 2.10). The data samples that are closest to the separating hyperplane or the decision boundary are known as *support vectors*. The distance between the support vectors and the separating hyperplane is also known as the *margin* and increasing the margin results in lower variance and higher generalizability. Parameters such as the slack variable C controls the width of the margin and must be tuned properly to avoid overfitting.

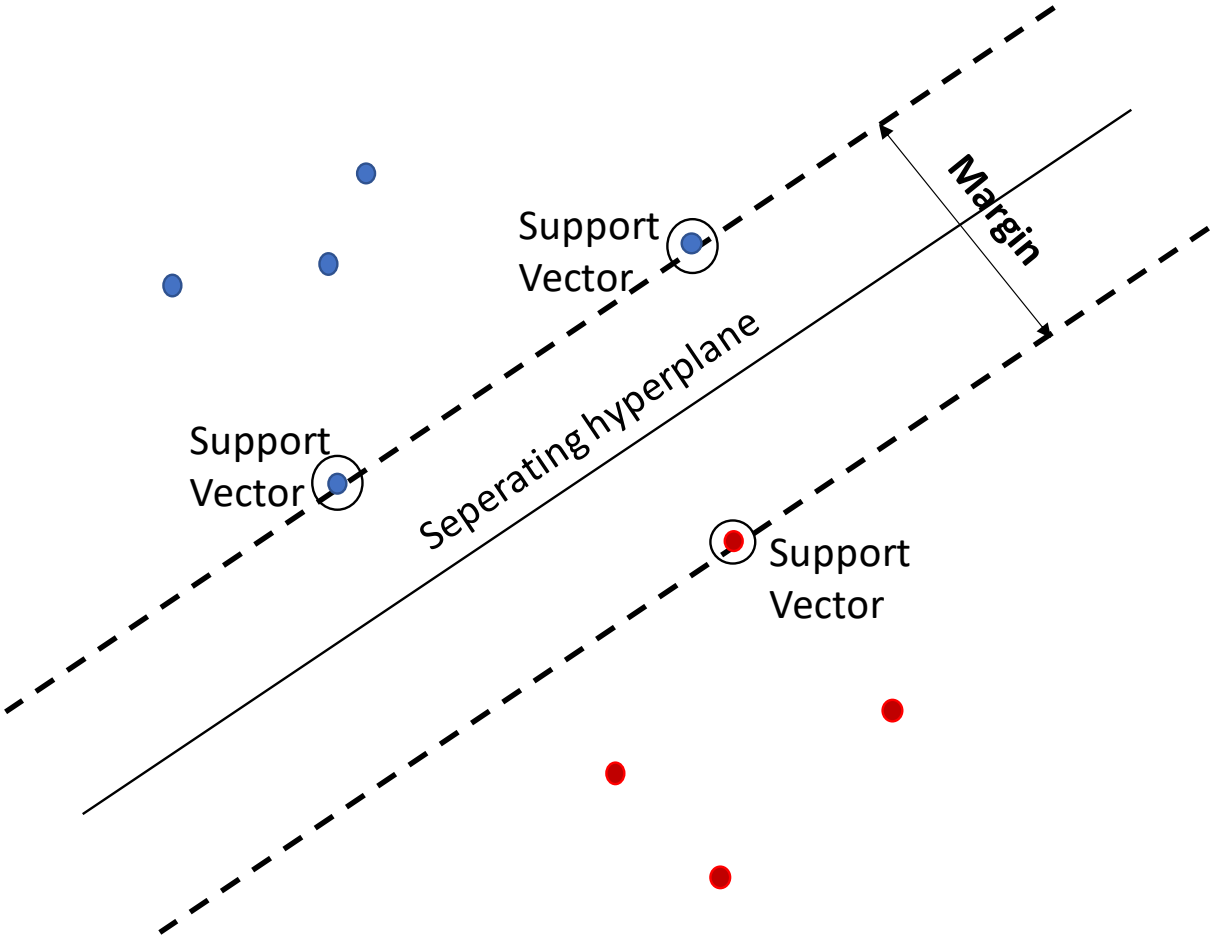

Figure 2.10: A Support vector machine selects the hyperplane that maximizes the margin

2.5 Evaluation metrics

To evaluate the performance of binary classifiers, we first construct a confusion matrix and use that to calculate the performance metrics. *Accuracy* is one such metric that tells us the fraction of correctly classified examples from our data. However, for an imbalanced classification problem, higher accuracy can be due to a model predicting all samples from the test set as belonging to the majority class. Hence, even a classification accuracy of 99% for a highly skewed data set is often misleading. To surmount this problem, we use a variety of other performance metrics that are immune to class imbalance. One such metric is the Mathews Correlation Coefficient (or MCC), which takes into account all the four confusion matrix (Figure 2.11) categories and is usually a preferred way to judge a classifier's performance that has been trained on imbalanced data.

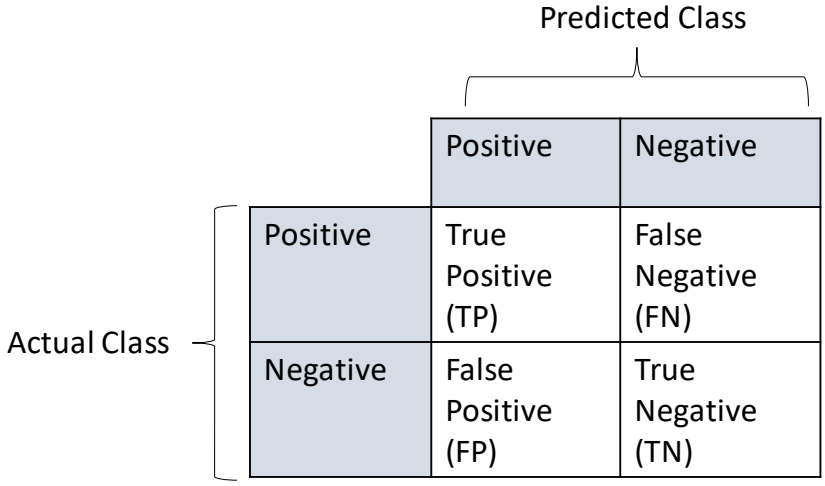

Figure 2.11: Confusion matrix (or error matrix) used to judge the performance of a classifier

Metric	Formula
Sensitivity (or Recall)	$TP/(TP+FN)$
Specificity (or True Negative Rate)	$TN/(TN+FP)$
Positive Predictive Value (or PPV)	$TP/(TP+FP)$
Negative Predictive Value (or NPV)	$TN/(TN+FN)$
Mathews Correlation Coefficient (or MCC)	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$
Composite Score (or CS)	$Sensitivity + Specificity + PPV + NPV$

Table 2.1: Commonly used classification metrics and their formulation

We evaluated the performance of our classifiers by calculating each of the

above mentioned metrics in Table 2.1. In addition, we also reported the *Area under the Curve (AUC)* values which is derived by calculating the area under the plot of True Positive Rate (TPR) vs False Positive Rate (FPR) under different classification thresholds. AUC ranges from 0 to 1 and a higher AUC indicates better classification performance. To compare our models' performances with that of other mutation effect predictors, we used a metric known as *Composite Score* as shown in Table 2.1. The value of this metric ranged between 0 and 4 and provided a robust estimate of the classifier's performance in distinguishing between the two classes of mutations.

2.6 Voting ensembles

An ensemble classifier is a very powerful technique that is used to increase the performance of ML models. A meta-classifier that is obtained by combining multiple ML models often produces a model that is better than each of the individual models (Figure 2.12) (Dietterich (2000)).

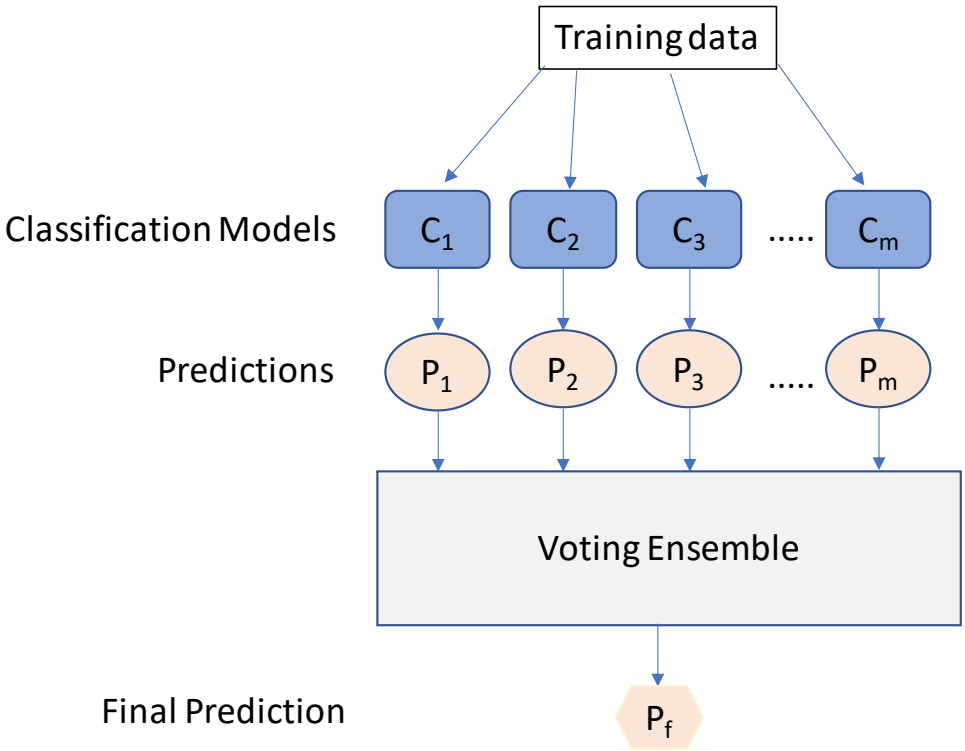

Figure 2.12: Figure depicting the workings of an Ensemble classifier

A voting ensemble forms the simplest case of an ensemble classifier. *Hard*

voting involves predicting the output $\hat{y}$ based on a majority voting rule for each classifier C_i, i.e., $\hat{y} = mode\{C_1(x), C_2(x), ..., C_m(x)\}$. In case a classifier outputs the predicted probabilities in addition to the class labels, we can use *soft voting* to obtain ensemble predictions. $\hat{y} = \text{argmax}_i \sum_{j=1}^{m} w_j p_{ij}$, where w_j is the weight assigned to the j^{th} classifier and p are predicted probabilities. We can weigh the predictions by $\frac{1}{error}$ such that a model with a large error is given a lower weight and vice versa. The resulting ensemble is more diverse because each learner doesn't contribute equally to the final predictions.

2.7 Related work

The effect of the neighborhood context of mutations within a cancer genome on the overall progression of the disease is a relatively new concept and in this section we discuss some of the recently published studies that have explored this area of research. Cancer is caused due to the accumulation of somatic mutations due to multiple mutational processes that are active during the course of the disease (Stratton *et al.*, 2009). But except for a few mutagens such as tobacco in lung cancer, UV light in skin cancer or certain DNA mismatch repair defects (MMR) in Hereditary Non-Polyposis Colon Cancer (HNPCC), the exact cause of their occurrence is unknown. These mutational processes are said to leave an *imprint* or a *signature* on the genome of a cancer patient. Identifying these mutational signatures are of special interest in developing a personalized approach to treat cancer.

Alexandrov *et al.* (2013) analyzed ∼4 million mutations from 7000 cancer types and extracted more than 20 different mutational signatures. To understand the concept behind mutational signatures, let us consider the mutations in the *TP53* gene for skin cancer. If we catalog all *TP53* mutations across patients suffering from skin cancer, we will find that the majority of the mutations in the *TP53* gene are C>T which closely matches the observations from the experiments carried out to study the effect of ultraviolet light on the rate of C>T substitutions. The authors made a similar observation for smoking-related lung cancer abnormalities and C>A substitutions. Basically, the *TP53* gene bears the

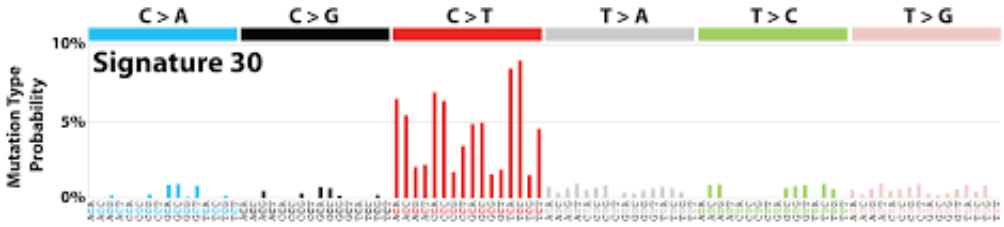

Figure 2.13: A typical mutational signature from the COSMIC database showing the relative abundances of each of the 96 possible mutation types

mark or signature of the changes that first occurred in the human genome due to exposure to specific mutagens.

In this study, the authors first explored the concept of mutational signatures in the context of single base substitutions. There are primarily six classes of base substitutions - C>A, C>T, C>G, T>A, T>C, T>G. Since it is not possible to differentiate on which strand the mutation first occurred, complementary substitutions such as G>T and C>A is indistinguishable from one another. For each of the six classes, information regarding the 5' and 3' adjacent bases was extracted. This resulted in a total of 96 distinct mutation types (A[C>A]A, T[C>A]A, etc.). The total number of times a particular mutation type is observed constitutes the mutational catalog. Statistical methods such as non-negative matrix factorization were then used to extract 21 distinct mutational signatures. There was an overall diverse representation of each of the 96 mutation types. However, some signatures displayed a relatively high specificity of mutations type and nucleotide context. Signature 10, for instance, contained only two out of the 96 possible substitutions. Signature 3, by contrast, exhibited a more or less equal representation of most of the 96 mutation types. The signatures also displayed a tissue-type specificity. Signatures 1A and 1B were present in 25 out of the 30 cancer types included on this study. Both were primarily made up of C>T substitutions centered on an NpCpG mutational context. The phenomenon of localized hypermutation or clusters of C>T/C>G mutations at specific nucleotide contexts is known as *"kataegis"*. Although the underlying mechanisms for most mutational signatures is unknown, Nik-Zainal *et al.* (2012) postulated that the enzymatic activity of the APOBEC enzyme family can be a possible reason for the predominantly C>T and C>G substitutions for Signatures 2 and 13.

Considerable evidence suggests that the neighborhood nucleotide sequences

have an effect on point mutations. Early studies by Cooper *et al.* (1995) based on germline mutations have supported the hypermutability of CpG dinucleotides as the primary reason for C>T substitutions. Subsequent work by Krawczak *et al.* (1998) identified similar patterns for the remaining 11 types of point substitutions. Zhao *et al.* (2002) investigated the substitution patterns and neighboring nucleotide effects of ~ 2 million SNPs publicly available through the National Center for Biotechnology Information (NCBI). This study also enumerated the effects of neighboring nucleotide sequences on various mutational and evolutionary processes.

Krawczak *et al.* (1998) and Zhao *et al.* (2002) both observed a direct association between the influence of neighbors and the number of the flanking nucleotide sequences used for the analysis (or distance from the mutated position). Alexandrov *et al.* (2013), however, focused on the immediate flanking bases to extract the mutational signatures. Recent work by Aggarwala and Voight (2016) showed that a 7-mer sequence context accounted for a high proportion of the variation in the mutation rate across samples. These results indicated that an expanded sequence context could identify higher-order interactions leading to an increase in the predictive power of model. Zhu *et al.* (2017*b*) used log-linear modelling to identify specific sequence motifs affecting point mutations and concluded that major effects of the neighborhood sequences on germline mutations is concentrated around ± 2 bp from the mutation position.

Recent studies have also focused on the effect of neighborhood nucleotide sequences on the deleteriousness of cancer-causing mutations. Dietlein *et al.* (2020) used whole-exome sequencing data from $\sim 11,000$ tumor-normal pairs and extracted 460 driver genes using probabilistic models. They observed that passenger mutations cluster in particular nucleotide contexts and mutations that deviate from these contexts usually provide a signal in favor of driver mutations. Using a 20-bp window size around the cancer-causing mutation, they identified tumor-specific driver genes implicated in known oncogenic pathways. Agajanian *et al.* (2019) integrated traditional machine learning models with deep *convolutional neural networks (CNN)* for prediction of cancer driver mutations. First, a CNN was trained using the neighborhood sequence features extracted using various string-based feature representation methods. Then the DNA-based scores

generated by the CNN were integrated with other genomic features to derive the final driver prediction model.

CHAPTER 3

Neighborhood features enable prediction of driver and passenger mutations from sequenced cancer genomes

In this chapter, we focus on identifying neighborhood-based features for predicting driver mutations using machine learning approaches. First, we discuss the techniques adopted for extracting raw nucleotide sequences from the neighborhood of cancer-causing mutations. Then we use commonly used text vectorization techniques to map the string-based features to a numerical format. After generating the processed feature matrix, we build density models to understand the differences in the distributions between driver and passenger mutations. Finally, we train binary classification models and validate them using independent test sets.

3.1 Methods

3.1.1 Mutation datasets for building and evaluating the models

Our training data consisted of the list of missense mutations whose effects were determined from experimental assays and were compiled in the study conducted by Brown *et al.* (2019). In this study, missense mutations from 58 genes that were pan-cancer based were combined from five different datasets (Table 3.1).

These mutations were presented as amino acid substitutions based on their protein coordinates (eg. F595L, L597Q, etc). Since we were interested in studying the effects of neighboring DNA nucleotide sequences, we had to map them to their corresponding genomic coordinates (gDNA) for further analysis. We

Study	Description
Olivier *et al.* (2002)	Experimentally determined missense mutations based on the functional transactivation activities. "Functional" and "partially functional" mutations were treated as "neutral" and "non functional" mutations were treated as "non neutral"
Martelotto *et al.* (2014)	Curated list of mutations based on experimental evidence collected from literature. Mutations with documented evidence of deleteriousness were classified as "damaging" and those without were considered "neutral"
Starita *et al.* (2015) Mahmood *et al.* (2017)	Experimentally verified BRCA1 mutations and their ability to participate in homology-directed repair
Campbell *et al.* (2017)	Drivers of hypermutation in DNA polymerase epsilon and polymerase delta genes (POLE/POLD1)
Ng *et al.* (2018)	Annotated missense mutations based on their impact on cell viability in Ba/FC and MCF10A models. Whenever the cell viability was higher than the wild-type, the mutation was labelled "activating" or "non-neutral" and when it was similar to the wild-type, it was labelled as "neutral"

Table 3.1: Combination of mutations from five separate studies into a single dataset of missense mutations for training purposes

used the publicly available TransVar web-interface (Zhou *et al.* (2015)) for this purpose. The final training set was made up of 5265 single nucleotide variants (4131 neutral and 1134 non-neutral).

For external validation, we considered somatic mutation data from five different sources. First, we considered a literature curated list of 140 neutral and 849 non-neutral single nucleotide variants that were categorized on the basis of functional evidence published by Martelotto *et al.* (2014) as part of the benchmarking study to rank various mutation effect prediction algorithms.

Second, we used a subset of mutations published by the recently released Cancer Mutation Census (CMC). The CMC (Tate *et al.*, 2019) is a database that integrates all coding somatic mutation data from the COSMIC database in an effort to prioritize variants driving different forms of cancer. It contains functional evidence obtained using both manual curation and computational predictions from multiple sources. For our validation experiments, we chose only single

nucleotide variants that were classified as missense and were derived from the CGC-classified list of tumor suppressor genes and oncogenes. Based on the various evidence criteria set forth by the database, we considered only mutations categorized as tier 1, 2, and 3 for our study. From this list, we further removed all overlapping mutations with our training set and derived a final set of 277 mutations for further analysis.

The Catalog of Validated Oncogenic Mutations from the Cancer Genome Interpreter (Tamborero *et al.*, 2018) database contains a high confidence list of pathogenic alterations compiled from several sources such as the DoCM (Ainscough *et al.*, 2016), ClinVar (Landrum *et al.*, 2016), OncoKB (Chakravarty *et al.*, 2017) and the Cancer Biomarkers Database (Tamborero *et al.*, 2018). Only missense somatic mutations flagged as "cancer" were extracted for our validation experiments. After removing all overlapping mutations with our training set, we obtained a final list of 1628 driver mutations. This constituted our third validation set. The fourth validation dataset consisted of the list of top 50 hotspot mutations reported in the comprehensive study done by Rheinbay *et al.* (2017). In this study, mutation data was accumulated from the Pan-Cancer Analysis of Whole Genomes (PCAWG) consortium and involved analyzing more than 2700 cancer genomes derived from more than 2500 patients. A total of 33 coding missense mutations from five well known cancer genes: *TP53*, *PIK3CA*, *NRAS*, *KRAS*, *IDH1* were extracted from this study. Mao *et al.* (2013) published mutation datasets to compute the ability of their driver prediction tool (CanDrA) to predict rare driver mutations. These datasets were constructed using the following criteria:

- GBM and OVC mutations that were reported in the COSMIC database only once.
- The reported mutations had no other mutations within 3bp of their position and were not part of either the training or test datasets for building the machine learning model (CanDrA).

This final validation set was used to judge our model's ability to predict rare driver mutations based solely on the neighborhood sequences. After removing all overlapping mutations with the training set, we obtained a total of 34 GBM mutations and 38 OVC mutations. A summary of all the mutational datasets used

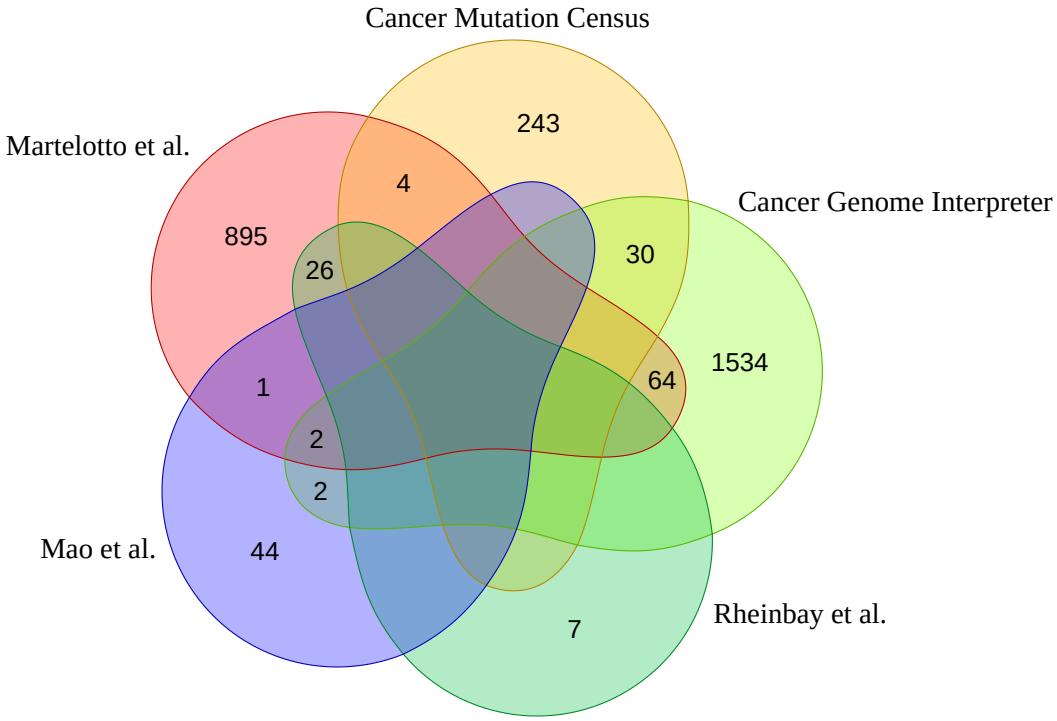

Figure 3.1: Intersection between the five mutation datasets used for validation

in our study is available in Table 3.2. Except the rare driver mutation dataset published by Mao *et al.* (2013), there is considerable overlap between the set of mutations used to train/validate our model and COSMIC (v91) (Futreal *et al.* (2004)). Figure 3.1 shows the intersection between the various mutational datasets used to validate our machine learning model. As evident from this figure, each dataset contributed a considerable amount of unique mutations for testing the generalizability of the final model. All our predictions are derived using the forward strand and were based on the GRCh37 (ENSEMBL release 87) build of the human genome.

3.1.2 Feature extraction

Sequence-based features

We used the raw nucleotide sequences surrounding a mutation as features for our analysis. Each unique mutation was represented as a triplet (Chromosome,

Type	Study / Database Name	Description	Sample size	Overlap with COSMIC ($n > 2$) mutations
Training	Brown *et al.* (2019)	Missense mutations from 58 cancer genes generated from experimental assays	5265 mutations (Non-neutral: 1134, Neutral: 4131)	4334
Validation	Martelotto *et al.* (2014)	A literature curated list of mutations from 15 cancer genes used to benchmark 15 mutation-effect prediction algorithms	989 mutations (Non-neutral: 140, Neutral: 849)	989
Validation	Catalog of Validated Oncogenic Mutations Tamborero *et al.* (2018)	High confidence pathogenic missense variants compiled from several sources mined from the Cancer Genome Interpreter database	1628 non-neutral mutations	1568
Validation	Rheinbay *et al.* (2017)	Recurrent single point driver mutations in the coding region compiled from the Pan-Cancer Analysis of Whole Genomes Consortium	33 non-neutral mutations	33
Validation	Mao *et al.* (2013)	Rare driver mutations from GBM and OVC cancer types	72 mutations (GBM: 34 non-neutral mutations, 38 non-neutral mutations)	0
Validation	Cancer Mutation Census (COSMIC v92)	Tier 1,2 and 3 single nucleotide missense variants derived from the CGC-classified list of TSGs and OGs categorized into different functional classes both through manual curation and computational predictions	277 non-neutral mutations	269

Table 3.2: Summary of datasets used in this study

Position, Type) where "Type" refers to one of the 12 types of point substitution (A>T, A>G, A>C, T>A, T<G, T>C, G>A, G>C, G>T, C>T, C>A, C>G). We then extracted the surrounding raw nucleotide sequences from the reference genome for a given mutation position using the bedtools getfasta (5) command. Window size for a particular mutation is defined as the number of nucleotides upstream and downstream from the mutated position. Hence, considering all possible window sizes between 1 and 10 and including the wild-type nucleotide at the mutated position, we obtained nucleotide strings of length 3, 5, 7, 9, 11, 13, 15, 17, 19, and 21, respectively. We also considered the chromosome number and the type of point substitution as features for our analysis. Now, for particular window size, in order to map the nucleotide strings to a numerical format, we used the following two widely used feature transformation approaches (Figure 3.2):

- **One-hot encoding:** Each neighboring nucleotide was represented as a binary vector of size 4 containing all zero values except the index of that nucleotide, which was marked as 1. Thus "A" was encoded as [1, 0, 0, 0], "G" as [0, 1, 0, 0] and so on. This particular feature representation resulted in a feature space of size 8n +2, where n=1, 2, 3 ... 10. We used the pandas function `get_dummies()` to perform this task.
- **Overlapping *k*-mers:** In this type of feature representation, the neighboring nucleotide string sequences for a given window size were represented as overlapping k-mers of length 2, 3, and 4. For instance, an arbitrary sequence of window size 3 {ATTTGGA}, where 'T' is the wild type base at the mutated position, can be decomposed into overlapping *k*-mers of size 2 {AT, TT, TT, TG, GG, GA}, 3 {ATT, TTT, TTG, TGG, GGA} and 4 {ATTT, TTTG, TTGG, TGGA} respectively. To map these overlapping k-mers to a numerical format, we applied two commonly used encoding techniques known as CountVectorizer and TfidfVectorizer. The CountVectorizer returns a vector encoding whose length is equal to that of the vocabulary (total number of unique k-mers in the data set) and contains an integer count for the number of times a given k-mer has appeared in our dataset. A Term Frequency – Inverse Document Frequency (TF-IDF) vectorizer assigns scores to each k-mer based on
  - how often the given k-mer appears in the dataset
  - how much information the given k-mer provides, i.e., whether it is common or rare in our dataset.

The derivation of both the TFIDF and The Count vectorizer scores were implemented in Python using the `feature_extraction` module from Scikit-learn. The final processed training set used to build the machine learning models

Window size	Number of one-hot encoded features	Number of k-mers possible for a given k-mer size		
		k=2 ($\|\sum\|$=16)	k=3 ($\|\sum\|$=64)	k=4 ($\|\sum\|$=256)
w=1	8	2	1	0
w=2	16	4	3	2
w=3	24	6	5	4
w=4	32	8	7	6
w=5	40	10	9	8
w=6	48	12	11	10
w=7	56	14	13	12
w=8	64	16	15	14
w=9	72	18	17	16
w=10	80	20	19	18

Table 3.3: Number of one-hot encoded features and possible k-mers for a given window size. The size of the vocabulary is given in brackets

was represented as a matrix of size mn, where m = total number of coding point mutations and n = size of the vocabulary. The matrix entries were basically the TF-IDF or the CountVectorizer scores. The number of one-hot encoded features, k-mers, and the size of the vocabulary possible for each window size is shown in Table 3.3.

Descriptive genomic features

In addition to the neighborhood features, a set of 29 features (Table 3.4) previously used to train the cancer-specific driver missense mutation annotation tool, CanDrA (Mao *et al.*, 2013) were extracted from the following three data portals: CHASM's SNVBox (Carter *et al.*, 2009), Mutation Assessor (Reva *et al.*, 2011) and ANNOVAR (Wang *et al.*, 2010). Among them were conservation scores, amino acid substitution features, exon features, and functional impact scores computed by algorithms such as VEST (Carter *et al.*, 2013) and CHASM (Carter *et al.*, 2009). A tiny fraction (0.1%) of the UniProtKB annotations were not available from the SNVBox database for our training data. We used the k-nearest neighbors based imputation technique to substitute the missing features with those of the nearest mutations within the same gene. Our external validation datasets were free from any missing information.

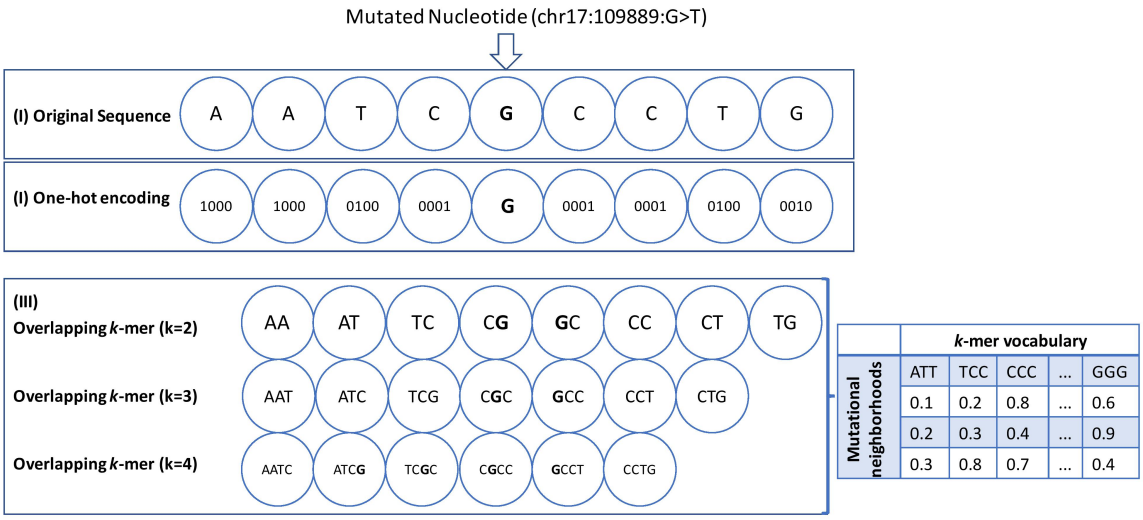

Figure 3.2: A diagram representing the features derived from the neighborhood nucleotide sequences of the point mutations for an arbitrary window size of 4

Table 3.4: List of the descriptive genomic features used to train our machine learning models (* denotes the features that were among the top 50 features used to derive NBDriver)

Features	Source	Significance
UniprotDOM_PostModEnz*	SNVBox	Site in an enzymatic domain responsible for any kind of post-translational modification
ExonSnpDensity*	SNVBox	The number of SNPs in the exon where the mutation is located divided by the length of the exon.
HMMPHC*	SNVBox	Calculated based on the degree of conservation of the residue, the mutation and the most probable amino acid in a match state of a hidden Markov model built with SAM–T2K software.

(continued on next page)

Table 3.4 – continued from previous page

Features	Source	Significance
MGARelEntropy*	SNVBox	Kullback–Leibler divergence calculated for the column of Multiz–46-way alignment (corresponding to the location of the mutation) and that of a background distribution of amino acid residues computed from a large sample of multiple sequence alignments.
HMMRelEntropy*	SNVBox	Kullback–Leibler Divergence calculated for the column of the SAM–T2K multiple sequence alignment (corresponding to the location of the mutation) and that of a background distribution of amino acid residues computed from a large sample of multiple sequence alignments.
Gerp++*	ANNOVAR	Evolutionary conservativeness score using a top to bottom based species distance calculation
AAPAM250	SNVBox	Amino acid substitution score from the PAM250 matrix.
PredBFactorM*	SNVBox	These features consist of the probability that the wild type residue backbone is intermediate.
PredRSAE*	SNVBox	These features consist of the probability of the wild type accessibility residue being exposed.
PredBFactorS*	SNVBox	These features consist of the probability that the wild type residue backbone is stiff.

(continued on next page)

Table 3.4 – continued from previous page

Features	Source	Significance
AABLOSUM*	SNVBox	Amino acid substitution score from the BLOSUM 62 matrix
AAGrantham*	SNVBox	The Grantham distance from reference to mutation amino acid residue.
AAEx*	SNVBox	Amino acid substitution score from the EX matrix.
ExonHapMapSnpDensity	SNVBox	The number of HapMap verified SNPs (dbSNP build 131) in the exon where the mutation is located divided by the length of the exon.
HMMEntropy*	SNVBox	The Shannon entropy calculated for the column of the SAM–T2K multiple sequence alignment, corresponding to the location of the mutation.
MGAEntropy*	SNVBox	The Shannon entropy calculated for the column of the Multiz–46-way alignment, corresponding to the location of the mutation.
UniprotREGIONS*	SNVBox	Region of interest in the protein sequence, which are defined experimentally, could be related to protein-protein interaction and biological process regulation etc.

(continued on next page)

Table 3.4 – continued from previous page

Features	Source	Significance
ExonConservation*	SNVBox	The conservation score for the entire exon calculated from a 46-species phylogenetic alignment using the UCSC Genome Browser (hg19). Scores are given for windows of nucleotides. We retrieve the scores for each region that overlaps the exon in which the base substitution occurred and calculated a weighted average of the conservation scores where the weight is the number of bases with a particular score.
PredRSAB*	SNVBox	These features consist of the probability of the wild type accessibility residue being buried.
PredStabilityL*	SNVBox	These features consist of the probability that the wild stability type residue contributes to overall protein stability in a manner that is highly destabilizing, Stability estimates for the neural network training data were calculated using the FoldX force field.
PredStabilityH*	SNVBox	These features consist of the probability that the wild stability type residue contributes to overall protein stability in a manner that is highly stabilizing, Stability estimates for the neural network training data were calculated using the FoldX force field.

(continued on next page)

Table 3.4 – continued from previous page

Features	Source	Significance
AAMJ*	SNVBox	Amino acid substitution score from the Miyazawa-Jernigan contact energy matrix
Variant Conservation Score	Mutation Assessor	Entropy-based evolutionary conservativeness score.
Variant Specificity Score	Mutation Assessor	Species subgrouping based evolutionary conservativeness score.
Functional Impact Score	Mutation Assessor	Additive combination of variant conservation score and variant specificity score.
VEST*	SNVBox	VEST pathogenicity score for missense variants
Cancer Driver Score*	SNVBox	$1 - CHASM$ cancer driver score. A quantity closer to 1 means that the mutation is more likely a cancer driver.
Chromosome*	Genomic Feature	Denotes location of the mutation in the genome
Substitution Type*	Genomic Feature	Denotes the type of base substitution

Steps taken to extract descriptive genomic features

- **SNVBox** (Wong *et al.*, 2011) is a MySQL database containing pre-computed structural, amino acid, functional, sequence alignment features for all codons in the human genome. The CRAVAT or the Cancer-Related Analysis of VAriants Toolkit (Masica *et al.*, 2017; VanderPlas, 2016) web server (v.4.3) was used to extract these features for training purposes. The default analysis programs, VEST 3.0, CHASM 3.0 and Snvget were selected. The following input format was used:

UID	Chromosome	Position	Strand	Reference	Alternate
TR1	chr14	233343	+	A	T
TR2	chr3	56776	+	T	C

All the features tagged as SNVBox in Table 3.4, were downloaded and stored as a seperate .csv file.

- **ANNOVAR** (Wang *et al.*, 2010) is an efficient software tool to perform functional annotation of a group of variants. Only one feature, namely, the GERP++ score was extracted from ANNOVAR. The following input format was used:

Chromosome	Start	End	Reference	Alternate	ID
chr14	233343	233343	A	T	TR1
chr3	56776	56776	T	C	TR2

The GERP++ scores were stored as a separate .csv file.

- **Mutation Assessor** (Reva *et al.*, 2011) is a functional impact prediction tool for missense mutations. We annotated our list of variants using the web API. To submit a new variant, the following url was called:

  `http://mutationassessor.org/r3/?cm=var&var=<substitution>`

  The "genomic substitution" field contained the genomic coordinates in the following format "Chromosome,Position,Reference,Alternate". Thus, an example query to the Mutation Assessor server looked like:

  `http://mutationassessor.org/r3/?cm=var&var=7,55211080,G,A`

  Three functional scores, namely, Functional Impact Score, Variant Specificity Score and Variant Conservation Score were extracted from Mutation Assessor. Successive API calls were made using the `requests` module in Python.

3.2 Density estimation

A kernel density estimator (or KDE) takes an n-dimensional dataset as an input and outputs an estimate of the underlying n-dimensional probability distribution. A Gaussian KDE tries to center one gaussian component per data point, essentially resulting in a non-parametric estimation of the density. One of the hyperparameters for a kernel density estimator is the bandwidth, which controls the kernel's size at each data point, thereby affecting the "smoothness." We estimated the underlying probability distributions for the neutral and non-neutral neighborhoods using a Gaussian kernel density estimator.

We randomly selected an equal number (n) of neutral and non-neutral mutations from our training data with replacement for a single run of the kernel density estimation algorithm and particular window size. We tuned the bandwidth hyperparameter for each class of mutations using a 5-fold cross-validation approach and used the best set of hyper parameters to derive the kernel density estimates. Finally, we used the Jensen–Shannon (JS) distance metric to

calculate the similarity between the two class-wise density estimates. The JS distance between two probability distributions is based on the Kullback–Leibler (KL) divergence, but unlike KL divergence, it is bounded and symmetric. For two probability vectors, p and q, it is given by,

$$JS = \frac{1}{2}\sqrt{(D(p||m) + D(q||m)}, \tag{3.1}$$

where $m = \frac{1}{2}(p + q)$ and D is the KL divergence.

The significance of the estimated distances between the probability estimates was calculated using a randomized bootstrapping approach. Specifically, we randomly sampled with replacement twice the number ($2n$) of mutations from the same training set irrespective of the labels. We then split the dataset in half, randomly assigning each half to neutral and non-neutral mutations, respectively. This was followed by a similar process of tuning the hyperparameters, deriving the density estimates, and calculating the distances between them. We had a total of seven different neighborhood-based feature representations: One-hot encoding, Count vectorizer (k-mer sizes of 2, 3, and 4), and TF-IDF vectorizer (k-mer sizes of 2, 3, and 4). Each KDE estimation experiment was repeated 30 times for all window sizes between 1 and 10 and all seven feature representations.

Next, the best median JS distance estimate from the original experiments was reported for the given window size. The percentage of runs of the randomized experiments for which the estimated distance was greater than this estimate was considered the *p*-value. A schematic workflow of the entire process for a single run of the kernel density estimation experiment is shown in Figure 3.3. Figure 3.3(A) depicts the original experiment to calculate the distance between the driver and passenger neighborhoods' kernel density estimates. An equal number of driver and passenger mutations were sampled with replacement, followed by tuning the "bandwidth" parameter and subsequent estimation of the densities. The reported metric is the JS distance, and it is used to quantify how "distinguishable" two probability distributions are from each other. It is bounded between 0 and 1, where 0 represents the case where the two probability distributions are equal and vice versa.

Figure 3.3(B) shows the bootstrapping approach adopted to compute the

significance of the density estimates calculated in Figure 3.3(A). This involved random sampling of twice the number of the driver or passenger mutations from Figure 3.3(A) irrespective of the labels, followed by randomly splitting the data into driver and passenger labels and then repeating the entire tuning and density estimation process similar to Figure 3.3(A). Both Figure 3.3(A) and Figure 3.3(B) were repeated 30 times for all possible window sizes between 1 and 10. The significance of the difference between the medians of the original and the bootstrapped JS distances was then reported.

The `KernelDensity()` from the scikit-learn "neighbors" module was used to derive the density estimates, and `jensenshannon()` from the scipy "spatial.distance" submodule was used to calculate the distance metric.

3.3 Classification models

To build our binary classification models, we implemented three classifiers: the Random Forest classifier, the Extra Trees classifier (also known as Extreme Random Forests), and the generative KDE classifier (Section 2.4.1). Both the tree-based classifiers are discriminative and composed of a large collection of decision trees where the final output is derived by combining every tree's predictions using a majority voting scheme. The main difference between the two lies in the selection of splits or cut points in order to split the individual nodes. Random Forest chooses an optimal split for each feature under consideration, whereas Extra Trees chooses it randomly. All the classification models were written using the predefined functions available in the scikit-learn (v. 0.22) module (Pedregosa *et al.*, 2011).

3.4 Model selection and tuning

3.4.1 Repeated cross-validation experiments

Owing to the relatively smaller sample size (5265 mutations) of the training set of mutations, we adopted a repeated 10-fold cross-validation approach to build-

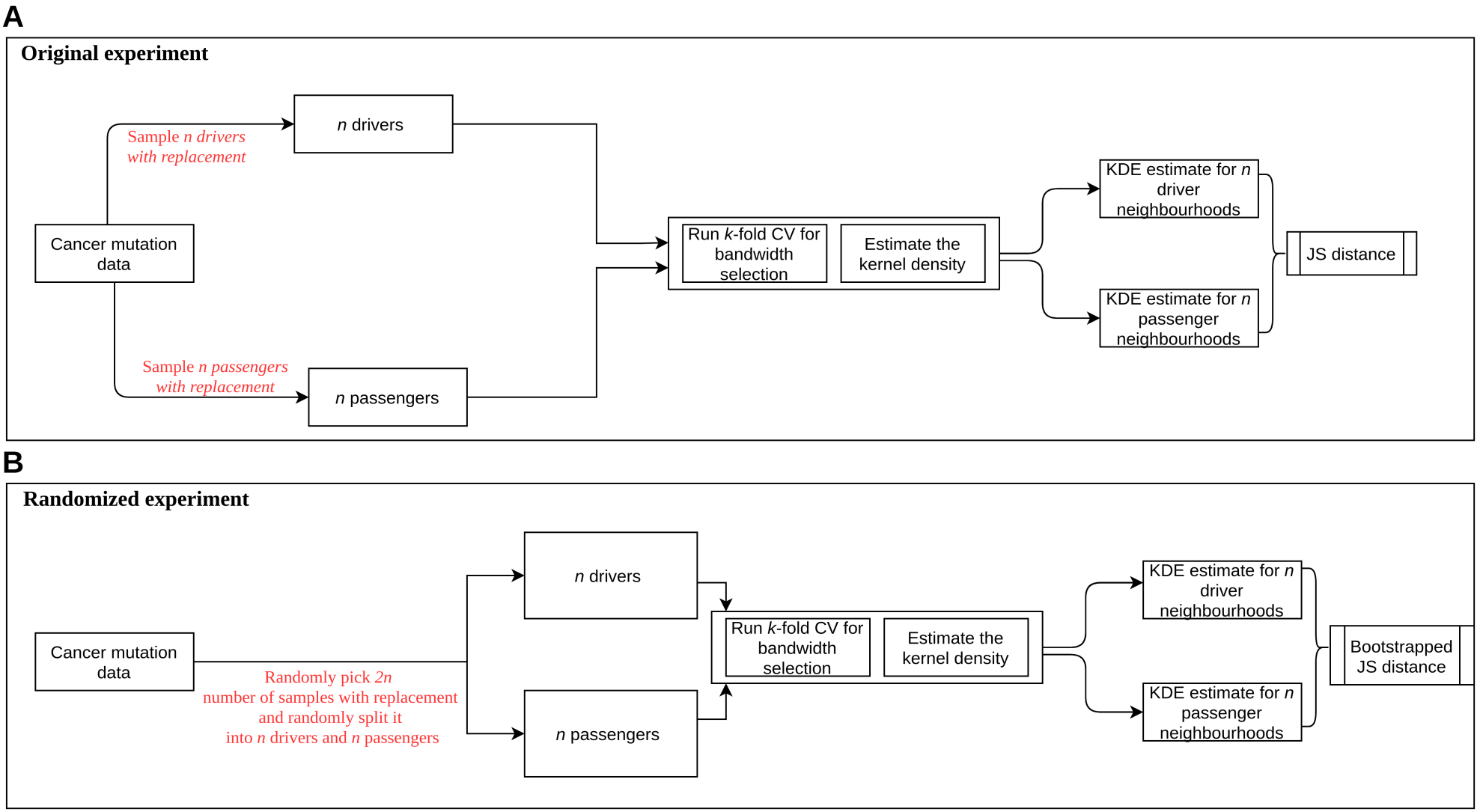

Figure 3.3: Workflow depicting one run of the kernel density estimation experiment.

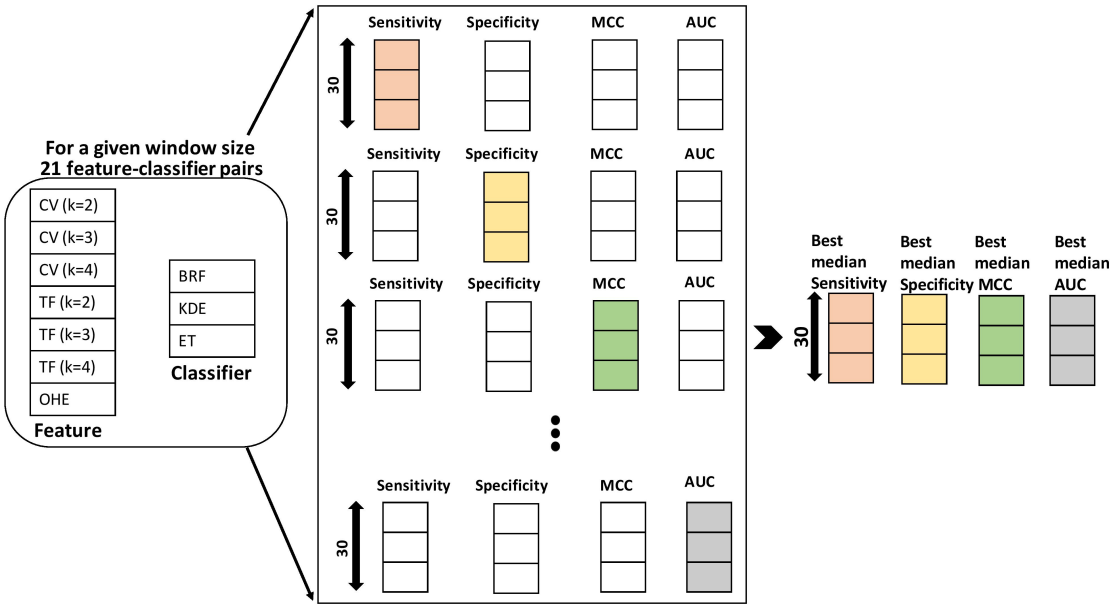

Figure 3.4: Diagram depicting the different classification models constructed as part of the repeated cross-validation experiments.

ing our model. First, the dataset was split into ten equal subsets in a stratified fashion. Nine of the ten subsets were combined into one training set. In each training phase, we performed feature selection using the extra trees classifier, cross-validated grid search-based parameter tuning, training the classifier using the best parameters, and obtaining the corresponding prediction scores on the hold-out test set. This process was repeated three times, thereby resulting in 30 unique train and test splits, and the median performance metrics were reported. For a given window size, we experimented with a total of seven feature representations (One-hot encoding, Count vectorizer (k-mer size=2, 3 and 4), TF-IDF vectorizer (k-mer size=2, 3 and 4) and three binary classifiers (Random Forests, Extra Trees and Kernel Density Estimation). So overall, we had 21 distinct feature–classifier pairs (Figure 3.4). For each such pair, we obtained 30 values each of sensitivity, specificity, AUC, and MCC. We reported the results from that pair, which had the best overall median value.

3.4.2 Derivation of the binary classification model to distinguish between driver and passenger mutations

To derive the final machine learning model, all overlapping mutations between the training set (Brown *et al.*, 2019) and the validation set (Martelotto *et al.*, 2014) were discarded, and the classifier was retrained on the reduced train set (4549 mutations: 544 drivers and 4005 passengers). The set of 989 mutations published by Martelotto *et al.* (2014) formed our independent test set. Due to the inherent imbalance in the dataset, we implemented an undersampling technique known as Repeated Edited Nearest Neighbors (Wilson (1972)) to downsize the majority class and consequently obtain a balanced dataset for subsequent training. Predictions were obtained using two separate feature sets: 1) only neighborhood features based on the raw nucleotide sequences (or the neighborhood-only-model) and 2) neighborhood features plus the descriptive genomic features (or **NBDriver**). In addition to Random Forests, Extra trees, and the KDE classifier, we also experimented with a fourth classifier: a linear kernel SVM to obtain these predictions. Various combinations of these classifiers were implemented as ensemble models using the `VotingClassifier()` of the ensemble module in scikit-learn.

3.4.3 Feature selection

We adopted an impurity-based feature selection technique for feature selection using the extra trees classifier to derive a ranked list of the top predictive features for our analysis. For the repeated cross-validation experiments, the features that were within the top 30 percentile of the most important features were selected and subsequently used to train our models. However, for deriving NBDriver, we built several classification models based on the top n (n=20, 30, 40, 50, 60) features and chose the one that gave the best overall classification performance. The TF-IDF and the Countvectorizer scores were used as features for our analysis and were implemented using the `feature_extraction` module in scikit-learn. In both cases, a new vocabulary dictionary of all the *k*-mers was learned from the training data using the `fit_transform()` and the corre-

sponding term-document matrix was returned. Using the same vocabulary, the scores of the *k*-mers from the test data were obtained using the `transform()` and were subsequently used in our analysis.

3.4.4 Hyperparameter tuning and classifier threshold selection

Hyperparameter tuning was done using a cross-validation based grid search technique over a parameter grid. To perform the cross-validation based tuning procedure we used the `GridSearchCV()` from the `model_selection` module in scikit-learn. To further fine-tune the classifiers, we experimented with various classification thresholds from 0 to 1 with step sizes 0.001 and chose the one that gave the best AUROC. For an imbalanced classification problem, using the default threshold of 0.5 is not a viable option and often results in incorrect prediction of the minority class examples.

3.4.5 Comparison with other pan-cancer mutation effect predictors

Similar to the benchmarking study conducted by Martelotto *et al.* (2014), we compared the generated binary classifiers with nine pan-cancer mutation effect prediction tools: Mutation Taster (Schwarz *et al.*, 2010), FATHMM (-cancer) (Shihab *et al.*, 2013), Condel (Gonzalez-Perez *et al.*, 2012), FATHMM (missense) (Shihab *et al.*, 2013), PROVEAN (v1.1.3) (Choi *et al.*, 2012), SIFT (Ensemble 66) (Sim *et al.*, 2012), Polyphen2 (Adzhubei *et al.*, 2010), Mutation Assessor (Reva *et al.*, 2011) and VEST (Carter *et al.*, 2009) using the set of 989 literature-curated mutations. For each of these predictors, we used the prediction labels based on predefined score cutoffs published as part of this study. We added two new prediction algorithms (CHASMplus (pan-cancer) (Tokheim and Karchin, 2019) and CanDrA+ (Cancer-in general) (Mao *et al.*, 2013) to the list, and the score cutoffs were decided in the following manner: For CHASMplus, due to the absence of a default threshold, all possible thresholds between 0 and 1 with step

sizes of 0.01 were tested, and the one that gave the highest composite score was reported. All mutations with predicted scores greater than this optimal threshold were labeled as drivers and vice versa. For CanDrA+, we used the default prediction categories. Predictions for CHASMplus and CanDrA+ were obtained from the OpenCRAVAT web server (Pagel *et al.*, 2020) and executable package published by Mao *et al.* (2013) respectively. Different mutation effect predictors were combined using the majority voting rule to obtain better predictive power, and ensemble models were created. While comparing two algorithms, in order to derive the significance of the difference between any two classification metrics, we adopted the same strategy as Martelotto *et al.* (2014). Briefly, we derived the 95% CI for each of these classification metrics by repeated sampling with replacement with 1000 iterations. If the generated CI's touch or there was no overlap, the difference was considered significant ($P < 0.05$) based on the results of the analysis done by Ng *et al.* (2018).

3.5 Results

3.5.1 Distributional differences between driver and passenger neighborhoods

We estimated the underlying probability distributions of the driver and passenger neighborhood sequences using kernel density estimation. We computed the Jensen-Shannon distance metric to understand how "distinguishable" they are from one another. The JS metric is bounded between 0 and 1, with 0 signifying perfectly similar distributions and vice versa. For the Brown *et al.* (2019) dataset, the maximum median JS distance between neutral and non-neutral neighborhood distributions, calculated across 30 runs of bootstrapping experiments, was 0.275 (for a window size of 2). The minimum was 0.211 (for window sizes 7-10). Except for window size 1, all other window sizes had a significant JS distance value ($P < 0.05$). The complete list of KDE results obtained using the training set of 5265 mutations is attached in Table 3.5. Figure 3.5 shows the variation in the JS distances between the class-wise density estimates.

Window Size	Feature Type	Median JS distance (original)	Median JS distance (randomized)	*p*-value
1	TF (k=2)	0.345	0.34	NS
2	OHE	0.275	0.221	< 0.05
3	CV (k=2)	0.219	0.170	< 0.05
4	TF (k=3)	0.214	0.167	< 0.05
5	CV (k=3)	0.211	0.166	< 0.05
6	TF (k=4)	0.210	0.166	< 0.05
7	CV (k=2)	0.211	0.165	< 0.05
8	TF (k=3)	0.211	0.164	< 0.05
9	TF (k=3)	0.211	0.166	< 0.05
10	TF (k=4)	0.211	0.166	< 0.05

Table 3.5: KDE analysis: Median JS distances for both the original and randomized experiments for different window sizes (OHE=One-hot encoding; TF=TF-IDF vectorizer; CV=Count vectorizer)

Two types of boxplots, one for the original and another for the randomized experiments, have been shown here along with the p-values, which approximates the probability that the original median distance can be obtained by chance. Except for window 1, all other window sizes had a significant (** $P < 0.05$) difference between the original and the randomized JS distances.

3.5.2 Classification results

Repeated cross-validation using neighborhood features generated robust classification models

The best median sensitivity of 0.938 was obtained using features derived from a Count vectorizer and subsequent training using a random forest classifier for window sizes 1, 5, 6, and 9, respectively. However, the best median specificity of 0.807, AUC of 0.832, and MCC of 0.584 were obtained using a TF-IDF based feature representation trained using a KDE classifier for a window size of 10. The variation in the sensitivity, specificity, AUC, and MCC with different window sizes obtained during the repeated cross-validation experiments using the initial training set of 5265 mutations is shown in Figure 3.6, Figure 3.7, Figure 3.8 and Figure 3.9 respectively. For each window size, feature representations among CV (CountVectorizer), TF (TF-IDF vectorizer), and OHE (One-hot encoding) that gave the best performances in terms of I) Sensitivity II) Speci-

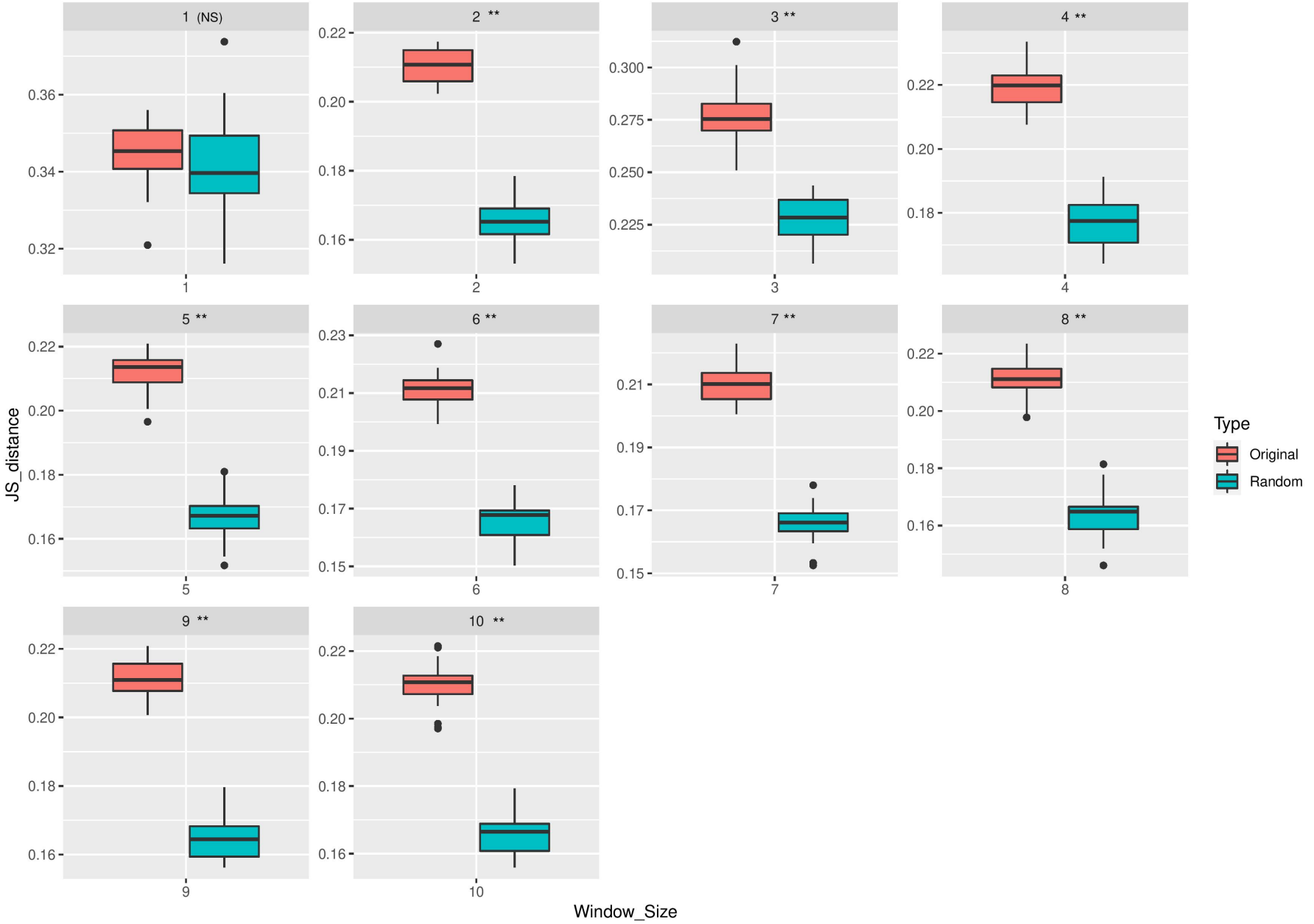

Figure 3.5: Variation in JS distances between the class-wise estimated densities for every window size between 1 and 10.

ficity III) AUC and (IV) MCC is displayed. In the case of specificity, AUC and MCC, the addition of more nucleotides (or increase in the window size) resulted mostly in a significant increase ($P < 0.05$; Wilcoxon signed-rank test) in the corresponding metric. However, for sensitivity, a significant increase was observed only when the window size was increased from 4 to 9 and 7 to 9, respectively. The repeated cross-validation results and the statistical tests to determine the increase in the performance metrics with the increase in window size are shown in Appendix A.1 and Appendix A.2 respectively.

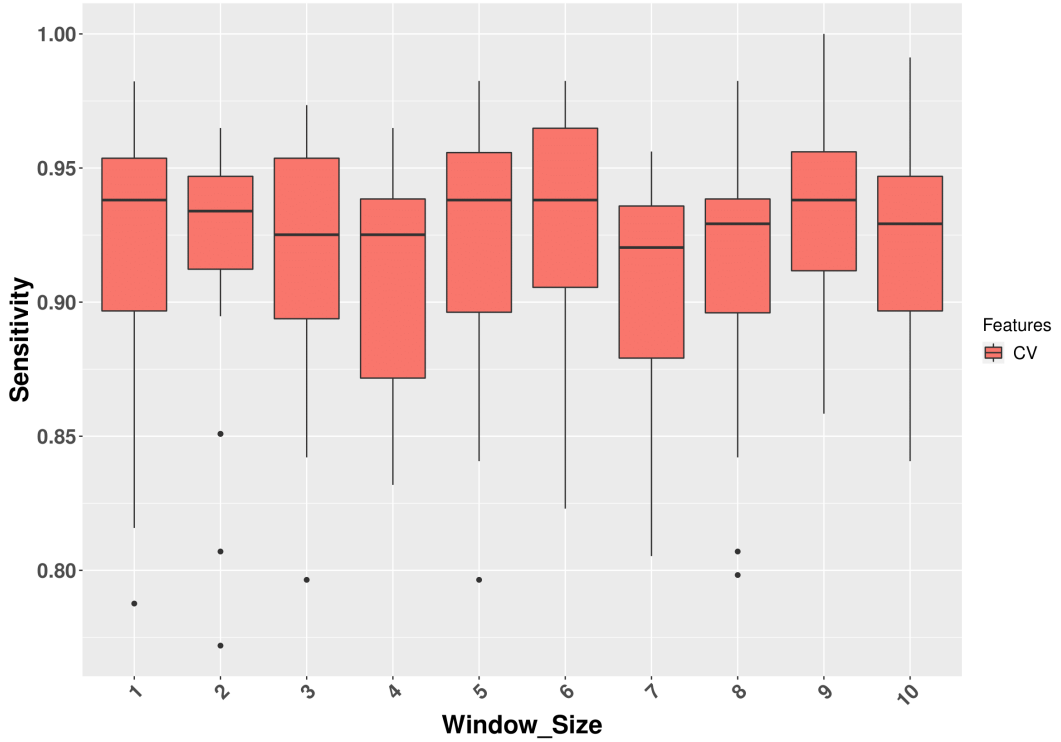

Figure 3.6: Variation in the sensitivity (for the neighborhood-only-model) with different window sizes obtained during the repeated cross-validation experiments using the initial training set of 5265 mutations.

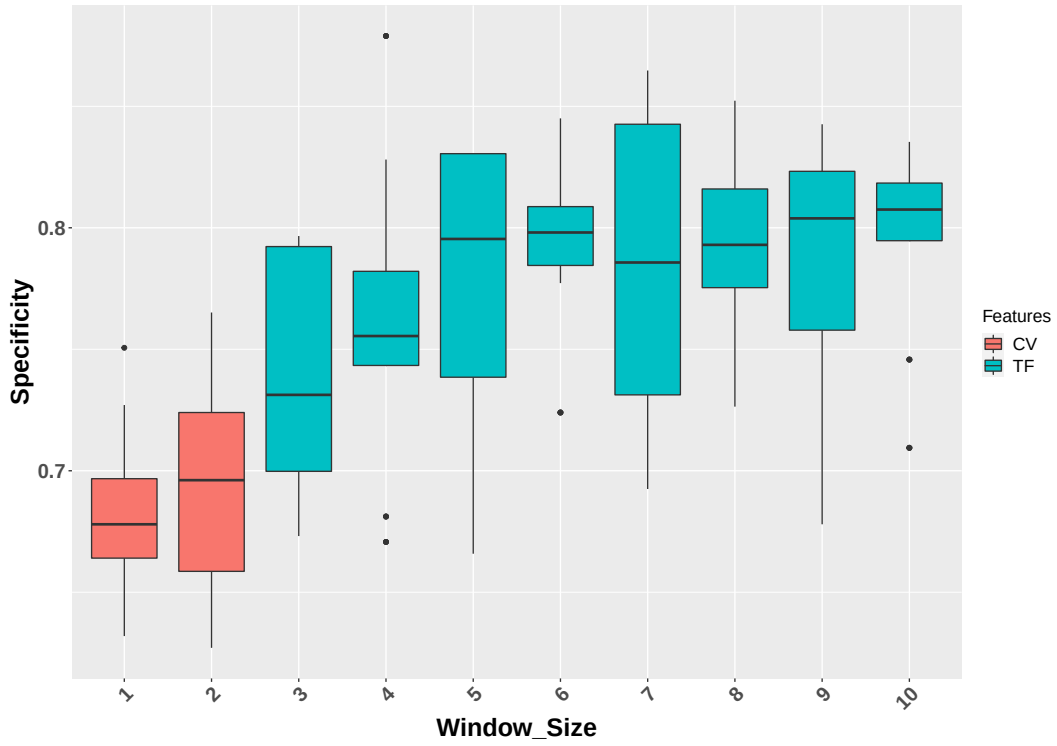

Figure 3.7: Variation in the specificity (for the neighborhood-only-model) with different window sizes obtained during the repeated cross-validation experiments using the initial training set of 5265 mutations.

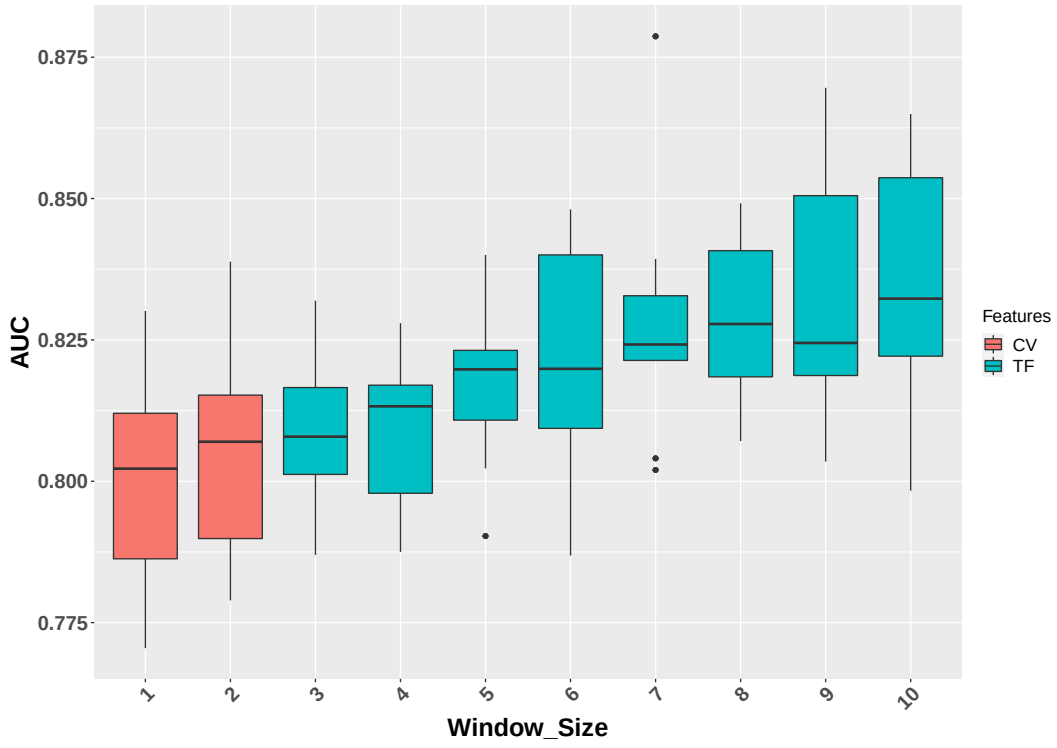

Figure 3.8: Variation in the AUC (for the neighborhood-only-model) with different window sizes obtained during the repeated cross-validation experiments using the initial training set of 5265 mutations.

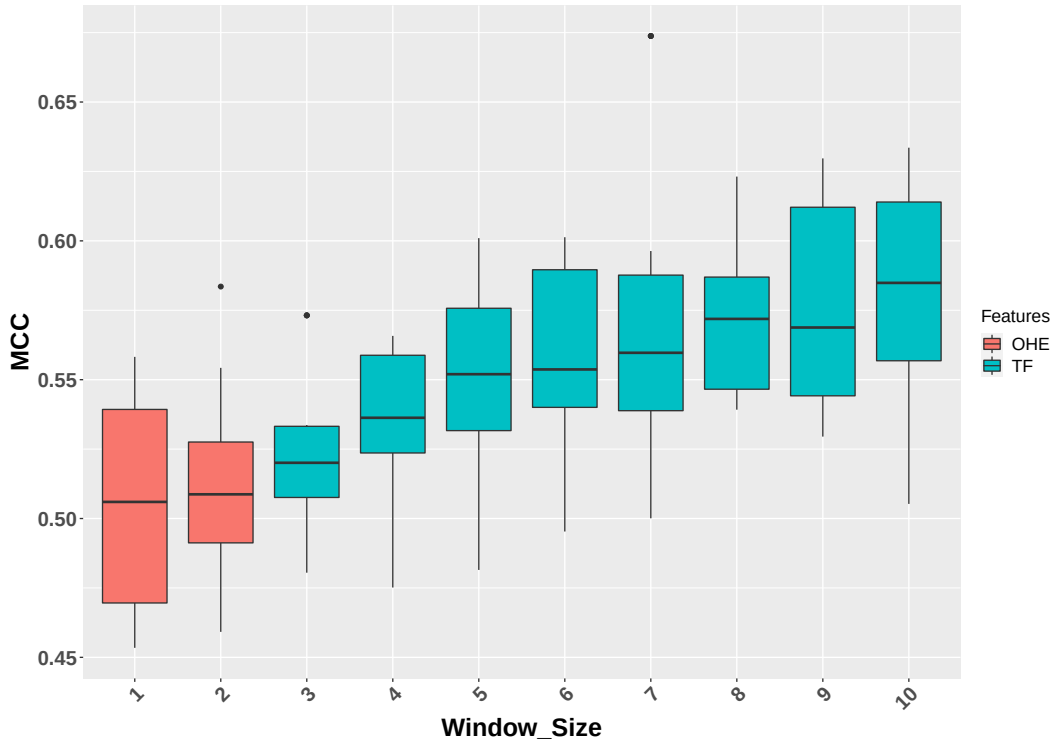

Figure 3.9: Variation in the MCC (for the neighborhood-only-model) with different window sizes obtained during the repeated cross-validation experiments using the initial training set of 5265 mutations.

External Validation using literature-curated list of mutations

Using only the neighborhood nucleotide sequences as features, a composite score of 2.7954 (Table 3.6) on the independent test set (Martelotto *et al.*, 2014), was obtained using an Extra Trees classifier. This neighborhood-only model was trained on features extracted using the Count vectorizer technique on a window size of 10. NBDriver was trained by combining the neighborhood features and the descriptive genomic features. Out of the various classifiers implemented, an ensemble model consisting of a linear kernel SVM and a KDE classifier gave the best results. Compared to the neighborhood-only model, there was a significant increase ($P < 0.05$) in accuracy (=0.891), sensitivity (=0.93), NPV (=0.608), Composite Score (=3.123) and MCC (=0.561). However, this was accompanied by a significant ($P < 0.05$) drop in specificity (=0.643). There was no significant change in PPV though. A ranked list of the 50 features used to train NBDriver is shown in Appendix A.3. Out of those 50 features, 26 were neighborhood-based features or the TF-IDF scores of the overlapping 4-mers extracted from a window

size of 10. The plot displaying the variation in the AUROC with various classification thresholds is shown in Figure 3.10. The best results were obtained using a threshold of 0.119. Consequently, all mutations with the prediction scores above this threshold were classified as drivers and vice versa. Overall, on this benchmarking dataset, NBDriver ranked fourth in terms of the composite score, fifth in terms of specificity and second in terms of NPV, PPV, Sensitivity and Accuracy. By contrast, although neighborhood-only-model was the top-ranking tool in terms of Specificity and PPV, it didn't perform well in terms of the other metrics. Owing to the superior performance of NBDriver, all subsequent external validation was performed using this model only.

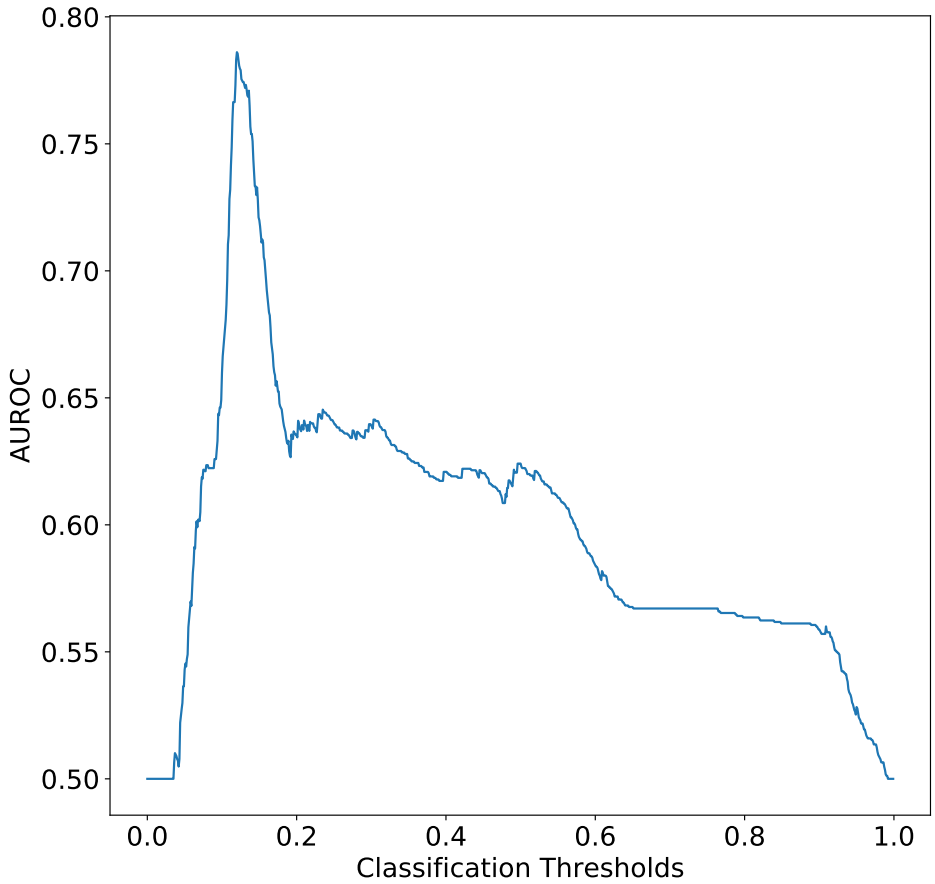

Figure 3.10: Plot showing the variation in AUROC with the different classification thresholds obtained while deriving NBDriver.

Voting ensemble of prediction algorithms

We also assessed the effect of combining multiple top-ranked single predictors into an ensemble model. The contribution of NBDriver to the overall ensemble was evaluated by obtaining predictions with and without the tool. The top performing ensemble consisting of NBDriver, CHASMplus, FATHMM (cancer), Mutation Taster and Condel resulted in a composite score of 3.504, accuracy of 0.942 and an NPV of 0.88, significantly higher ($P < 0.05$) than every single predictor evaluated in the study. Removing NBDriver from the ensemble resulted in a significant decrease ($P < 0.05$) in the composite score, NPV, MCC, Accuracy and Sensitivity. However, it was accompanied by a significant increase in specificity and no significant PPV change for the reduced ensemble (Table 3.6). Another ensemble model consisting of NBDriver, Mutation Taster and Condel gave similar results (Composite score=3.504) as the previous one. Though there was no significant difference in terms of MCC, Composite Score, PPV, Sensitivity and Accuracy, there was a significant increase in the NPV and a significant decrease in the specificity for this ensemble (Table 3.7). A complete set of all the different combinations of the single predictors evaluated in this study is present in Appendix A.4.

Evaluation using mutations derived from the Cancer Mutation Census

Based on the various evidence criteria set forth by the Cancer Mutation Census database, a particular mutation can be classified into tier 1, 2 or 3 with tier ~ 1 mutations having the highest level of evidence of being a driver and so on. From the list of missense mutations in the CMC not present in our training data, NBDriver could accurately predict all 19 tier 1, 25 out of 28 tier 2 and 179 out of 230 tier 3 mutations, achieving an overall accuracy of 81%. On the other hand, the ensemble model consisting of NBDriver, Condel and Mutation Taster could accurately predict all 19 tier 1, 27 out of 28 tier 2 and 214 out of 230 tier 3 mutations achieving an overall accuracy of 94%.

Algorithm	**Accuracy**	**Sensitivity**	**Specificity**	**PPV**	**NPV**	**Composite Score**	**MCC**
Mutation Taster	0.8857	0.9081	0.75	0.9566	0.5738	3.1885	0.590
FATHMM (Cancer)	0.91	0.9788	0.4929	0.9213	0.7931	3.1861	0.580
CHASMplus (pancancer)	0.85	0.852	0.85	0.972	0.486	3.16	0.570
NBDriver	**0.891**	**0.931**	**0.643**	**0.941**	**0.608**	**3.123**	**0.561**
Neighborhood-only model	0.85	0.629	0.907	0.9744	0.285	2.7954	0.370
Condel	0.8584	0.9258	0.45	0.9108	0.5	2.7866	0.392
FATHMM (missense)	0.8251	0.8775	0.5071	0.9152	0.4057	2.7055	0.351
PROVEAN	0.7371	0.7444	0.6929	0.9363	0.3089	2.6825	0.327
SIFT	0.8099	0.861	0.5	0.9126	0.3723	2.6459	0.32
Polyphen-2	0.7978	0.8422	0.5286	0.9155	0.3558	2.6421	0.317
Mutation Assessor	0.747	0.7665	0.6286	0.9259	0.3077	2.6287	0.3
VEST	0.7503	0.8269	0.2857	0.8753	0.2139	2.2018	0.1
CanDrAplus (Cancer-in-general)	0.592	0.857	0	0.99	0	1.847	-0.03

Table 3.6: Comparison of the generated binary classifiers with other mutation effect prediction algorithms using the benchmarking dataset by Martelotto *et al.* (2014)

Algorithm	Accuracy	Sensitivity	Specificity	PPV	NPV	Composite Score	MCC
NBDriver + CHASMplus+ FATHMM (cancer) + Mutation Taster + Condel	0.945	0.985	0.689	0.95	0.88	3.504	0.746
CHASMplus+ FATHMM (cancer) + Mutation Taster + Condel	0.921	0.942	0.771	0.96	0.71	3.384	0.691
NBDriver + Mutation Taster + Condel	0.942	0.99	0.65	0.945	0.919	3.504	0.745

Table 3.7: Evaluating the contribution of NBDriver to the top performing ensemble

Evaluation using mutations derived from the Cancer Genome Interpreter Database

Using pathogenic mutations compiled from various sources, we found that our model NBDriver could accurately identify 1274 out of 1628 non-overlapping missense driver mutations achieving an overall accuracy of 78%. The model correctly identified all three mutations from the Cancer Biomarkers Database, 39 out of 47 mutations from the DoCM database, 23 out of 31 mutations from the Martelotto *et al.* (2014) and 1209 out of 1547 mutations from the Oncokb database. On the other hand, the ensemble model made up of the NBDriver, Condel and Mutation Taster could accurately predict 1519 out of 1628 mutations achieving an overall accuracy of 93%.

Evaluation using recurrent single point driver mutations reported by Rheinbay *et al.* (2017)

Out of the top 33 hotspot mutations identified in the study (Rheinbay *et al.*, 2017) as recurrently mutated, NBDriver correctly identified 27 mutations as drivers. However, Mutation Taster displayed superior performance by correctly

identifying all 33 mutations. Except for the *KRAS* oncogene, NBDriver correctly identified all mutations from the other four genes (*NRAS*, *TP53*, *PIK3CA* and *IDH1*) as cancer drivers.

Evaluation using rare driver mutations found in Glioblastoma and Ovarian Cancer

Using the list of rare drivers reported by the developers of the driver prediction tool CanDrA (Mao *et al.*, 2013), we evaluated NBDriver's ability to identify less frequent alterations in the cancer genome. Overall, NBDriver alone could identify 29 out of 34 (85%) glioblastoma mutations and 20 out of 38 (53%) ovarian cancer mutations. All these mutations belonged to eight known OVC-related genes (*ARID1A*, *CDK12*, *ERBB2*, *MLH1*, *MSH2*, *MSH6*, *PIK3R1*, *PMS2*) and seven known GBM-related genes (*ATM*, *EGFR*, *MDM2*, *NF1*, *PDGFRA*, *PIK3CA*, *ROS1*). The ensemble model made up of NBDriver, Condel and Mutation Taster performed better than the single predictor by identifying 32 out of 34 (94%) glioblastoma mutations and 24 out of 38 (63%) ovarian cancer mutations.

Stratification of the predicted driver genes based on literature

We combined the list of genes with at least one missense driver mutation prediction from NBDriver into a catalog of 138 putative driver genes. From this list, we calculated the fraction of mutations correctly predicted by NBDriver (Appendix A.6) and also compared our gene set against those already published in six landmark pan-cancer studies for driver gene identification. Bailey *et al.* (2018) identified 299 driver genes from 9423 tumor exomes by combining the predictions from 26 different computational tools. Martincorena *et al.* (2017) used the normalized ratio of non-synonymous to synonymous mutations (dN/dS model) to identify driver genes from 7664 tumors. They reported a total of 180 putatively positively-selected driver genes and 369 known cancer genes from three main sources:
1) 174 cancer genes from the version 73 of the COSMIC database (Tate *et al.*, 2019) 2) 214 significantly mutated genes across 4742 tumor patients identified

by Lawrence *et al.* (2014) using the MutSigCV tool and 3) 204 genes identified through a literature search. Two marker papers from TCGA (Hoadley *et al.*, 2018; Cancer Genome Atlas Research Network *et al.*, 2014) identified 132 significantly mutated genes using the MutSigCV tool. Tamborero *et al.* (2018) identified a list of 291 high-confidence drivers from 3205 tumor samples using a rule-based approach. Dietlein *et al.* (2020) modelled the nucleotide context around driver mutations and identified 460 driver genes based on nucleotide context. Apart from the aforementioned studies, the overlap between our list of genes and two well-established cancer gene repositories: the Cancer Gene Census (Futreal *et al.*, 2004; Forbes *et al.*, 2015) and the Intogen database (Martínez-Jiménez *et al.*, 2020) was also reported. We identified 124 (=89%) of our predicted driver genes as canonical cancer genes present in the Cancer Gene Census. Among the remaining genes, six were cataloged as drivers in at least two of the pan-cancer studies or mutation databases as mentioned above (Appendix A.5). A total of eight genes (*CTLA4*, *IGF1R*, *PIK3CD*, *TGFBR1*, *RAD54L*, *SHOC2*, *CDKN2B* and *XRCC2*) remained were not identifiable from any of the landmark studies or databases and required further validation.

We further investigated the prediction results obtained from NBDriver for each of the 138 cancer genes separately. Appendix A.6 shows the actual and the fraction of mutations correctly predicted by NBDriver for each of the five independent validation datasets. Figure 3.11 displays the list of 25 genes where NBDriver correctly predicted less than 70% of the mutations across the five independent validation datasets. From this figure we observed that in case of *KRAS* and *MYOD1*, NBDriver couldn't predict a single mutation correctly. NBDriver also displayed consistently poor gene-wise prediction results across multiple validation datasets. In case of the *ARAF* gene, for instance, NBDriver could accurately predict only 57% and 20% of the mutations from the Cancer Mutation Census and the Cancer Genome Interpreter, respectively. Similar observations were made for *DICER1*, *SMAD4* and *NF1* genes. The reason behind NBDriver's selective under-performance for this particular group of genes was not clear from our study and might warrant further investigation.

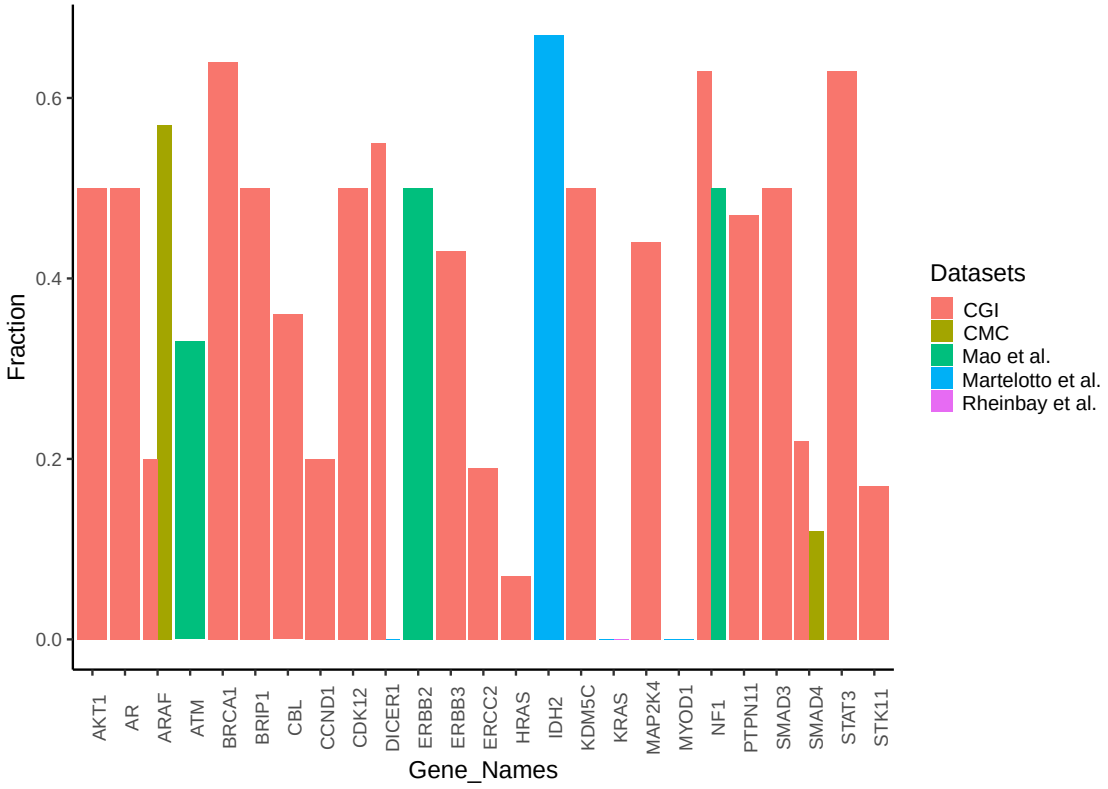

Figure 3.11: Plot showing the list genes where NBDriver correctly predicted less than 70% of the mutations (CMC=Cancer Mutation Census; CGI=Cancer Genome Interpreter).

3.6 Discussion

Our investigation aimed to compare the raw neighborhood sequences of driver and passenger mutations and exploiting any observed distributional differences to build robust classification models. We showed that except for one window size, a significant difference in the distributions between the neighborhoods of driver and passenger mutations was present in our cohort. Next, using TF-IDF and Count vectorizer scores derived from the overlapping *k*-mers, we trained a KDE-based generative classifier and two other tree-based classifiers. One important distinction between NBDriver and other methods is the inclusion of overlapping *k*-mers extracted from the neighborhood of mutations as features for further analysis. NBDriver was trained using a small set (=50) of highly discriminative features, 52% of which were neighborhood scores. Using this model, we could accurately predict 89% of all the literature-curated mutations outlined in the Martelotto *et al.* (2014) study, 81% of the high confidence list

of mutations recently published by the Cancer Mutation Census, 78% of all the actionable alterations reported in the Cancer Genome Interpreter, 82% of all the hotspot mutations reported from pan-cancer genome analysis, 85% and 53% of rare driver mutations found in GBM and OVC respectively. Ensemble models obtained by combining the predictions from other commonly used mutation effect predictors with NBDriver performed significantly better than the individual predictors in all five validation datasets. These results underscore the importance of including neighborhood features to build mutation effect prediction algorithms.

Our feature selection results illustrate the differences in the underlying biological processes governing driver and passenger mutations similar to the observations made by Mao *et al.* (2013). Using the training data, we observed that a mutation is more likely to be a driver if it occurs in genomic regions that were evolutionarily conserved. The mean GERP score for driver mutations was significantly higher (Wilcoxon test; $P < 2.2E-16$) than that of passengers (Figure 3.12(C)). Similarly, driver mutations were more common in genomic sites that had a significantly higher (Wilcoxon test; $P < 2.2E-16$) Positional Hidden Markov Model (HMM) conservation score (or HMMPHC) as compared to passengers (Figure 3.12(D)).

Among the other features, we found that driver mutations tend to occur on amino acid residues that have stiff backbones and have less solvent accessibility as denoted by the significantly higher (Wilcoxon test; $P = 2.1E-09$) "PredBFactorS" probability measure (Figure 3.12(B)) and the significantly lower (Wilcoxon test; $P = 5.4E-10$) "PREDRSAE" probability measure (Figure 3.12(A)) respectively. We observed similar class-wise distributional differences among features that were indicative of protein domain knowledge.
"UniprotDOM_PostModEnz" denotes the presence or absence of a mutation in a site within an enzymatic domain responsible for post-translational modification (or PTM). PTM-related mutations are often responsible for changes in protein functions and alterations of regulatory pathways eventually leading to carcinogenesis. "UniprotREGIONS" is another binary feature that tells us whether a mutation occurred in an experimentally defined region of interest in the protein sequence such as those associated with protein-protein interactions and regulation of biological processes. In our analysis, we observed that a considerable

portion (31%) of driver mutations clustered around PTM sites, contrasted by around 0.4% of passengers (Figure 3.12(E)). Similarly, around 37% of driver mutations were located in protein domains that were experimentally defined as regions of interest as compared to around 11% of passengers (Figure 3.12F). The TF-IDF algorithm is used to weigh a word (or *k*-mer in our case) and assign importance to the word in the given set of documents (or neighborhood sequences in our case). Hence, the higher the TF-IDF score, the more relevant/important that word is in that particular document. Our feature selection results indicated that for the 26 neighborhood sequence-based features, the mean TF-IDF scores for drivers were significantly higher (Wilcoxon test; $P < 0.05$) than that of passengers. This result suggests that top neighborhood features chosen by NBDriver are more specific to the driver neighborhoods than the passengers. Figure 3.13 shows the class-wise variation in the mean TF-IDF scores among the 26 neighborhood features used to train NBDriver.

We validated the true positive mutations that were identified by NBDriver with existing literature. The predicted driver mutations from the CMC have been implicated in many different types of cancers. For instance, mutations such as Y1248C and M1268I occur in the proto-oncogene *MET* and are associated with poor prognosis in renal cell carcinoma (Jeffers *et al.*, 1997). Similarly, the W515L mutations in the *MPL* oncogene have been shown to be helpful in identifying patients with Chronic Myeloproliferative Neoplasms (Akpınar *et al.*, 2013). Hotspot mutations occurring in the codon 835 of the *FLT3* oncogene have been implicated in the majority of AML and ALL patients (Liang *et al.*, 2003). Recurrent aberrations such as D816X and V560D found in the *KIT* oncogene have been associated with patients suffering from AML and gastric cancer respectively (Yui *et al.*, 2017; VanderPlas, 2016). Identifying recurrent alterations at the same genomic site across multiple samples (or hotspots) within the cancer genome is of special interest because of the universal evidence of these regions being responsible for positive selection driving tumorigenesis. Hotspot mutations in the genes reported by (Rheinbay *et al.*, 2017), correctly identified as drivers by NBDriver have been implicated in many different cancers. The *NRAS* gene, for instance, is part of the Ras family of oncogenes and plays an important role in cell division, differentiation and apoptosis. Pathogenic alterations in this gene

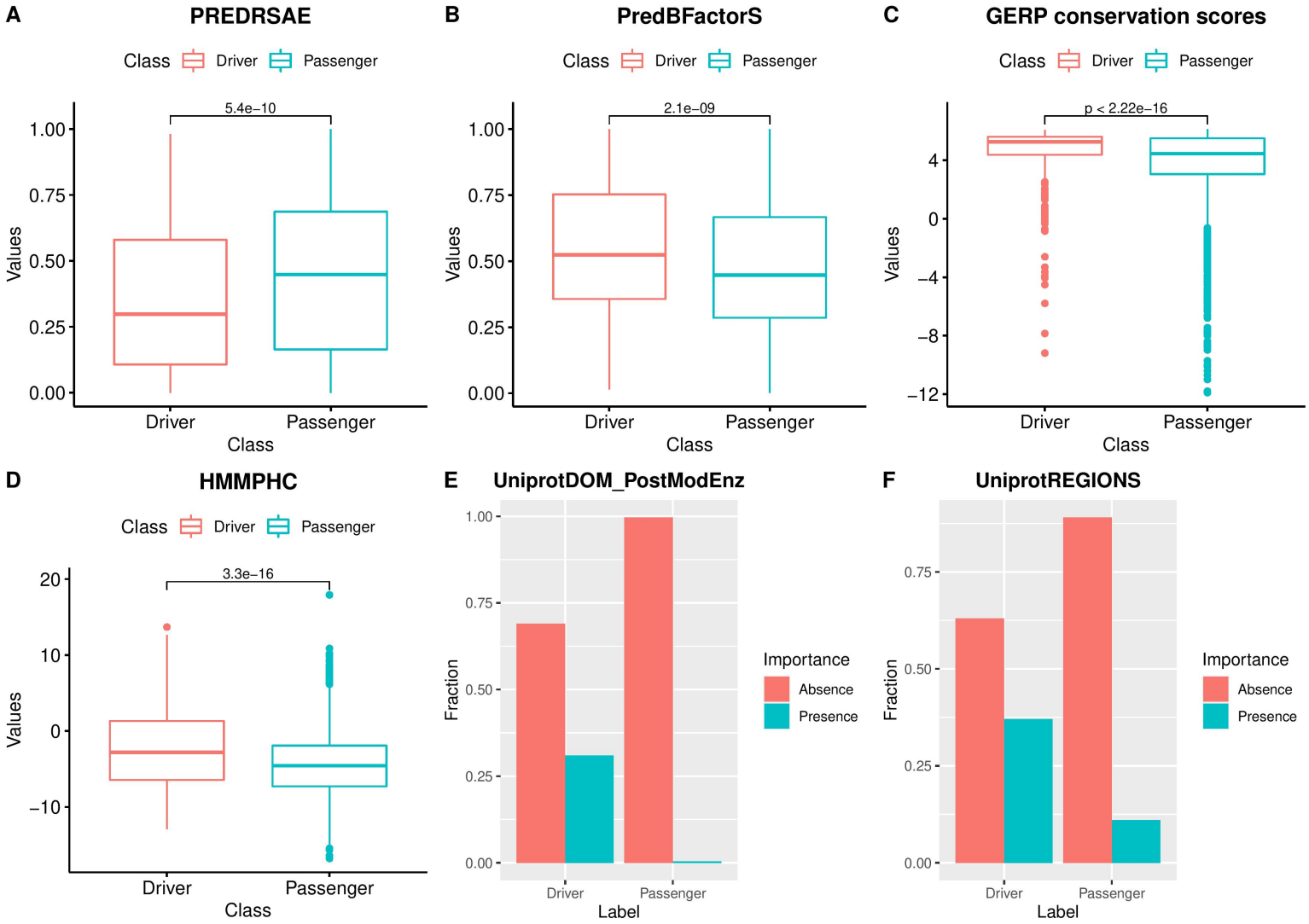

Figure 3.12: Differences in the distribution of features between driver and passenger mutations observed from the training data used to derive NBDriver.

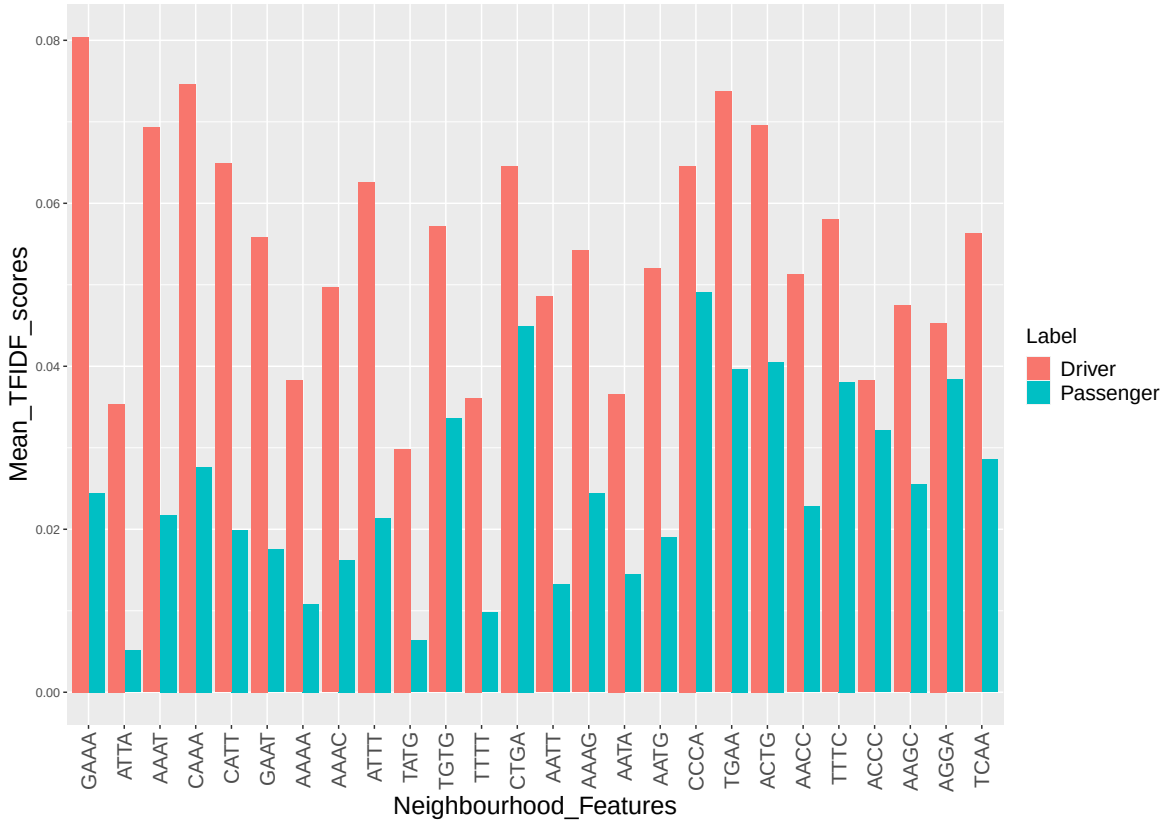

Figure 3.13: Class-wise variation in the mean TF-IDF scores among the 26 neighborhood features used to train NBDriver.

have been implicated in many different forms of cancers (Bos, 1989). Missense mutations in the well-known tumor suppressor gene, *TP53* mostly responsible for DNA repair and cell division is frequently altered in different cancer types (Olivier *et al.*, 2010). *PIK3CA* is one of the most commonly mutated oncogenes and activating mutations within this gene has been implicated in a wide variety of cancers (Samuels and Waldman, 2010). Finally, the *IDH1* oncogene is frequently mutated in glioma and acute myeloid leukemia (Yang *et al.*, 2012). Identifying rare driver mutations, especially those with low prevalence in non-hotspot regions, can be of particular interest in cancer genomics.

Although NBDriver didn't display a high predictive performance in identifying rare driver mutations in ovarian cancer (accuracy=53%), it performed reasonably well with respect to glioblastoma (accuracy=85%). Also, all eight OVC-related genes with at least one predicted driver missense mutation from NBDriver have been implicated in ovarian cancer. The ARID1A gene, for instance, has been found to be mutated in over 50% of ovarian clear cell carcinomas (Wiegand *et al.*, 2010). Similarly, the downregulation of recurrently

mutated tumor suppressor gene *CDK12*, has been associated with genome instability in serious ovarian carcinoma (Popova *et al.*, 2016). Recent studies have indicated that the expression of the human epidermal growth factor receptor 2 (*HER2*) gene has been associated with poor prognosis in ovarian cancer patients (Luo *et al.*, 2018). Four DNA mismatch repair genes (*MLH1*, *MSH2*, *MSH6*, *PMS2*) which are responsible for maintaining genomic stability have been identified as prognostic biomarkers associated with improved survival among ovarian cancer patients (Zhao *et al.*, 2018). Finally, *PIK3R1* has been identified as an oncogene in human ovarian and colon tumors (Philp *et al.*, 2001).

Although our method's focus was to identify missense driver mutations from sequenced cancer genomes, most of the genes (130 out of 138) containing at least one predicted mutation belonged to the CGC or other large-scale driver gene discovery studies. The protein products of the eight remaining genes not flagged as drivers by any of the databases/studies had known functional roles in maintaining the stability of the cancer genome and promoting tumor development. The *CTLA4* gene modulates immune response by serving as checkpoints for T-cell activation, essentially decreasing the ability of the T cells from attacking cancer cells. Immune checkpoint inhibitors which are designed to "block" these checkpoints have drastically changed the treatment outcomes for several cancers (Rotte, 2019). Transcriptomic profiling of blood samples drawn from cervical cancer patients identified *IGF1R* as a biomarker for increased risk of treatment failure (Moreno-Acosta *et al.*, 2012). Overexpression of the *PIK3CD* gene has been associated with cell proliferation in colon cancer and is responsible for poor prognosis among patients (Chen *et al.*, 2019). Multiple studies have indicated an association with polymorphisms observed in *TGFBR1* and cancer susceptibility (Pasche *et al.*, 2014; VanderPlas, 2016). Similarly, polymorphisms detected in the *RAD54L* is a genetic marker associated with the development of meningeal tumors (Leone *et al.*, 2003). *SHOC2* has been reported to be a regulator of the Ras signalling pathway and is associated with poor prognosis among breast cancer patients (Geng *et al.*, 2020). Inactivation of the *CDKN2B* gene is responsible for the progression of pancreatic cancer (Tu *et al.*, 2018). With the help of massively parallel sequencing studies, rare mutations in the *XRCC2* gene have been linked to increased breast cancer susceptibility among patients (Park

et al., 2012*b*).

3.7 Limitations

Our study does have some imitations. First, we used a representative dataset of driver and passenger mutations whose labels were not *in silico* predictions from other mutation effect prediction algorithms but derived from experimentally validated functional and transforming impacts from various sources. This resulted in a relatively small sample size for supervised classification. However, this approach also minimized the chances of inadvertently introducing false-positive mutations into the training set used to derive the class-wise density estimates for the driver and passenger neighborhoods or the machine learning models. There is also evidence (Chen *et al.*, 2017) suggesting that a sizeable proportion of mutations present in large mutational databases are mostly false positives, reflecting sequencing errors due to DNA damage. Moreover, NBDriver derived using this high confidence list of mutations performed reasonably well across all five independent validation sets and produced 138 driver genes with sufficient literature evidence suggesting that our initial choice of the training dataset was overall beneficial. Second, since missense mutations are the most abundant form of somatic alterations (Vogelstein *et al.*, 2013), our machine learning models were all trained using missense mutations only. However, in principle, our approach could be extended to other types of mutations as well. Additionally, during the external validation analysis, although NBDriver performed very well in terms of PPV (=0.941), the NPV (=0.608) was relatively low (Table 3.6). To identify biologically relevant mutations for further functional validation, NPV is often overlooked as a classification metric. A high NPV allows us to exclude passenger mutations with greater confidence and reduces the number of driver mutations incorrectly labeled as passengers (false negatives). However, we observed that adding different combinations of multiple single predictors into ensemble models resulted in a significant improvement in the NPV (Table 3.7). This is very similar to the observations made in an earlier study by Martelotto *et al.* (2014). Last, we trained our machine learning models using the combined dataset (Brown *et al.*, 2019) containing mutational effects determined

from experimental assays not specific to any cancer type. Hence, all our models were pan-cancer based. Consequently, a cancer-type specific analysis in the future would require known pathogenic and neutral mutations from specific tumor types.

CHAPTER 4

Conclusion and Future Work

Building robust machine learning models to distinguish between driver and passenger mutations from sequenced cancer genomes is an evolving research area. Many previous studies have tackled this problem by extracting different features, such as genomic, structural, conservation, and others. However, prioritizing mutations based on the context of the neighborhood is a relatively new concept. Using publicly available cancer mutation data in this study, we first analyzed the underlying distributional differences between driver and passenger neighborhoods. Our results indicated that except for one window size (n=1), there is a significant difference in the driver and passenger neighborhoods' distributions. Next, we built binary classification models to differentiate between the two types of mutations using various text-based feature representation techniques. We integrated genomic, structural, and conservation features into our initial neighborhood-only model to improve the overall classification performances. We observed that out of the 50 features used to derive NBDriver, 26 were neighborhood sequence-based features. This model gave classification performances comparable with other state-of-the-art mutation effect predictors across five separate validation sets. The predicted true positive mutations were part of genes with experimental support of being functionally relevant from multiple sources.

Our study can be used as a starting point for investigating several interesting research problems in the future. Some of them are listed below.

First, a user-friendly tool to differentiate between driver and passenger mutations based on the top 50 features obtained from our analysis can be developed. Although extracting the neighborhood sequence features is relatively straightforward, extracting the genomic, conservation, and structural features from different sources is complicated. Like the different mutation effect predictors, the proposed tool should generate all these features on the go and return the processed feature matrix for a given set of mutations. Next, it should run NBDriver

and the ensemble models and generate predictions in the form of probability scores returned by the classifiers instead of crisp class labels. The users will then choose to define their thresholds while deriving the driver and passenger mutations' final list. Second, more efficient feature representations of the sequence neighborhoods can be derived to train the machine learning models. Popular deep learning approaches, such as *Recurrent Neural Networks* can be used for this purpose. Using these feature representations, unsupervised techniques, such as clustering, can group different mutations into biologically relevant clusters. Further, it would be interesting to observe whether there are certain sequence motifs in the genome's non-coding portion that are more prone to developing driver mutations than others. Fredriksson *et al.* (2017) showed that recurrent promoter mutations in melanoma occur almost exclusively at cytosines, surrounded by a distinct nucleotide context ("TTCCG"). Finally, future experiments using a much larger sample size need to be performed to derive neighborhood-sequence-based classification scores for all possible missense mutations in the genome across several cancer types. This would be possible if future large-scale sequencing studies such as MSK-IMPACT (Cheng *et al.*, 2017), PCAWG (Rheinbay *et al.*, 2017), ICGC (Zhang *et al.*, 2011) and GENIE (AACR Project GENIE Consortium, 2017) produce a more complete catalog of missense driver mutations with functional evidence in a cancer-type specific manner.

Annotating missense mutations from sequenced cancer genomes hold much promise for precision medicine. Recently, many databases have been developed to deposit cancer-causing mutations. The CGC (Futreal *et al.* (2004); Forbes *et al.* (2015)) project which is part of the COSMIC database, aims to catalog all cancer driver genes (TSGs or OGs). OncoKB (Chakravarty *et al.* (2017)) is a comprehensive database containing evidence-based information regarding somatic alterations. MutPanning (Dietlein *et al.* (2020)) provides a comprehensive resource for driver genes identified from 28 cancer types based on mutations observed in unusual nucleotide contexts. Sleeping Beauty Cancer Driver Database (SBCDDB) contains a list of driver genes identified by the Sleeping Beauty insertional mutagenesis (Newberg *et al.* (2018)). Databases such as DriverDB3 (Liu *et al.* (2020)) and Intogen (Martínez-Jiménez *et al.* (2020)) integrate cancer driver genes and provides essential visualizations of large-scale omics datasets.

However, these platforms are not suited to annotate and prioritize thousands of samples at once. The complexity of any ML algorithm depends directly on the number of features used to train the models. Complex driver mutation prioritization tools use lots of features to effectively characterize cancer-causing variants. NBDriver, on the other hand, uses a small set (50) of highly discriminative features that is easily obtainable from online databases. 26 out of the 50 features are based on neighborhood sequences and can be extracted from the corresponding genome build. The remaining features are all present in the SNVBOX database and can be accessed via the command line interface. This relatively less complicated feature representation makes the underlying ML model fast and easy to obtain predictions using thousands of variants at a time.

Overall, this thesis has shown that we can utilize the local sequence context surrounding a particular mutation to predict its deleteriousness. This relatively novel strategy of utilizing the sequence neighborhoods for driver mutation identification can dramatically improve the annotation process's efficiency for unknown mutations.

APPENDIX A

Online Appendices

This appendix provides links to the excel files containing additional results for the analyses presented in this thesis. These files are available online at the following Github repository. The necessary codes required to reproduce the results presented in this study is available at the following link (https://github.com/RamanLab/NBDriver).

A.1 Repeated cross-validation results using the neighborhood sequences as features

This excel file titled `RepeatedCV_Results` contains the repeated cross-validation results using just the neighborhood sequences as features. We report the classification performances on the basis of four metrics: Sensitivity, Specificity, AUROC and MCC.

A.2 Variation in the classification performances with increase in the window size

This excel file titled `Variation_in_ClassificationResults` lists the results of the Wilcoxon signed-rank test which was designed to test the significance of the increase in the classification performances with the increase in the size of the neighborhood. Each entry (x,y) in this table signifies that if we increase our window size from x to y, we get an overall increase in the corresponding classification metric, and the increase is significant (WRS test; $P < 0.05$).

A.3 Ranked list of the 50 features used the train NBDriver

This excel file titled `RankedFeatures` contain the list of 50 features used to train the machine learning model. Here, all 26 neighborhood features have been shaded in red.

A.4 Ensemble model performances

This excel file titled `EnsembleModels` contain the results of combining various mutation effect predictors on the Martelotto *et al.* (2014) dataset.

A.5 Stratification of driver genes based on literature evidence

This excel file titled `IdentifiedDriverGenes` contain the list of 138 driver genes identified while validating our model on the five independent validation sets. The overlap of these genes with several published landmark studies, the CGC and their classification into TSG/OG is also displayed in this table. Novel driver genes that had no overlap with any of the above sources are also highlighted.

A.6 Gene-wise prediction results obtained using NBDriver

This excel file titled `GenewisePredictionResults` contain the fraction of mutations belonging to the 138 genes correctly predicted by NBDriver along with the total number of mutations.

REFERENCES

1. **AACR Project GENIE Consortium** (2017). AACR Project GENIE: Powering Precision Medicine through an International Consortium. *Cancer Discovery*, **7**(8), 818–831. ISSN 2159-8290.

2. **Adzhubei, I. A.**, **S. Schmidt**, **L. Peshkin**, **V. E. Ramensky**, **A. Gerasimova**, **P. Bork**, **A. S. Kondrashov**, and **S. R. Sunyaev** (2010). A method and server for predicting damaging missense mutations. *Nature Methods*, **7**(4), 248–249. ISSN 1548-7105.

3. **Agajanian, S.**, **O. Oluyemi**, and **G. M. Verkhivker** (2019). Integration of random forest classifiers and deep convolutional neural networks for classification and biomolecular modeling of cancer driver mutations. *Frontiers in Molecular Biosciences*, **6**, 44. ISSN 2296-889X.

4. **Aggarwala, V.** and **B. F. Voight** (2016). An expanded sequence context model broadly explains variability in polymorphism levels across the human genome. *Nature Genetics*, **48**(4), 349–355. ISSN 1546-1718. URL https://doi.org/10.1038/ng.3511.

5. **Ainscough, B. J.**, **M. Griffith**, **A. C. Coffman**, **A. H. Wagner**, **J. Kunisaki**, **M. N. Choudhary**, **J. F. McMichael**, **R. S. Fulton**, **R. K. Wilson**, **O. L. Griffith**, and **E. R. Mardis** (2016). DoCM: A database of curated mutations in cancer. *Nature Methods*, **13**(10), 806–807. ISSN 1548-7105.

6. **Akpınar, T. S.**, **V. S. Hançer**, **M. Nalçacı**, and **R. Diz-Küçükkaya** (2013). MPL W515L/K Mutations in Chronic Myeloproliferative Neoplasms. *Turkish Journal of Hematology*, **30**(1), 8–12. ISSN 13007777.

7. **Alexandrov, L. B.**, **S. Nik-Zainal**, **D. C. Wedge**, **S. A. Aparicio**, **S. Behjati**, **A. V. Biankin**, **G. R. Bignell**, **N. Bolli**, **A. Borg**, **A.-L. Børresen-Dale**, *et al.* (2013). Signatures of mutational processes in human cancer. *Nature*, **500**(7463), 415–421.

8. **Arnedo-Pac, C.**, **L. Mularoni**, **F. Muiños**, **A. Gonzalez-Perez**, and **N. Lopez-Bigas** (2019). OncodriveCLUSTL: a sequence-based clustering method to identify cancer drivers. *Bioinformatics*, **35**(22), 4788–4790. ISSN 1367-4803. URL https://doi.org/10.1093/bioinformatics/btz501.

9. **Bailey, M. H.**, **C. Tokheim**, **E. Porta-Pardo**, **S. Sengupta**, **D. Bertrand**, **A. Weerasinghe**, **A. Colaprico**, **M. C. Wendl**, **J. Kim**, **B. Reardon**, *et al.* (2018). Comprehensive characterization of cancer driver genes and mutations. *Cell*, **173**(2), 371–385.

10. **Bos, J. L.** (1989). ras oncogenes in human cancer: A review. *Cancer Research*, **49**(17), 4682–4689. ISSN 0008-5472.

11. **Breiman, L.** (2001). Random forests. *Machine Learning*, **45**(1), 5–32.

12. **Brown, A.-L.**, **M. Li**, **A. Goncearenco**, and **A. R. Panchenko** (2019). Finding driver mutations in cancer: Elucidating the role of background mutational processes. *PLoS Computational Biology*, **15**(4), e1006981.

13. **Campbell, B. B.**, **N. Light**, **D. Fabrizio**, **M. Zatzman**, **F. Fuligni**, **R. de Borja**, **S. Davidson**, **M. Edwards**, **J. A. Elvin**, **K. P. Hodel**, *et al.* (2017). Comprehensive analysis of hypermutation in human cancer. *Cell*, **171**(5), 1042–1056.

14. **Cancer Genome Atlas Research Network** *et al.* (2008). Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, **455**(7216), 1061.

15. **Cancer Genome Atlas Research Network** *et al.* (2011). Integrated genomic analyses of ovarian carcinoma. *Nature*, **474**(7353), 609.

16. **Cancer Genome Atlas Research Network** *et al.* (2014). Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, **511**(7511), 543–550.

17. **Carlson, J.**, **A. E. Locke**, **M. Flickinger**, **M. Zawistowski**, **S. Levy**, **R. M. Myers**, **M. Boehnke**, **H. M. Kang**, **L. J. Scott**, **J. Z. Li**, *et al.* (2018). Extremely rare variants reveal patterns of germline mutation rate heterogeneity in humans. *Nature Communications*, **9**(1), 1–13.

18. **Carter, H.**, **S. Chen**, **L. Isik**, **S. Tyekucheva**, **V. E. Velculescu**, **K. W. Kinzler**, **B. Vogelstein**, and **R. Karchin** (2009). Cancer-specific high-throughput annotation of somatic mutations: Computational prediction of driver missense mutations. *Cancer Research*, **69**(16), 6660–6667.

19. **Carter, H.**, **C. Douville**, **P. D. Stenson**, **D. N. Cooper**, and **R. Karchin** (2013). Identifying mendelian disease genes with the variant effect scoring tool. *BMC Genomics*, **14**(3), 1–16.

20. **Carter, S. L.**, **K. Cibulskis**, **E. Helman**, **A. McKenna**, **H. Shen**, **T. Zack**, **P. W. Laird**, **R. C. Onofrio**, **W. Winckler**, **B. A. Weir**, *et al.* (2012). Absolute quantification of somatic dna alterations in human cancer. *Nature Biotechnology*, **30**(5), 413–421.

21. **Chakravarty, D.**, **J. Gao**, **S. Phillips**, **R. Kundra**, **H. Zhang**, **J. Wang**, **J. E. Rudolph**, **R. Yaeger**, **T. Soumerai**, **M. H. Nissan**, **M. T. Chang**, **S. Chandarlapaty**, **T. A. Traina**, **P. K. Paik**, **A. L. Ho**, **F. M. Hantash**, **A. Grupe**, **S. S. Baxi**, **M. K. Callahan**, **A. Snyder**, **P. Chi**, **D. C. Danila**, **M. Gounder**, **J. J. Harding**, **M. D. Hellmann**, **G. Iyer**, **Y. Y. Janjigian**, **T. Kaley**, **D. A. Levine**, **M. Lowery**, **A. Omuro**, **M. A. Postow**, **D. Rathkopf**, **A. N. Shoushtari**, **N. Shukla**, **M. H. Voss**, **E. Paraiso**, **A. Zehir**, **M. F. Berger**, **B. S. Taylor**, **L. B. Saltz**, **G. J. Riely**, **M. Ladanyi**, **D. M. Hyman**, **J. Baselga**, **P. Sabbatini**, **D. B. Solit**, and **N. Schultz** (2017). OncoKB: A Precision Oncology Knowledge Base. *JCO Precision Oncology*, (1), 1–16.

22. **Chen, C.**, **A. Liaw**, **L. Breiman**, *et al.* (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*, **110**(1-12), 24.

23. **Chen, J.-S.**, **J.-Q. Huang**, **B. Luo**, **S.-H. Dong**, **R.-C. Wang**, **Z.-k. Jiang**, **Y.-K. Xie**, **W. Yi**, **G.-M. Wen**, and **J.-F. Zhong** (2019). PIK3CD induces cell growth and invasion by activating AKT/GSK-3/-catenin signaling in colorectal cancer.

24. **Chen, K.**, **J. W. Wallis**, **M. D. McLellan**, **D. E. Larson**, **J. M. Kalicki**, **C. S. Pohl**, **S. D. McGrath**, **M. C. Wendl**, **Q. Zhang**, **D. P. Locke**, **X. Shi**, **R. S. Fulton**, **T. J. Ley**, **R. K. Wilson**, **L. Ding**, and **E. R. Mardis** (2009). BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nature Methods*, **6**(9), 677–681. ISSN 1548-7105.

25. **Chen, L.**, **P. Liu**, **T. C. Evans**, and **L. M. Ettwiller** (2017). DNA damage is a pervasive cause of sequencing errors, directly confounding variant identification. *Science*, **355**(6326), 752–756. ISSN 0036-8075, 1095-9203.

26. **Cheng, D. T.**, **M. Prasad**, **Y. Chekaluk**, **R. Benayed**, **J. Sadowska**, **A. Zehir**, **A. Syed**, **Y. E. Wang**, **J. Somar**, **Y. Li**, *et al.* (2017). Comprehensive detection of germline variants by msk-impact, a clinical diagnostic platform for solid tumor molecular oncology and concurrent cancer predisposition testing. *BMC Medical Genomics*, **10**(1), 33.

27. **Choi, Y.**, **G. E. Sims**, **S. Murphy**, **J. R. Miller**, and **A. P. Chan** (2012). Predicting the functional effect of amino acid substitutions and indels. *PLoS One*, **7**(10), e46688.

28. **Cibulskis, K.**, **M. S. Lawrence**, **S. L. Carter**, **A. Sivachenko**, **D. Jaffe**, **C. Sougnez**, **S. Gabriel**, **M. Meyerson**, **E. S. Lander**, and **G. Getz** (2013). Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology*, **31**(3), 213–219.

29. **Cooper** *et al.* (1995). The nature and mechanisms of human gene mutation. *The Metabolic and Molecular Bases of Inherited Disease*.

30. **Davies, H.**, **G. R. Bignell**, **C. Cox**, **P. Stephens**, **S. Edkins**, **S. Clegg**, **J. Teague**, **H. Woffendin**, **M. J. Garnett**, **W. Bottomley**, *et al.* (2002). Mutations of the BRAF gene in human cancer. *Nature*, **417**(6892), 949–954.

31. **Dees, N. D.**, **Q. Zhang**, **C. Kandoth**, **M. C. Wendl**, **W. Schierding**, **D. C. Koboldt**, **T. B. Mooney**, **M. B. Callaway**, **D. Dooling**, **E. R. Mardis**, *et al.* (2012). MuSiC: Identifying mutational significance in cancer genomes. *Genome Research*, **22**(8), 1589–1598.

32. **Dietlein, F.**, **D. Weghorn**, **A. Taylor-Weiner**, **A. Richters**, **B. Reardon**, **D. Liu**, **E. S. Lander**, **E. M. Van Allen**, and **S. R. Sunyaev** (2020). Identification of cancer driver genes based on nucleotide context. *Nature Genetics*, **52**(2), 208–218.

33. **Dietterich, T. G.**, Ensemble methods in machine learning. *In International workshop on multiple classifier systems*. Springer, 2000.

34. **Ding, J.**, **A. Bashashati**, **A. Roth**, **A. Oloumi**, **K. Tse**, **T. Zeng**, **G. Haffari**, **M. Hirst**, **M. A. Marra**, **A. Condon**, *et al.* (2012). Feature-based classifiers for somatic mutation detection in tumour–normal paired sequencing data. *Bioinformatics*, **28**(2), 167–175.

35. **Drake, J. W.** (1969). Mutagenic mechanisms. *Annual Review of Genetics*, **3**(1), 247–268.

36. **Forbes, S. A.**, **D. Beare**, **P. Gunasekaran**, **K. Leung**, **N. Bindal**, **H. Boutselakis**, **M. Ding**, **S. Bamford**, **C. Cole**, **S. Ward**, *et al.* (2015). COSMIC: Exploring the world's knowledge of somatic mutations in human cancer. *Nucleic Acids Research*, **43**(D1), D805–D811.

37. **Franceschini, A.**, **D. Szklarczyk**, **S. Frankild**, **M. Kuhn**, **M. Simonovic**, **A. Roth**, **J. Lin**, **P. Minguez**, **P. Bork**, **C. Von Mering**, *et al.* (2012). String v9. 1: Protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, **41**(D1), D808–D815.

38. **Fredriksson, N. J.**, **K. Elliott**, **S. Filges**, **J. Van den Eynden**, **A. Ståhlberg**, and **E. Larsson** (2017). Recurrent promoter mutations in melanoma are defined by an extended context-specific mutational signature. *PLoS Genetics*, **13**(5), e1006773.

39. **Futreal, P. A.**, **L. Coin**, **M. Marshall**, **T. Down**, **T. Hubbard**, **R. Wooster**, **N. Rahman**, and **M. R. Stratton** (2004). A census of human cancer genes. *Nature Reviews Cancer*, **4**(3), 177–183.

40. **Garraway, L. A.** (2013). Genomics-driven oncology: Framework for an emerging paradigm. *Journal of Clinical Oncology*, **31**(15), 1806–1814. PMID: 23589557.

41. **Geng, W.**, **K. Dong**, **Q. Pu**, **Y. Lv**, and **H. Gao** (2020). SHOC2 is associated with the survival of breast cancer cells and has prognostic value for patients with breast cancer. *Molecular Medicine Reports*, **21**(2), 867–875.

42. **Gonzalez-Perez, A.**, **J. Deu-Pons**, and **N. Lopez-Bigas** (2012). Improving the prediction of the functional impact of cancer mutations by baseline tolerance transformation. *Genome Medicine*, **4**(11), 89.

43. **Gonzalez-Perez, A.** and **N. Lopez-Bigas** (2012). Functional impact bias reveals cancer drivers. *Nucleic Acids Research*, **40**(21), e169–e169.

44. **Hanahan, D.** and **R. A. Weinberg** (2011). Hallmarks of cancer: The next generation. *Cell*, **144**(5), 646–674.

45. **Hershberg, R.** and **D. A. Petrov** (2008). Selection on codon bias. *Annual Review of Genetics*, **42**, 287–299.

46. **Hoadley, K. A.**, **C. Yau**, **T. Hinoue**, **D. M. Wolf**, **A. J. Lazar**, **E. Drill**, **R. Shen**, **A. M. Taylor**, **A. D. Cherniack**, **V. Thorsson**, *et al.* (2018). Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell*, **173**(2), 291–304.

47. **Hodgkinson, A.** and **A. Eyre-Walker** (2011). Variation in the mutation rate across mammalian genomes. *Nature Reviews Genetics*, **12**(11), 756–766.

48. **Hodis, E.**, **I. R. Watson**, **G. V. Kryukov**, **S. T. Arold**, **M. Imielinski**, **J.-P. Theurillat**, **E. Nickerson**, **D. Auclair**, **L. Li**, **C. Place**, *et al.* (2012). A landscape of driver mutations in melanoma. *Cell*, **150**(2), 251–263.

49. **Hua, X.**, **H. Xu**, **Y. Yang**, **J. Zhu**, **P. Liu**, and **Y. Lu** (2013). DrGaP: A powerful tool for identifying driver genes and pathways in cancer sequencing studies. *The American Journal of Human Genetics*, **93**(3), 439–451.

50. **Illingworth, R. S.**, **U. Gruenewald-Schneider**, **S. Webb**, **A. R. Kerr**, **K. D. James**, **D. J. Turner**, **C. Smith**, **D. J. Harrison**, **R. Andrews**, and **A. P. Bird** (2010). Orphan CpG islands identify numerous conserved promoters in the mammalian genome. *PLoS Genetics*, **6**(9), e1001134.

51. **Iorio, F.**, **T. A. Knijnenburg**, **D. J. Vis**, **G. R. Bignell**, **M. P. Menden**, **M. Schubert**, **N. Aben**, **E. Gonçalves**, **S. Barthorpe**, **H. Lightfoot**, *et al.* (2016). A landscape of pharmacogenomic interactions in cancer. *Cell*, **166**(3), 740–754.

52. **Jeffers, M.**, **L. Schmidt**, **N. Nakaigawa**, **C. P. Webb**, **G. Weirich**, **T. Kishida**, **B. Zbar**, and **G. F. V. Woude** (1997). Activating mutations for the Met tyrosine kinase receptor in human cancer. *Proceedings of the National Academy of Sciences of the United States of America*, **94**(21), 11445–11450.

53. **Koboldt, D. C.**, **Q. Zhang**, **D. E. Larson**, **D. Shen**, **M. D. McLellan**, **L. Lin**, **C. A. Miller**, **E. R. Mardis**, **L. Ding**, and **R. K. Wilson** (2012). VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research*, **22**(3), 568–576.

54. **Krawczak, M.**, **E. V. Ball**, and **D. N. Cooper** (1998). Neighboring-nucleotide effects on the rates of germ-line single-base-pair substitution in human genes. *The American Journal of Human Genetics*, **63**(2), 474–488.

55. **Kumar, P.**, **S. Henikoff**, and **P. C. Ng** (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nature Protocols*, **4**(7), 1073.

56. **Landrum, M. J.**, **J. M. Lee**, **M. Benson**, **G. Brown**, **C. Chao**, **S. Chitipiralla**, **B. Gu**, **J. Hart**, **D. Hoffman**, **J. Hoover**, *et al.* (2016). ClinVar: Public archive of interpretations of clinically relevant variants. *Nucleic Acids Research*, **44**(D1), D862–D868.

57. **Lawrence, M. S.**, **P. Stojanov**, **C. H. Mermel**, **J. T. Robinson**, **L. A. Garraway**, **T. R. Golub**, **M. Meyerson**, **S. B. Gabriel**, **E. S. Lander**, and **G. Getz** (2014). Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature*, **505**(7484), 495–501.

58. **Lawrence, M. S.**, **P. Stojanov**, **P. Polak**, **G. V. Kryukov**, **K. Cibulskis**, **A. Sivachenko**, **S. L. Carter**, **C. Stewart**, **C. H. Mermel**, **S. A. Roberts**, *et al.* (2013). Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, **499**(7457), 214–218.

59. **Leone, P. E.**, **M. Mendiola**, **J. Alonso**, **C. Paz-y Miño**, and **A. Pestaña** (2003). Implications of a RAD54L polymorphism (2290C/T) in human meningiomas as a risk factor and/or a genetic marker. *BMC Cancer*, **3**(1), 6.

60. **Liang, D.**, **L. Shih**, **I. Hung**, **C. Yang**, **S. Chen**, **T. Jaing**, **H. Liu**, **L. Wang**, and **W. Chang** (2003). FLT3-TKD mutation in childhood acute myeloid leukemia. *Leukemia*, **17**(5), 883–886.

61. **Liu, S.-H.**, **P.-C. Shen**, **C.-Y. Chen**, **A.-N. Hsu**, **Y.-C. Cho**, **Y.-L. Lai**, **F.-H. Chen**, **C.-Y. Li**, **S.-C. Wang**, **M. Chen**, *et al.* (2020). Driverdbv3: a multi-omics database for cancer driver gene research. *Nucleic acids research*, **48**(D1), D863–D870.

62. **Luo, H.**, **X. Xu**, **M. Ye**, **B. Sheng**, and **X. Zhu** (2018). The prognostic value of HER2 in ovarian cancer: A meta-analysis of observational studies. *PloS One*, **13**(1), e0191972.

63. **Mahmood, K.**, **C.-h. Jung**, **G. Philip**, **P. Georgeson**, **J. Chung**, **B. J. Pope**, and **D. J. Park** (2017). Variant effect prediction tools assessed using independent, functional assay-based datasets: Implications for discovery and diagnostics. *Human Genomics*, **11**(1), 1–8.

64. **Mao, Y.**, **H. Chen**, **H. Liang**, **F. Meric-Bernstam**, **G. B. Mills**, and **K. Chen** (2013). CanDrA: Cancer-specific driver missense mutation annotation with optimized features. *PloS One*, **8**(10), e77945.

65. **Martelotto, L. G.**, **C. K. Ng**, **M. R. De Filippo**, **Y. Zhang**, **S. Piscuoglio**, **R. S. Lim**, **R. Shen**, **L. Norton**, **J. S. Reis-Filho**, and **B. Weigelt** (2014). Benchmarking mutation effect prediction algorithms using functionally validated cancer-related missense mutations. *Genome Biology*, **15**(10), 484.

66. **Martincorena, I.**, **K. M. Raine**, **M. Gerstung**, **K. J. Dawson**, **K. Haase**, **P. Van Loo**, **H. Davies**, **M. R. Stratton**, and **P. J. Campbell** (2017). Universal patterns of selection in cancer and somatic tissues. *Cell*, **171**(5), 1029–1041.

67. **Martínez-Jiménez, F.**, **F. Muiños**, **I. Sentís**, **J. Deu-Pons**, **I. Reyes-Salazar**, **C. Arnedo-Pac**, **L. Mularoni**, **O. Pich**, **J. Bonet**, **H. Kranas**, *et al.* (2020). A compendium of mutational cancer driver genes. *Nature Reviews Cancer*, **20**(10), 555–572.

68. **Masica, D. L.**, **C. Douville**, **C. Tokheim**, **R. Bhattacharya**, **R. Kim**, **K. Moad**, **M. C. Ryan**, and **R. Karchin** (2017). CRAVAT 4: Cancer-related analysis of variants toolkit. *Cancer Research*, **77**(21), e35–e38.

69. **Michaelson, J. J.**, **Y. Shi**, **M. Gujral**, **H. Zheng**, **D. Malhotra**, **X. Jin**, **M. Jian**, **G. Liu**, **D. Greer**, **A. Bhandari**, *et al.* (2012). Whole-genome sequencing in

autism identifies hot spots for de novo germline mutation. *Cell*, **151**(7), 1431–1442.

70. **Millar, C. B.**, **J. Guy**, **O. J. Sansom**, **J. Selfridge**, **E. MacDougall**, **B. Hendrich**, **P. D. Keightley**, **S. M. Bishop**, **A. R. Clarke**, and **A. Bird** (2002). Enhanced CpG mutability and tumorigenesis in MBD4-Deficient mice. *Science*, **297**(5580), 403–405.

71. **Moreno-Acosta, P.**, **O. Gamboa**, **M. S. De Gomez**, **R. Cendales**, **G. D. Diaz**, **A. Romero**, **J. B. Serra**, **Z. Conrado**, **A. Levy**, **C. Chargari**, *et al.* (2012). IGF1R gene expression as a predictive marker of response to ionizing radiation for patients with locally advanced HPV16-positive cervical cancer. *Anticancer Research*, **32**(10), 4319–4325.

72. **Newberg, J. Y.**, **K. M. Mann**, **M. B. Mann**, **N. A. Jenkins**, and **N. G. Copeland** (2018). Sbcddb: Sleeping beauty cancer driver database for gene discovery in mouse models of human cancers. *Nucleic acids research*, **46**(D1), D1011–D1017.

73. **Ng, P. K.-S.**, **J. Li**, **K. J. Jeong**, **S. Shao**, **H. Chen**, **Y. H. Tsang**, **S. Sengupta**, **Z. Wang**, **V. H. Bhavana**, **R. Tran**, *et al.* (2018). Systematic functional annotation of somatic mutations in cancer. *Cancer Cell*, **33**(3), 450–462.

74. **Nik-Zainal, S.**, **L. B. Alexandrov**, **D. C. Wedge**, **P. Van Loo**, **C. D. Greenman**, **K. Raine**, **D. Jones**, **J. Hinton**, **J. Marshall**, **L. A. Stebbings**, *et al.* (2012). Mutational processes molding the genomes of 21 breast cancers. *Cell*, **149**(5), 979–993.

75. **Nowell, P. C.** (1976). The clonal evolution of tumor cell populations. *Science*, **194**(4260), 23–28.

76. **Oesper, L.**, **A. Mahmoody**, and **B. J. Raphael** (2013). Theta: Inferring intra-tumor heterogeneity from high-throughput DNA sequencing data. *Genome Biology*, **14**(7), R80.

77. **Olivier, M.**, **R. Eeles**, **M. Hollstein**, **M. A. Khan**, **C. C. Harris**, and **P. Hainaut** (2002). The IARC TP53 database: New online mutation analysis and recommendations to users. *Human Mutation*, **19**(6), 607–614.

78. **Olivier, M.**, **M. Hollstein**, and **P. Hainaut** (2010). TP53 mutations in human cancers: Origins, consequences, and clinical use. *Cold Spring Harbor Perspectives in Biology*, **2**(1), a001008.

79. **Pagel, K. A.**, **R. Kim**, **K. Moad**, **B. Busby**, **L. Zheng**, **C. Tokheim**, **M. Ryan**, and **R. Karchin** (2020). Integrated informatics analysis of cancer-related variants. *JCO Clinical Cancer Informatics*, **4**, 310–317.

80. **Park, C.**, **W. Qian**, and **J. Zhang** (2012*a*). Genomic evidence for elevated mutation rates in highly expressed genes. *EMBO Reports*, **13**(12), 1123–1129.

81. **Park, D.**, **F. Lesueur**, **T. Nguyen-Dumont**, **M. Pertesi**, **F. Odefrey**, **F. Hammet**, **S. L. Neuhausen**, **E. M. John**, **I. L. Andrulis**, **M. B. Terry**, *et al.* (2012*b*). Rare mutations in XRCC2 increase the risk of breast cancer. *The American Journal of Human Genetics*, **90**(4), 734–739.

82. **Pasche, B.**, **M. J. Pennison**, **H. Jimenez**, and **M. Wang** (2014). TGFBR1 and cancer susceptibility. *Transactions of the American Clinical and Climatological Association*, **125**, 300.

83. **Pedregosa, F.**, **G. Varoquaux**, **A. Gramfort**, **V. Michel**, **B. Thirion**, **O. Grisel**, **M. Blondel**, **P. Prettenhofer**, **R. Weiss**, **V. Dubourg**, *et al.* (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, **12**, 2825–2830.

84. **Perry, S. C.** and **R. G. Beiko** (2010). Distinguishing microbial genome fragments based on their composition: Evolutionary and comparative genomic perspectives. *Genome Biology and Evolution*, **2**, 117–131.

85. **Philp, A. J.**, **I. G. Campbell**, **C. Leet**, **E. Vincan**, **S. P. Rockman**, **R. H. Whitehead**, **R. J. Thomas**, and **W. A. Phillips** (2001). The phosphatidylinositol 3'-kinase p85alpha gene is an oncogene in human ovarian and colon tumors. *Cancer Research*, **61**(20), 7426–7429.

86. **Pollard, K. S.**, **M. J. Hubisz**, **K. R. Rosenbloom**, and **A. Siepel** (2010). Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Research*, **20**(1), 110–121.

87. **Popova, T.**, **E. Manié**, **V. Boeva**, **A. Battistella**, **O. Goundiam**, **N. K. Smith**, **C. R. Mueller**, **V. Raynal**, **O. Mariani**, **X. Sastre-Garau**, *et al.* (2016). Ovarian cancers harboring inactivating mutations in CDK12 display a distinct genomic instability pattern characterized by large tandem duplications. *Cancer Research*, **76**(7), 1882–1891.

88. **Raphael, B. J.**, **J. R. Dobson**, **L. Oesper**, and **F. Vandin** (2014). Identifying driver mutations in sequenced cancer genomes: Computational approaches to enable precision medicine. *Genome Medicine*, **6**(1), 1–17.

89. **Reva, B.**, **Y. Antipin**, and **C. Sander** (2011). Predicting the functional impact of protein mutations: Application to cancer genomics. *Nucleic Acids Research*, **39**(17), e118–e118.

90. **Rheinbay, E.**, **M. M. Nielsen**, **F. Abascal**, **G. Tiao**, **H. Hornshøj**, **J. M. Hess**, **R. I. Pedersen**, **L. Feuerbach**, **R. Sabarinathan**, **T. Madsen**, *et al.* (2017). Discovery and characterization of coding and non-coding driver mutations in more than 2,500 whole cancer genomes. *BioRxiv*, 237313.

91. **Rogozin, I. B.**, **Y. I. Pavlov**, **A. Goncearenco**, **S. De**, **A. G. Lada**, **E. Poliakov**, **A. R. Panchenko**, and **D. N. Cooper** (2018). Mutational signatures and mutable motifs in cancer genomes. *Briefings in Bioinformatics*, **19**(6), 1085–1101.

92. **Rotte, A.** (2019). Combination of CTLA-4 and PD-1 blockers for treatment of cancer. *Journal of Experimental & Clinical Cancer Research*, **38**(1), 255.

93. **Samet, J. M.** (1989). Radon and lung cancer. *JNCI: Journal of the National Cancer Institute*, **81**(10), 745–758.

94. **Samuels, Y.** and **T. Waldman** (2010). Oncogenic Mutations of PIK3CA in Human Cancers. *Current topics in microbiology and immunology*, **347**, 21–41. ISSN 0070-217X.

95. **Saunders, C. T.**, **W. S. Wong**, **S. Swamy**, **J. Becq**, **L. J. Murray**, and **R. K. Cheetham** (2012). Strelka: Accurate somatic small-variant calling from sequenced tumor–normal sample pairs. *Bioinformatics*, **28**(14), 1811–1817.

96. **Schwarz, J. M.**, **C. Rödelsperger**, **M. Schuelke**, and **D. Seelow** (2010). Mutationtaster evaluates disease-causing potential of sequence alterations. *Nature Methods*, **7**(8), 575–576.

97. **Shihab, H. A.**, **J. Gough**, **D. N. Cooper**, **I. N. Day**, and **T. R. Gaunt** (2013). Predicting the functional consequences of cancer-associated amino acid substitutions. *Bioinformatics*, **29**(12), 1504–1510.

98. **Sim, N.-L.**, **P. Kumar**, **J. Hu**, **S. Henikoff**, **G. Schneider**, and **P. C. Ng** (2012). SIFT Web Server: Predicting effects of amino acid substitutions on proteins. *Nucleic Acids Research*, **40**(W1), W452–W457.

99. **Sjöblom, T.**, **S. Jones**, **L. D. Wood**, **D. W. Parsons**, **J. Lin**, **T. D. Barber**, **D. Mandelker**, **R. J. Leary**, **J. Ptak**, **N. Silliman**, *et al.* (2006). The consensus coding sequences of human breast and colorectal cancers. *Science*, **314**(5797), 268–274.

100. **Starita, L. M.**, **D. L. Young**, **M. Islam**, **J. O. Kitzman**, **J. Gullingsrud**, **R. J. Hause**, **D. M. Fowler**, **J. D. Parvin**, **J. Shendure**, and **S. Fields** (2015). Massively parallel functional analysis of BRCA1 RING domain variants. *Genetics*, **200**(2), 413–422.

101. **Stratton, M. R.**, **P. J. Campbell**, and **P. A. Futreal** (2009). The cancer genome. *Nature*, **458**(7239), 719–724.

102. **Subramanian, A.**, **P. Tamayo**, **V. K. Mootha**, **S. Mukherjee**, **B. L. Ebert**, **M. A. Gillette**, **A. Paulovich**, **S. L. Pomeroy**, **T. R. Golub**, **E. S. Lander**, *et al.* (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(43), 15545–15550.

103. **Tamborero, D.**, **A. Gonzalez-Perez**, and **N. Lopez-Bigas** (2013). OncodriveCLUST: Exploiting the positional clustering of somatic mutations to identify cancer genes. *Bioinformatics*, **29**(18), 2238–2244.

104. **Tamborero, D.**, **C. Rubio-Perez**, **J. Deu-Pons**, **M. P. Schroeder**, **A. Vivancos**, **A. Rovira**, **I. Tusquets**, **J. Albanell**, **J. Rodon**, **J. Tabernero**, *et al.* (2018). Can-

cer genome interpreter annotates the biological and clinical relevance of tumor alterations. *Genome Medicine*, **10**(1), 25.

105. **Tate, J. G.**, **S. Bamford**, **H. C. Jubb**, **Z. Sondka**, **D. M. Beare**, **N. Bindal**, **H. Boutselakis**, **C. G. Cole**, **C. Creatore**, **E. Dawson**, *et al.* (2019). COSMIC: The catalogue of somatic mutations in cancer. *Nucleic Acids Research*, **47**(D1), D941–D947.

106. **Thurman, R. E.**, **E. Rynes**, **R. Humbert**, **J. Vierstra**, **M. T. Maurano**, **E. Haugen**, **N. C. Sheffield**, **A. B. Stergachis**, **H. Wang**, **B. Vernot**, *et al.* (2012). The accessible chromatin landscape of the human genome. *Nature*, **489**(7414), 75–82.

107. **Tokheim, C.** and **R. Karchin** (2019). CHASMplus reveals the scope of somatic missense mutations driving human cancers. *Cell Systems*, **9**(1), 9–23.

108. **Tu, Q.**, **J. Hao**, **X. Zhou**, **L. Yan**, **H. Dai**, **B. Sun**, **D. Yang**, **S. An**, **L. Lv**, **B. Jiao**, *et al.* (2018). CDKN2B deletion is essential for pancreatic cancer development instead of unmeaningful co-deletion due to juxtaposition to cdkn2a. *Oncogene*, **37**(1), 128–138.

109. **VanderPlas, J.**, *Python Data Science Handbook: Essential Tools for Working With Data*. O'Reilly Media, Inc., 2016, 1st edition. ISBN 1491912057.

110. **Vandin, F.**, **E. Upfal**, and **B. J. Raphael** (2011). Algorithms for detecting significantly mutated pathways in cancer. *Journal of Computational Biology*, **18**(3), 507–522.

111. **Vogelstein, B.**, **N. Papadopoulos**, **V. E. Velculescu**, **S. Zhou**, **L. A. Diaz**, and **K. W. Kinzler** (2013). Cancer genome landscapes. *Science*, **339**(6127), 1546–1558. ISSN 0036-8075.

112. **Wang, K.**, **M. Li**, and **H. Hakonarson** (2010). ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research*, **38**(16), e164–e164. ISSN 0305-1048.

113. **Weinstein, J. N.**, **E. A. Collisson**, **G. B. Mills**, **K. R. M. Shaw**, **B. A. Ozenberger**, **K. Ellrott**, **I. Shmulevich**, **C. Sander**, **J. M. Stuart**, **C. G. A. R. Network**, *et al.*

(2013). The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics*, **45**(10), 1113.

114. **Wendl, M. C.**, **J. W. Wallis**, **L. Lin**, **C. Kandoth**, **E. R. Mardis**, **R. K. Wilson**, and **L. Ding** (2011). PathScan: A tool for discerning mutational significance in groups of putative cancer genes. *Bioinformatics*, **27**(12), 1595–1602.

115. **Wiegand, K. C.**, **S. P. Shah**, **O. M. Al-Agha**, **Y. Zhao**, **K. Tse**, **T. Zeng**, **J. Senz**, **M. K. McConechy**, **M. S. Anglesio**, **S. E. Kalloger**, *et al.* (2010). ARID1A mutations in endometriosis-associated ovarian carcinomas. *New England Journal of Medicine*, **363**(16), 1532–1543.

116. **Wilson, D. L.** (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3), 408–421.

117. **Wong, W. C.**, **D. Kim**, **H. Carter**, **M. Diekhans**, **M. C. Ryan**, and **R. Karchin** (2011). CHASM and SNVBox: Toolkit for detecting biologically important single nucleotide mutations in cancer. *Bioinformatics*, **27**(15), 2147–2148.

118. **Yang, H.**, **D. Ye**, **K.-L. Guan**, and **Y. Xiong** (2012). IDH1 and IDH2 mutations in tumorigenesis: Mechanistic insights and clinical perspectives.

119. **Yui, S.**, **S. Kurosawa**, **H. Yamaguchi**, **H. Kanamori**, **T. Ueki**, **N. Uoshima**, **I. Mizuno**, **K. Shono**, **K. Usuki**, **S. Chiba**, *et al.* (2017). D816 mutation of the KIT gene in core binding factor acute myeloid leukemia is associated with poorer prognosis than other KIT gene mutations. *Annals of Hematology*, **96**(10), 1641–1652.

120. **Zhang, J.**, **J. Baran**, **A. Cros**, **J. M. Guberman**, **S. Haider**, **J. Hsu**, **Y. Liang**, **E. Rivkin**, **J. Wang**, **B. Whitty**, *et al.* (2011). International Cancer Genome Consortium Data Portal—a one-stop shop for cancer genomics data. *Database*, **2011**.

121. **Zhao, C.**, **S. Li**, **M. Zhao**, **H. Zhu**, and **X. Zhu** (2018). Prognostic values of dna mismatch repair genes in ovarian cancer patients treated with platinum-based chemotherapy. *Archives of Gynecology and Obstetrics*, **297**(1), 153–159.

122. **Zhao, Z.** *et al.* (2002). Neighboring-nucleotide effects on single nucleotide polymorphisms: A study of 2.6 million polymorphisms across the human genome. *Genome Research*, **12**(11), 1679–1686.

123. **Zhou, W.**, **T. Chen**, **Z. Chong**, **M. A. Rohrdanz**, **J. M. Melott**, **C. Wakefield**, **J. Zeng**, **J. N. Weinstein**, **F. Meric-Bernstam**, **G. B. Mills**, *et al.* (2015). TransVar: A multilevel variant annotator for precision genomics. *Nature Methods*, **12**(11), 1002–1003.

124. **Zhu, W.**, **S. Wu**, and **Y. A. Hannun** (2017*a*). Contributions of the intrinsic mutation process to cancer mutation and risk burdens. *EBioMedicine*, **24**, 5–6.

125. **Zhu, Y.**, **T. Neeman**, **V. B. Yap**, and **G. A. Huttley** (2017*b*). Statistical methods for identifying sequence motifs affecting point mutations. *Genetics*, **205**(2), 843–856.

GENERAL TEST COMMITTEE

CHAIRPERSON:	Dr. Amal Kanti Bera Professor Dept. of Biotechnology
GUIDE(S):	Dr. Karthik Raman Associate Professor Dept. of Biotechnology
	Dr. Balaraman Ravindran Professor Dept. of Computer Science and Engineering
MEMBERS:	Dr. Himanshu Sinha (Departmental member) Associate Professor Dept. of Biotechnology
	Dr. Swagatika Sahoo (member from other Department/ Institution) INSPIRE Fellow Dept. of Chemical Engineering