University of Glasgow | School of Computing Science

Topic Modelling for identifying signatures of mutational processes in cancer and virus genomes

Bipin Steephen

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

2nd September 2022

Abstract

The genomes inside somatic cells of human body are constantly exposed to different intrinsic and extrinsic mutagenic processes. Contributions from each of these mutagenic processes are different, but over the course of time they lead to increased variations in the genetic code and often leads to cancers. Analysis of different mutational signatures in genomes from different cancer samples allow one to examine how mutational processes such as aging, exposure to sunlight and smoking work. In this project, we have developed a novel framework to perform experiments to identify, quantify and evaluate common mutational processes and their activities using Latent Dirichlet Allocation Topic Modelling technique using cancer genomes datasets. Results shows that our method confirm many expected results, provide solutions to some signature extraction challenges, and provide an easy to use, scalable platform to conduct experiments with reproducible results.

Viruses inside hosts body are also subject to mutations. However, some mutagenic process such as aging are not relevant to virus genome mutations since duration of infection is comparatively shorter. Study of mutations in virus genomes inside human body could shed light to the signatures of attack and defence of human immune system. Promising performance of the developed framework evaluated with cancer genome data, together with the availability of publicly available genomes of different human viruses motivated the extraction and study of signatures of virus genome mutations in humans.

The study uses Latent Dirichlet Allocation, which is a probabilistic unsupervised machine learning method for Topic Modelling. Topic Modelling is often used to extract topics from documents in Natural Language Processing (NLP). In this project, Topic Modelling is employed to extract signatures of mutational processes from genome samples, which is a non-NLP task.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form. **Please note that you are under no obligation to sign this declaration but doing so would help future students.**

Name: Bipin Steephen Signature: Bipin Steephen

Acknowledgements

I would like to express my sincere gratitude to my project supervisor Dr. Ke Yuan for providing opportunity to work on such a relevant and interesting topic and thank him for the guidance and support provided throughout the project. I would also like to thank Mr. Kieran Lamb for sharing valuable dataset of SARS-CoV-2 virus mutations catalog which has helped a lot in this project.

Contents

Chapter 1 Introduction

In this project, publicly available mutational catalogue datasets are used to extract genome mutational signatures found in cancer and SARS-CoV-2 virus samples. An understanding of some basic concepts of molecular biology is essential to understand the data and interpret the results. Below sections provide a brief overview of some basic concepts and motivation of the project.

1.1 Cancer genome mutations

A human's life begins as a single cell. This single cell divides to give rise to two identical cells. Further cell divisions generate approximately 37.2 trillion cells in a human body at different locations with different purposes and expressed at different times. It is wonderful to appreciate the original single cell which contained the targeted instructions on how and when to behave towards each cell in different generations. DNA, or deoxyribonucleic acid, inside the nucleus of every cell in the body is the genetic material in humans and most of the living organisms [1]. During the cell division, DNA in the nucleus of the parent cell gets replicated and identical copies of original DNA are stored in both the daughter cells. The DNA contains instructions for each cell on how and when to behave. If DNAs are replicated identically, the DNA code in all the cells of our body would have been the same. However, during DNA replication, errors can sometimes occur. DNA is a polymer, and it is made up of monomers called nucleotides. The two strands of DNA (identified by 5" end and 3" end) are made up of sugar-phosphate and the strands are linked by hydrogen bond between nucleotide bases. There are 4 types of nucleotides in the DNA. They are differentiated based on the base component of nucleotide. The four bases are – Adenine (A), Thymine (T), Guanine (G) and Cytosine (C). The bases chemically bond together between two strands of DNA to appear as pairs in the DNA. Due to the nature of these hydrogen bonds, Adenine and Thymine always pair together (A-T), and Guanine always pair together with Cytosine (G-C). Human DNA contains almost 3.2 billion of base pairs and are stored in 23 pairs of chromosomes. The chromosomes contain genes, which are region of DNA encoded by sequences of bases to have a function. In humans, it is estimated that around 20k-25k genes are there. There are protein encoding genes and non-encoding genes which have different functions. As per the central dogma [2], when a gene is expressed, the sequence of DNA as converted into sequence of RNA by the process called transcription and RNA produce proteins by the process called translation. The category of protein encoding genes contain code for the proteins to be made. The cells do not turn on all the genes at all the times. It is controlled by a Gene expression mechanism, which activate or deactivate required genes. This is happening throughout the life for different biological processes such as the growth, development of organs, immunity etc.

Coming back to the bases in nucleotides (A-T, & G-C), which are the alphabet of DNA. Alterations can occur in the base pairs, and they are called mutations. Mutations happening in the genetic code of reproductive cells (egg and sperm) are called germline mutations and they are carried to downstream generations.

Mutations occurring in the cells other than egg and sperm are called somatic mutations. Somatic mutations are not passed to the next generations. This project focus only on somatic mutations. Sequence of 3 letters in DNA is called codons (words in genetic language). For the case of protein encoding genes, each codon corresponds to an amino acid or stop codon. 4x4x4 = 64 codons combinations are possible, and a codon table [3] maps the codon with corresponding amino acid. Amino acids link together to form proteins, which are the building blocks of our body. When a cell expresses a gene containing mutations, since the codons are altered, it could lead to generation of different amino acids and results in different proteins. Many mutations are harmless since they produce the same amino acid. However, the mutations can be passed to the next generation of cells and over time, excessive growth can occur in the region due to the presence of wrong proteins. The different categories of mutations are - substitution, insertion, deletion, translocation, and inversion [4]. This study mainly focusses on Single Base Substitution (SBS) which is a type of point mutation. There are many reasons for mutations including intrinsic processes such as aging and exposure to extrinsic mutagenic processes such as exposure to photons in UV light, smoking, exposure to some chemicals etc. Studies [5] showed that Cytosine to Adenine mutation is highly prevalent in TP53 genes in due to smoking mutational process by analyzing lung cancer samples. When C>A mutation occurs in one strand of DNA, Guanine will be converted into Thymine in the opposite strand due to the complementary nature of double helix. Studies further shown that UV radiation leads to predominant C>T mutations [6]. C>T substitution in one strand simultaneously causes G>A substitution on the complementary strand and we cannot distinguish between them. Each base could be mutated to any of other three bases. So, 12 substitutions are possible. However, since complementary substitution is indistinguishable, in line with other similar studies [7]–[14]. we have considered the following 6 SBS mutations: {C>A, C>G, C>T, T>A, T>C, T>G}. Additionally, the context around the mutation is also important. These studies considered one or two adjacent nucleotides to categorise mutation type. In this study one adjacent nucleotide on both sides of SBS mutation is considered. 4 combinations of bases are possible in each position, leads to 96 combinations(4x6x4). Example of a mutation type format considered in this project is: A[C>T]G which translates to the C>T substitution in a trinucleotide ACT. A typical cancer whole genome sequence sample would have many such mutation types with different quantities. A closer look at the cancer sample indicates the analogy of mutation types in cancer sample with words in documents.

- A[C>T]G *is a* mutation type *analogous to* a word
- Cancer sample *is* a collection of mutation types *like* a collection of words analogous to a document
- Mutation catalog *is* a collection of cancer samples *like* a collection of documents *analogous to* a corpus

Figure 1: Analogy between mutational signatures and NLP

Somatic mutations in cancer genomes are the result of cumulative effect of multiple mutational processes with different strengths. Decomposing individual mutation signatures from cumulative mutation load can provide insights into different underlying mutational processes [15]. Mutation profile of a cancer sample is visualized in in Figure 2. Since the mutation profile is caused by multiple mechanisms, we can say that this mutation profile carries different topics called signatures. This analogy with language offer possibility to apply NLP techniques like topic modelling to extract signatures. Topic modelling is an unsupervised machine learning technique when trained with documents of different topics say sports and politics, can detect the patterns of terms frequently occur in each topic, and identify the groups and calculate the strengths of constituting topics. LDA or Latent Dirichlet Allocation [16] is one of the mostly used topic modelling based on generative probabilistic method. This study aims to extract signatures using LDA and compare with COSMIC v3.0 reference signatures available in COSMIC website [17].

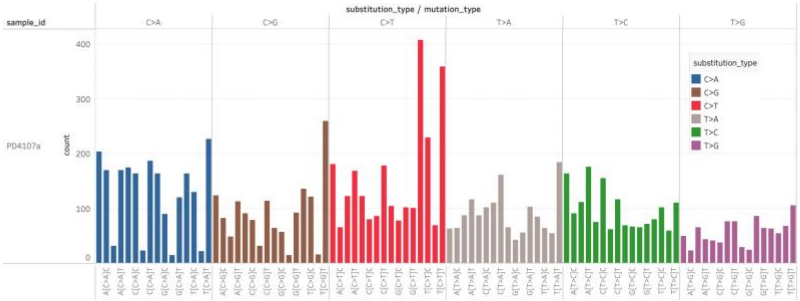


Figure 2: Visualization of mutation catalog of a breast cancer sample

1.2 Virus genome mutations

Viruses in hosts body are also subject to genetic mutations. Genetic material in different virus types may be different (DNA or single stranded RNA or double stranded RNA etc.). The mutational processes in cancer genomes are often vivid and of long duration. However, since viruses depends on host cells to reproduce and often the duration of infection is comparatively shorter, the influence of some mutational processes like are aging assumed to be irrelevant. The mutations in the virus genomes possibly hold the immune response of the host. Analysis of publicly available genome mutation data of different viruses to identify common patterns could provide new pathways to reveal hidden secrets of our immune system. This project focuses on genetic mutations in SARS-CoV-2 virus (single stranded RNA virus) due to the availability of publicly available datasets (GISAID [18]). One significant difference with cancer genome signatures is the number of mutation types. For the case of SARS-Cov-2, 4x12x4 = 192 mutation types are possible. This is double the number in cancer genomes since the mutation in the complementary strand is not there for the case of RNA virus. The experiment framework used to extract cancer genome signatures is used to further extract virus genome signatures. The study can be further extended to other viruses like Influenza viruses, Zika viruses etc.

1.3 Experiment framework

Another motivation of this project is to develop an experiment framework for signature extractions. Since a probabilistic method is used, multiple rounds of experiments and evaluations are often necessary. Experiments often need high iterations and passes through the documents and texts, and therefore could lead to longer execution times. Additionally, experiments on multiple cancer classifications or sub classification samples should be possible. Therefore, for an easy to use, scalable, platform to generate stable topics, evaluate and quantify the signatures, and provide reproducible results is required.

Chapter 2 Background

The completion of human genome project [19] and generation of somatic mutation catalogs on global scale by cancer genome project [20] enabled analysis of cancer genomes mutations. Stratton et al in 2011 [15] argued that specific mutational signatures are associated with every mutational process and accumulation of mutations over time are combinations of mutational processes. Nik-Zainal et al. [7], [12] predicted mutational signatures using cancer mutation catalogs. Alexandrov et al. [8], [10] utilized a method using Non-Negative Matrix Factorization (NMF) [21] to extract 30 distinct signatures. A more comprehensive study using more mutation catalog datasets are conducted and more than 49 signatures are extracted by Alexandrov et al [14]. The existence of mutational signatures is experimentally supported with the reproduction of mutational signatures using CRISPR-Cas9 technology by Zou et al. [22]. Many studies [7]–[10], [12], [13] used NMF based methods to extract mutational signatures from cancer mutation catalogs. Shiraishi et al. in 2015 [11] and Matsutani et al. in 2019 [23] used topic modelling based on Latent Dirichlet Allocation [16]. Shairashi et all [11] used EM algorithm to maximize likelihood. Matsutani et al [23]employed LDA with Variational Bayes Inference. Too few samples from whole genome sequences were a challenge in this study. Fantini D et al [24] developed an R based package using NMF method to extract mutational signatures.

Graudenzi et al. [25] studied intra host genomic diversity of SARS-Cov-2 virus with 1133 samples and extracted three non-overlapping signatures using NMF methods. In this project, we use mutation catalog dataset from 138384 samples and employ LDA based method.

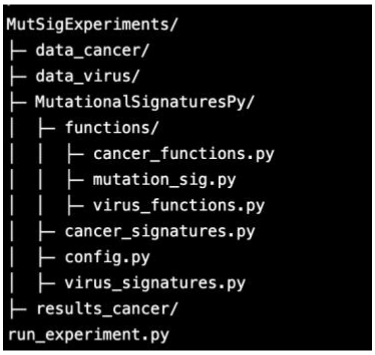
Many of the previous works on cancer mutational signatures are evaluated against the 30 mutational signatures from the study of Alexandrov et al [8] and which are shared in COSMIC database [17]. To my best of knowledge, no published papers found which compare against the more recent work of Alexandrov et al, 2020 [14] with extracted 49+ mutational signatures using a much larger mutation catalog dataset. Application of LDA based topic modelling for signature extraction from SARS-CoV-2 is a novel method to my best of knowledge. Further, another motivation of this study - to compare the mutational signatures from different human viruses to reveal patterns of human immune response is a novel one. Additionally, a python-based framework using LDA to conduct experiments for mutational signatures for cancer and virus genomes is also a new approach.

Chapter 3 Implementation

3.1 Experiment framework

A computational experiment framework called – MutSigExperiments is designed and developed for this project. Implementation is done using python code (version 3.9.7 64bit) using Visual Studio Code (version 1.54.3) as source code editor on a MacBook Air (M1, 2020). The source code of MutSigExperiments is uploaded on git - https://stgit.dcs.gla.ac.uk/2559542s/MutSigExperiments.

Refer Appendix A for step-by-step instructions to download the framework and get started.



```
MutSigExperiments/
├─ data_cancer/
├─ data_virus/
├─ MutationalSignaturesPy/
│  ├─ functions/
│  │  ├─ cancer_functions.py
│  │  ├─ mutation_sig.py
│  │  ├─ virus_functions.py
│  ├─ cancer_signatures.py
│  ├─ config.py
│  ├─ virus_signatures.py
├─ results_cancer/
run_experiment.py
```

Figure 3: Folder structure of source code

Functional and non-functional requirements considered for the development of framework is given in Appendix B. The framework supports signature extraction from different cancer mutations and virus mutations. Folder - MutationalSignaturePy is consists of .py files containing functions for extraction cancer and virus signature extraction workflows. These functions are the core of this framework. A config.py file is maintained to store the values for the configurable parameters. The helper functions are stored inside /functions. Common helper functions are kept inside mutation_sig.py. The downloaded and preprocessed files are stored under corresponding /data_ folders. Results of workflows are stored under corresponding /results_ folders with the experiment name given. Relevant information about the execution and results are stored in log files under /results_ folder. Trained models, grid search results, extracted signatures, signature probability matrices and visualizations are categorized and stored in the file system under /results_ folder. The code is expected to run from the main directory – /MutSigExperiments. Sample run_exeperiment.py files to call the workflows are shared under - /MutSigExperiments.

```
# Change directory
cd /MutSigExperiments

# Run experiment
MutSigExperiments> python run_experiment.py
```

Figure 4: Sample code to run experiments

Python file run_experiment.py will have methods to call the required workflows for – download data, preprocess data, perform grid search, train multiple models, combine signatures, ensemble training with their associated parameters. Refer to the sample file shared to get started.

3.2 Data

The dataset of cancer mutations catalogs used is for this study is derived from 4645 cancer samples from whole genome sequencing (WGS) and 19184 cancer samples from whole exome sequencing (WES). The mutation catalog is the same used by COSMIC v3.0 by Alexandrov et al, 2020 [14]. The dataset is publicly available to be downloaded in https://www.synapse.org/#!Synapse:syn11726616. Dataset consists of .csv files with count of mutations for each mutation type and trinucleotide for each sample. Sample names include the cancer classification and sub classification. In this study, we focused on mutation type dictionary of 96 types since only point mutations of type Single Base Substitutions (SBS) in unstranded trinucleotide context is considered. Framework supports downloading of 192, 1536 contexts, Double Base Substitutions (DBS), indels (insertion, deletions).

- Mutation type: CA
- Trinucleotide: TCG
- Sample name: Breast-AdenoCA::SP2293
- Count: 101

The sample – SP2293 of cancer classification Breast-AdenoCA found to have mutation type - T[C>A]G for 101 times.

Figure 5: How to read a data in mutation catalog

Upon invoking run_cancer_download() function, mutation catalog from the four sources and reference COSMIC v3.0 signatures are downloaded under /data_cancer directory. Data is downloaded from synapse website and therefore credentials are required for download. Credentials can be stored in config.py file.

To apply Topic Modeling algorithm, the mutation catalog needs to be converted into documents with words of mutation types. Example for 96 possible words (96 mutation type) are – C[C>A]T, T[C>T]C etc. In a particular sample, there will be many mutation types and many mutation types occur multiple times. So preprocessing is used to convert the downloaded mutational catalog into a word-document format. Each record corresponds to a sample. Text consists of all the mutation types and each type stored as many numbers of times as it is found in the sample.

- Mutation type: CA, Trinucleotide: TCG, Count: 3
- Mutation type: CT, Trinucleotide: ACT, Count: 2

Preprocessed text: T[C>A]G T[C>A]G T[C>A]G A[C>T]T A[C>T]T

Figure 6: Preprocessed text formation

Additionally, cancer classification, subclassification, number of mutations in the sample and source is included in the preprocessed data. Preprocessed csv file is stored under /data_ folder.

3.3 Method

3.3.1 Latent Dirichlet Allocation

This project uses genism implementation of Latent Dirichlet Allocation algorithm [26] for Topic Modelling. Gensim package version 4.2.0 is used. Gensim package needs to be installed in the machine to run the code. LDA is a generative probabilistic method. LDA consider documents as multinomial probability distributions of topics and topics as multinomial probability distributions of words. In LDA, the latent variable called topic is used to generate each word in a document. Model learning happens with term-document matrix decompose into product of (topic, words) and (document, topics) matrices [16].

The major hyperparameters in LDA are number of topics(K), alpha(α) and beta(β). Alpha represents the distribution of topics per document. Small alpha means fewer topics per document. Beta represents the distribution of terms within a topic. Both alpha and beta are parameters of prior distribution. Gensim LDA implementation uses Variational Bayes Inference method to learn the parameters. Another option is Gibbs sampling and another LDA implementation Mallet uses this method [27]. In Variational Bayer Inference method, Variational Lower Bound (also called Evidence Lower Bound) is tried to be maximized by minimizing the KL Divergence [16].

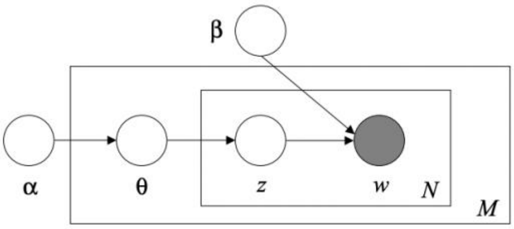


Figure 7: Graphical model representation of LDA [16]

Since LDA is an unsupervised method, one common challenge is to calculate the number of topics (K). One commonly adopted solution to this challenge is to use Hierarchical Dirichlet Process [28]which can infer the number of topics. However, the number of topics in this project are the mutational signatures. The number of mutational signatures falls in a finite range. Therefore, an approach of using LDA with grid search to maximize Variational Lower Bound to identify the number of topics is employed. Usually grid search is performed in machine learning with the aim to maximize or minimize the target evaluation measure. Here evaluation metric selected is Variational Lower Bound (VLB). Other considered options are pairwise mean cosine similarity and pairwise median cosine similarity between the extracted signatures, and coherence score. Coherence score seems not a fit for this study since it is a measure of semantic similarity, and we deal with Non-Natural Language dataset. Validity of using Mean and median cosine similarity metrics is not assessed for this project. On the other hand, VLB is a widely used intrinsic measure and have selected as the major evaluation measure for the grid search.

3.3.2 Grid search

The grid search experiments can be done by invoking run _gridsearch() function. The method can accept different parameters like source, cancer classification/sub classification, range of K, range of alpha, range of beta, minimum mutations, iterations, passes etc. for the experiments. The signatures are mainly extracted based on a major cancer classification (E.g., Breast) or a sub classification (E.g., Kidney-RCC). At the core, system call parallelized LDA implementation - ldamulticore [26] to train the model and calculate VLB at different K to choose the best value for number of topics. Then another grid search is performed using across grid of alpha and beta parameters to find the best hyperparameter values. Results are stored as .csv file and graph of evaluation measure is stored as .png files. Multiple grid searches are possible with different iterations and passes to ensure that the model parameters are converged. System is designed to choose the best values of hyperparameters automatically based on the selected evaluation measure, which can be controlled in config.py. Additionally, the grid search of K, alpha, and beta can be performed together, which could be better at accuracy at the cost of running time. This functionality is available in the framework, however, in this study, two step grid search method is used.

It is worth to note that although hyperparameters are tuned, the model parameters may not be necessarily tuned. Therefore, performing the grid search with high iterations and passes are often required for the model to converge. Passes indicate the number of epochs. It controls the number of times the training is done on the entire corpus. Iterations corresponds to the maximum number of iterations through the corpus during model training [26]. Higher values often correspond to better chance to converge at the cost of training time.

3.3.3 Training with tuned hyper parameters

The results of grid search are maintained in a .csv file, which will be referred by further modules to train models with tuned hyperparameters by bootstrapping and ensemble method. Hyperparameter values for the best grid search will be retrieved from the grid results csv. Option is given to bypass this and manually enter K, alpha, and beta. It is worthy to note that LDA need the corpus, dictionary, and tokens as inputs. Tokens are generated from the text representation of mutation catalog. Dictionary is generated using built in method in Gensim and corpus is generated using bag-of-word representation of the documents. Training is done using tuned hyperparameters using ldamulticore. Typically, the trainings with high iterations and passes are necessary, and it could take longer duration to execute. Therefore, trained models are stored in the /results_ and pretrained models can be loaded for further analysis if required.

Topics are extracted from the trained model and stored in csv file. Each row in the signature file corresponds to the probability of corresponding one out of 96 mutation type (word) in the topic. Multiple topics are stored as different columns. The signatures are visualized as bar chart with different colors for substitution type and stored as .png file. Topic probabilities are calculated across all the documents based on the model. Each row in topic probability matrix corresponds to a document and different topics are given in columns. The matrix provides probability of each topic to appear in each of the documents.

3.3.4 Bootstrapping

One important aspect of topic modelling is stability. The topics generated from a model could be very different from another model even with a small change in the hyperparameter/model parameter. A possible solution to improve stability is bootstrapping. Model training with tuned hyperparameters is performed multiple times by changing the random seed. The number of models to be generated can be specified while calling the tuned_train() method. Instead of one, many models will be generated. Results of each training will be stored in separate directories with its corresponding model files, signatures, topic probabilities, logs etc.

Next step is to combine the signatures from all the models. If the number of topics is K and number of models generated during the tuned training is M, then a total of KxM topics will be generated. For the case of breast cancer, K found to be 14 and N taken as 16. Then 16x14 = 224 topics will be generated. Stable topics needs to be identified among these. Each topic is a vector of 96 dimensions for cancer mutations. (For virus mutations, it is 192). Inter-topic distance needs to be computed and similar topics can be aggregated together. One option is to calculate the pairwise cosine similarity between the topics. The pairwise similarity matrix will be of shape KxM rows, KxM columns. (In this example, 224x224). To group the similar topics, an algorithm based on cosine similarity is employed. As part of this algorithm, index of items is stored in a list when the cosine similarity value is greater than a threshold (E.g., 0.9) for each column of the matrix. For example, consider column 0, which have the cosine similarity between topic 0 with all the 224 topics in different rows. All the topics with more than threshold cosine similarity is grouped. The same is done for all the 224 topics in the columns. Then duplicates are removed from this list by converting to set. Then similar topics are combined by taking mean for each of 96 dimensions. This is analogous to identifying the geographic center of a cluster by calculating mean of the corresponding coordinates. The same is done for all the groups in the list. Now a reduced new set of topics are in hand. Pairwise cosine similarity is calculated, and the same technique is applied until no pairs have cosine similarity more than the given threshold or until a maximum number of iterations reached. This max number is a parameter and can be controlled by config.py. When the similar topics converge (all the pairwise cosine similarity is less than threshold), resultant topics are extracted and stored in disk as .csv file with each row corresponds to a mutation type and each column corresponds to a topic.

```
# Let there are K topics and M models
# Total generated signatures = KxM
# Initial cosine similarity matrix will be of shape (KxM rows, KxM
columns)

  For N iterations or until previous signature = current signature:

    Calculate pairwise cosine similarity of signatures
    Define an empty list C

    Iterate through each column:
      Define an empty list R

      Iterate through each row:
        If cosine similarity > threshold:
          Append index of item to R
```

```
   Append R into C

Remove duplicates from C

For each R in C:
   Calculate mean of each row
   # Mean of probabilities of each mutation type

Get signatures
```

Figure 8: Algorithm to combine signatures from multiple models.

Another possible method to combine the signatures is to use clustering. An attempt made as part of this project to employ K-means clustering to combine similar topics. However, due to time limitation, this is later discarded. This will be an interesting future direction. In the attempt, signatures are combined from different tuned trainings. K-means clustering elbow method is used to calculate the best number of topics based on distortion or inertia. After deciding on K, clustering again performed to get the final topics.

3.3.5 Ensemble

The package Genism contains an interesting option as a solution to stability of topics called Ensemble LDA [29]. Using this implementation, workflow is created to ensemble models. Invoking the run_ensemble() method with relevant parameters allows system to use multiple workers to make multiple models with LDA. Number of models is given as input to the method. Generally, when the same number of models used in the tuned train is used for Ensemble LDA, the time taken to perform the execution is found to be lesser with Ensemble LDA. This could be due to the effective utilization of multiple cores of CPU. Like tuned train, tuned hyperparameters are retrieved from grid search .csv file. Required directories are created based on the folder structure. The Ensemble LDA is trained, and model is saved into the disk for future use. Since the training often take lot of time with high iterations and passes, trained models could be good assets for future analysis. Therefore, a functionality to create ensemble model from pretrained model is also made available. This can potentially reduce the training time significantly. The Ensemble LDA model is converted into ldamodel using built in function. Extraction of topics, topic probabilities, are done as explained for tuned training.

To evaluate the performance, the extracted signatures are compared with the reference COSMIC signatures. In this study, the reference COSMIC signature is v3.0 and is available to be downloaded as publicly available dataset in https://cancer.sanger.ac.uk/signatures/downloads/. Pairwise Cosine similarity is used to compare and topics having higher than a threshold similarity value are highlighted.

3.3.6 Support analysis

In the previous section, it was mentioned that the probability of each topic occurring in each document is calculated and stored as document probability matrix. This is a useful matrix to calculate the importance (or support) of the topics. Two methods are identified to calculate the support of the topics. First method is based on presence and second method is based on the probability of presence. Algorithms are given below. For example, assume that out of 1000 documents, signature-1 have a probability of presence (regardless of the value of probability) in 500 documents, then support is calculated as 500/1000 = 0.5. This indicate that, whether in a small proportion or not, signature-1 has 50% probability of occurring in each document. This is a good measure to filter out topics appear as noise. However, one disadvantage of this support measure is that this ignores the probability of the topics to appear in the document. For example, a signature could be found on very less probability in all documents and therefore could be noise. Since in the first method, we are checking only if topic is present or not, we ignore the probability in this support calculation.

Therefore, another support measure based on signature probability is also employed. In this method, the mean of probabilities of a particular topic for all the documents is calculated. Interpretation of this support is as follows. If the support is calculated as 0.25, this means that in each document, there is 25% probability that signature-1 is present. Also, the sum of this support of all the topics will be equal to or around 1. This is a good measure because it quantifies the probability of topics contribution in each document. One disadvantage is that the probability is divided among topics and when number of topics increases, support value gets reduced. But importance of the signature could be still high.

Therefore, the system supports the calculation of both the support measures. The calculated support values are stored along with the signatures as separate rows at the end. This is to quickly analyze and compare the support of all the signatures.

- Support-presence: Probability of given topic's presence in the document.
  High values indicate that the topic will be present in most of the documents (regardless of small or large probability) without considering other topics
- Support-probability: Probability of the topic's contribution in the document.
  High values indicate higher probability that the topic to occur in each document.

Figure 9: Support measures calculated for measuring signature stability.

```
Take a trained model:

  Generate document-topic probability matrix

  Iterate through topics:

    C = count (documents present corresponding to given topic)
    S = sum (topics probability across all documents)

    Support presence = C/document length
    Support probability = S/document length
```

Figure 10: Algorithm for calculation of support measure

3.4 Virus Mutational Signatures

The SARS-CoV-2 mutation catalog dataset used in this project contains 138384 samples of genomic mutations. The catalog is stored in Dropbox and can be downloaded programmatically by invoking download_covid_data() method. One important difference is that, in the case of covid data, there are 192 mutation types possible. This is because SARS-CoV-2 is a single stranded RNA virus [30]. Therefore, complementary mutation need not be considered as we did for cancer samples. So 4x12x4 = 192 combinations possible. Another thing to be noted is that, in the place of Thymine (T), Uracil (U) is present in RNA viruses like SARS-CoV-2 virus. However, to simplify the notation, notation T is used in place of U. It is to be noted that, unlike cancer signatures, the number of mutations is very less. This is because the covid virus DNA sequence length is much smaller (around 20k). Average mutation in a sample is calculated as low as 1.5 for the case of SARS-CoV-2 virus whereas it is much higher for the case of cancer samples. However, this study deals with the mutations occurring to SARS-CoV-2 virus in each week's timeframe. Therefore, preprocessing step is like that of cancer mutation catalog, which involves transforming the count of mutation types into texts of mutation types. However, mutations are aggregated for each week for SARS-CoV-2. Since the input data consists of 86 weeks data, the number of documents is 86. Rest of the processes are like cancer signature extraction. One difference is that similarity with COSMIC signatures is not calculated. For the case of SARS-CoV-2 weekly mutation catalog, the number of documents is less. Therefore, one challenge experienced is that the model was not converging easily. Due to this, trainings were done using higher iterations and passes. An analysis done by varying the number of iterations and passes, training the LDA and plotting the VLB. It is found that the VLB is increasing fast when increasing iterations and passes and reaching a plateau like region and increasing slowly afterwards. The same grid search of iterations is made available for cancer signatures as well. This is to help to get an idea on the range of iterations and passes because number of samples and mutations per sample is different across cancer classifications and different viruses.

Since this is an experimentation framework and multiple experiments are often needed to reach a conclusion, the framework helped to store the results in a well-organized directory structure. Additionally, log files are maintained throughout and links for the log files are made available in the console. In addition to get important information about the experiments' inputs and results, log files can be used to check the progress of the trainings.

Chapter 4 Experiments and Results

4.1 Cancer Mutational Signatures

Experiments conducted for mutation catalog of breast cancer samples. Data from both whole genome sequencing (WGS) and whole exome sequencing (WES) are considered and samples with mutations less than 100 are ignored for the experiment. This resulted in mutation catalog of 934 samples. 200 iterations and passes are used for the experiments and number of topics varied from 2 to 30. Maximum VLB is obtained at number of topics = 14, alpha = 0.31, beta = 0.31 as in Figure 11. Best value for K = 13 received with Median cosine similarity measure as in Figure 12.

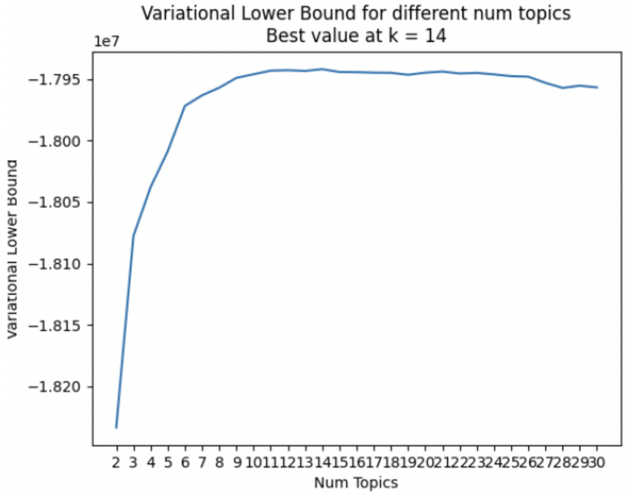


Figure 11: VLB for different values of K for Breast cancer signature experiments with iterations and passes = 200

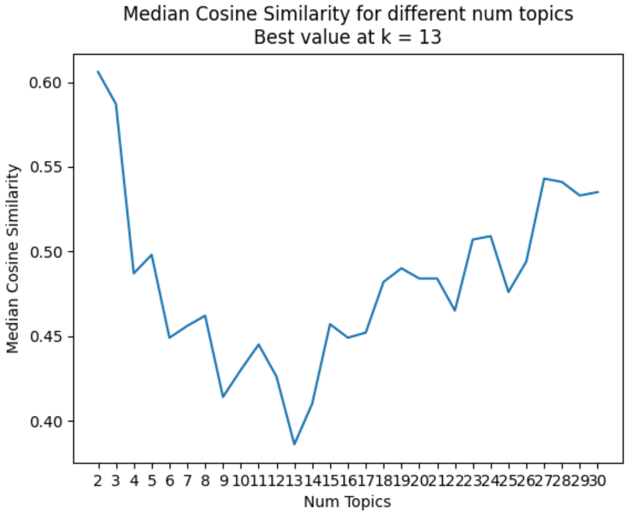


Figure 12: Median pairwise cosine similarity of extracted signatures for different values of K for Breast cancer signature experiments with iterations and passes = 200

The tuned hyper parameters are applied and LDA trained to generate 16 models. Signatures, and document topic probability matrix are extracted for each of them and compared with COSMIC v3.0 signatures using cosine similarity measure. Signatures with more than 0.8 cosine similarity are highlighted. Additionally, visualizations for extracted signature and document-topic probability are stored in file system. Figure 13 shows visualization of a document-signature probability matrix. Figure C.1 in Appendix shows visualization of a set of extracted signature before bootstrapping. Support measures are calculated for each of the extracted signatures. Support measure values generated for a training are given in Figure 14. The support probability measure turned out to be more useful in identifying noise signatures (6, 10 and 11 in this case).

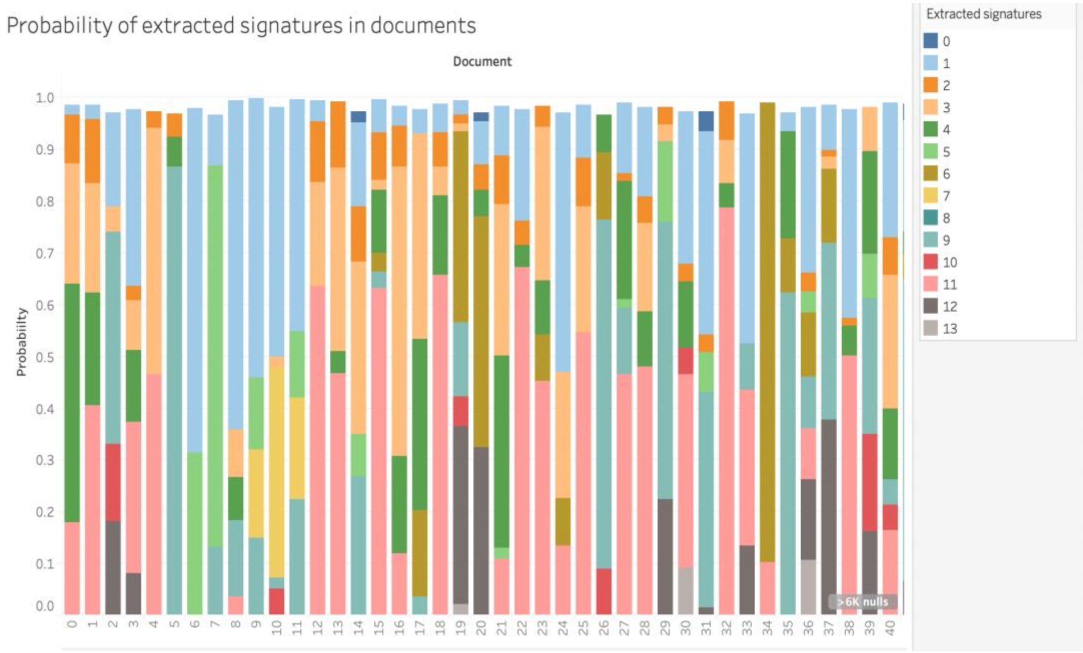


Figure 13: Visualization of document signature probability matrix generated using bootstrapped model trained with breast cancer samples.

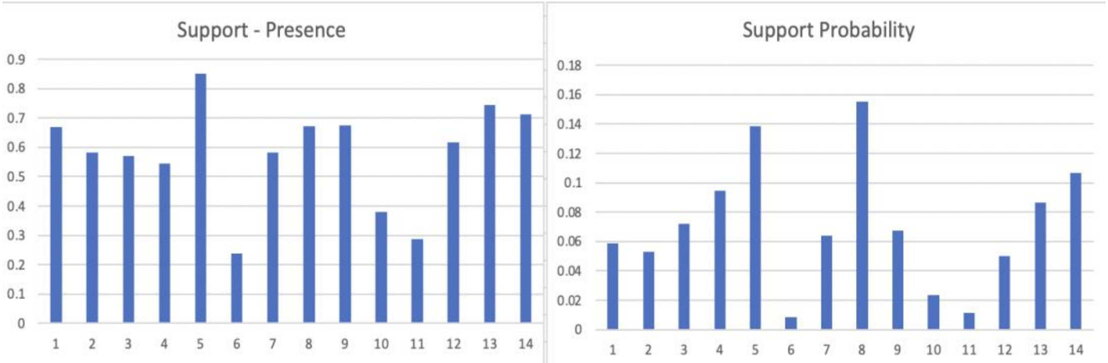


Figure 14: Left – Support presence, right – Support probability calculated for the signatures generated by a tuned training with breast cancer samples

The 14x16 = 224 signatures are bootstrapped using the algorithm specified in method section. Similar signatures are combined when cosine similarity between them is more than 0.9. On an average, the cosine similarity with COSMIC signatures found to be around 0.9 for matching signatures.

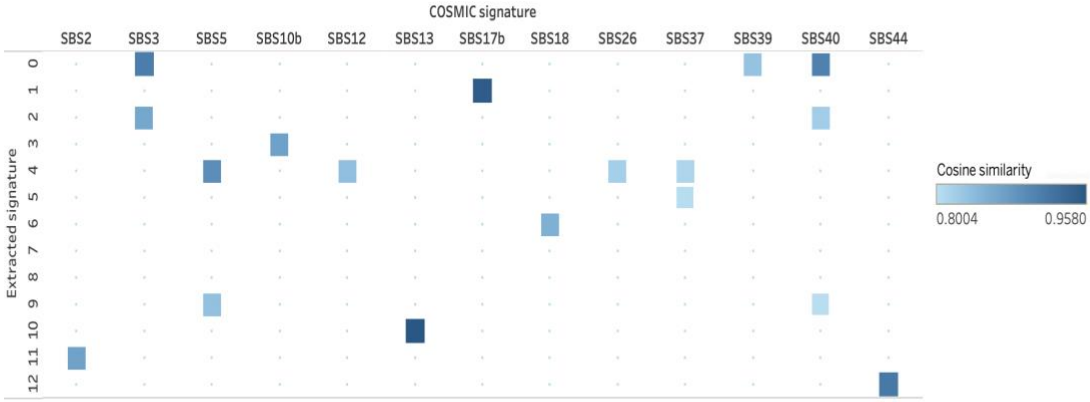


Figure 15: Extracted signatures (method – bootstrapping) from breast samples matching with COSMIC signatures (highlighted based on cosine similarity)

The experiment could extract 11 signatures matching with COSMIC signatures with average cosine similarity 0.86.

Additionally, a comparison with proportion of tumors with the signatures (available at https://www.nature.com/articles/s41586-020-1943-3/figures/3) from the study of Alexandrov et al, 2020 [14] shows that, in breast cancer samples, the signatures – SBS1, SBS2, SBS3, SBS8, SBS9, SBS13, SBS17A, SBS17B, SBS18, SBS37, SBS40 and SBS41 are more prominent. 8 out of 12 similar signatures are extracted in this experiment. Visualization for the extracted signatures is given in Appendix Figure C.2. Figure 16 shows visual comparison between a COSMIC signature and matching extracted signature in this experiment. SBS13 is associated with APOBEC activity and predominant in breast cancer [14] and is experimentally validated by Chan et al. 2015 [31].

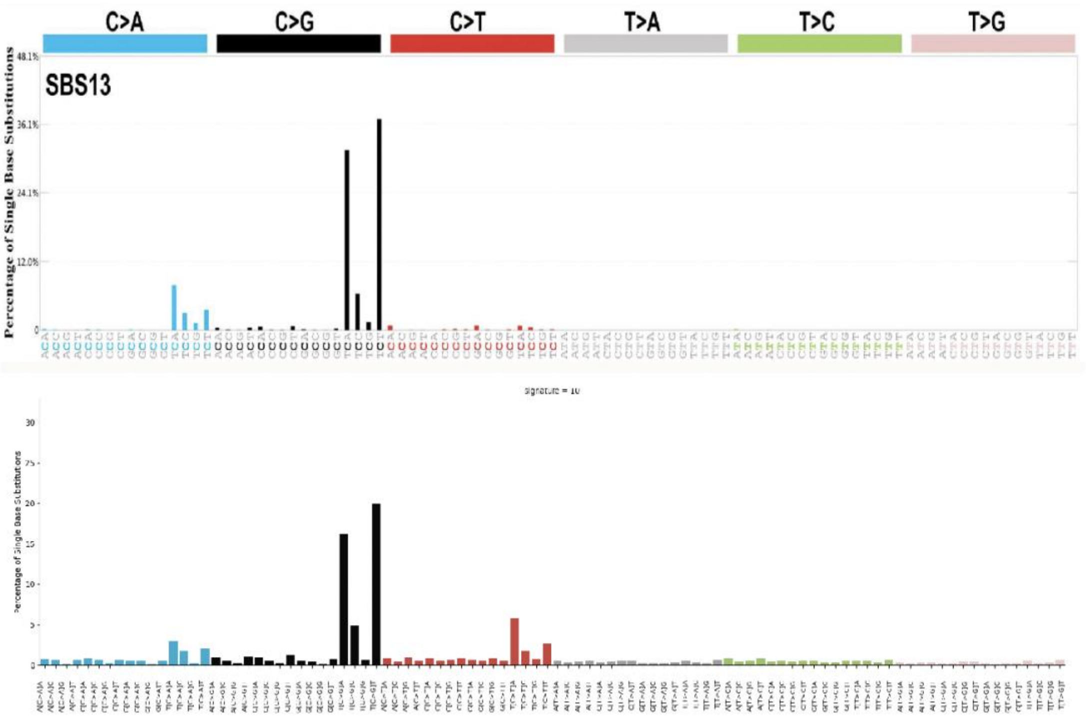
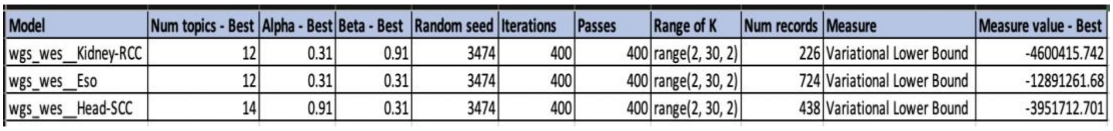


Figure 16: Top - COSMIC signature SBS13 [14]. Bottom - matching extracted signature in this experiment.

16 models are trained using ensemble LDA with same 200 iterations and passes. However, results were not promising. More experiments and analysis are needed with higher iterations and passes with breast cancer mutation catalogs.

Further experiments are conducted with higher iterations 400 on Head-SCC mutation data samples.



Model	Num topics - Best	Alpha - Best	Beta - Best	Random seed	Iterations	Passes	Range of K	Num records	Measure	Measure value - Best
wgs_wes__Kidney-RCC	12	0.31	0.91	3474	400	400	range(2, 30, 2)	226	Variational Lower Bound	-4600415.742
wgs_wes__Eso	12	0.31	0.31	3474	400	400	range(2, 30, 2)	724	Variational Lower Bound	-12891261.68
wgs_wes__Head-SCC	14	0.91	0.31	3474	400	400	range(2, 30, 2)	438	Variational Lower Bound	-3951712.701

Figure 17: Grid search results obtained for experiments of Kidney-SCC, Eso and Head-SCC

The results of ensemble training for Head-SCC provided 8 stable signatures which have strong correlation with the reference signatures - SBS1, SBS2, SBS3, SBS4, SBS5, SBS7a, SBS7b and SBS 40. Extracted signatures are given in Appendix Figure C.3. However, some signatures matched with more than one reference signatures as evident in Figure 18. This points to need for further experiments.

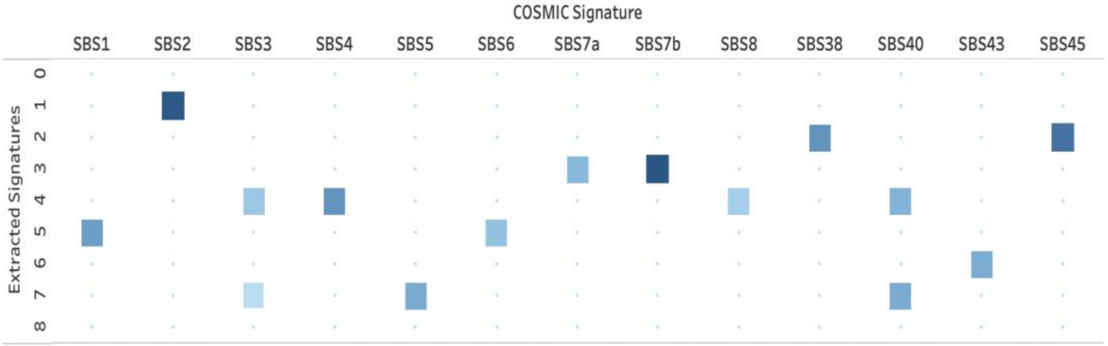


Figure 18: Extracted signatures (method – ensemble) from Head-SCC samples matching with COSMIC signatures (highlighted based on cosine similarity)

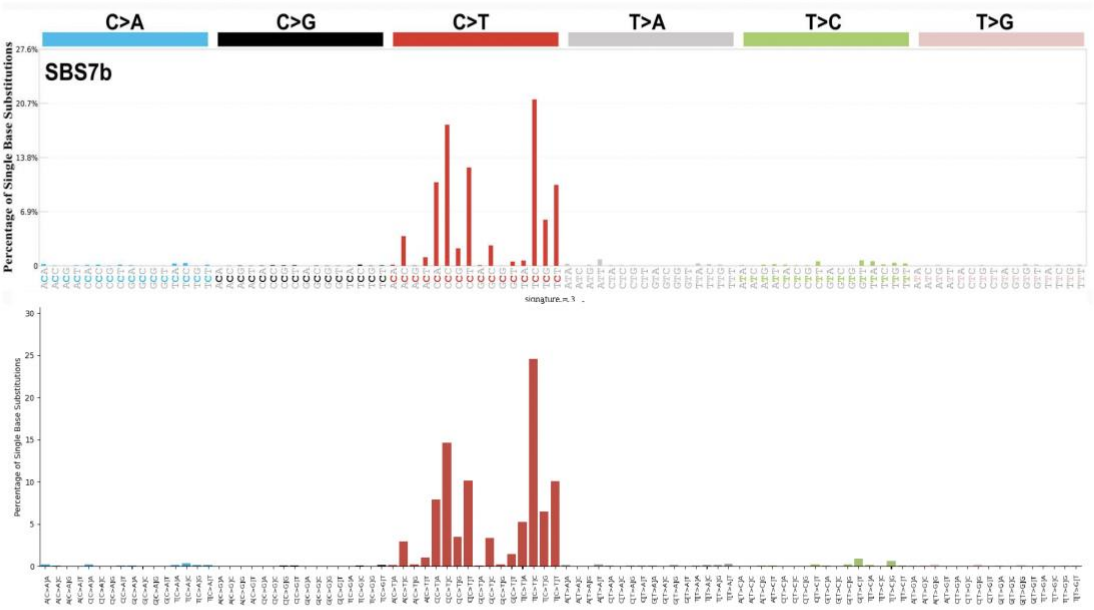


Figure 19: Top - COSMIC signature SBS7b [14]. Bottom - matching extracted signature in this experiment.

Figure 19 shows the comparison between SBS7b and extracted signature in Head Squamous Cell Carcinoma samples. SBS7b is associated with UV light exposure mutational process [14] and is experimentally validated Nik-Zainal et al. 2015 [32]. C>T substitution is predominant for mutational processes of UV light exposure, and this is in line with the previous study by Peng et al 1996 [6].

The experiment results also point out the presence of some non-matching signatures (E.g., Extracted signatures 7 and 8 from Figure 15). Also, many of the reference signatures are not successfully extracted with more than 0.8 threshold of cosine similarity (E.g., SBS1 for breast cancer). Additionally, some extracted signatures matched with more than one reference signatures. One potential reason could be non-converged model parameters. This could be solved with experiments with higher iterations and passes and check if model converges or not. To estimate the number of iterations and passes, a grid search like tuning of number of topics could be done.

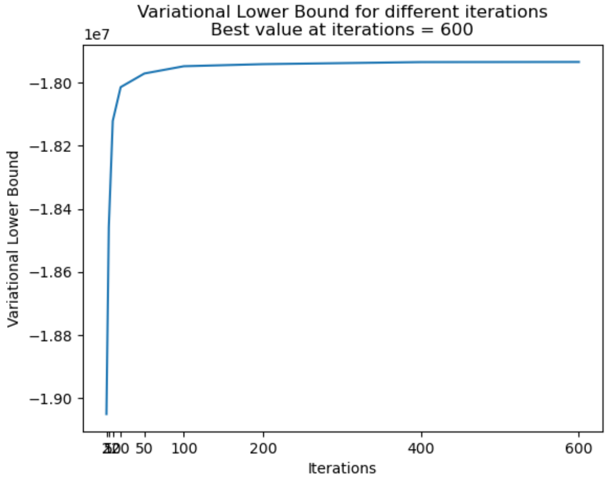


Figure 20: Variation in VLB for different iterations for breast cancer mutation catalog data.

Models trained with different iterations = [2, 5, 10, 20, 50, 100, 200, 400, 600] and VLB calculated and plotted. Value of passes set as equal to iterations for each model. It resulted in a Figure 20. VLB going up sharply till around 100 and increasing slowly afterwards. This does not explain the points highlighted in the previous paragraph. However, this experiment indicates the range of iterations to choose for the experiment.

4.2 Virus Mutational Signatures

Experiments conducted to extract signatures from SARS-CoV-2 mutation samples. Mutation load in the SARS-CoV-2 samples is much smaller compared to cancer samples. Additionally, since the study is focused on mutations happening over the duration of a week, mutations are aggregated for each week. This resulted 86 weeks of data from 18-Jan-2020 to 21-Aug-2021. Count of mutations are found to be lesser at the initial weeks. Since the number of documents are lesser, higher iterations and passes of 1000 selected for the experiment. Grid search to find the number of topics conducted by varying K from 2 to 8 with 1 step each time. Variational Lower Bound value is calculated for each K and tried to optimize for the maximum VLB.

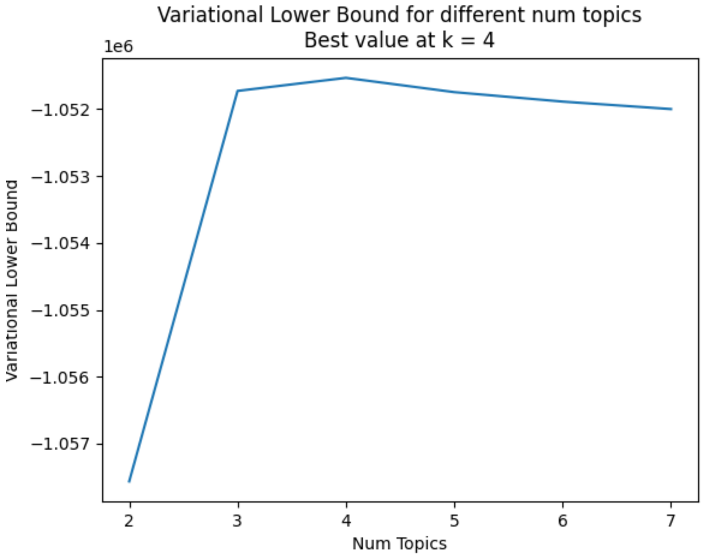


Figure 21: VLB for different values of K for SARS-CoV-2 signature experiments with iterations and passes = 1000

Maximum VLB received at K = 4. Grid search for alpha and beta is conducted by keeping K = 4 and grid search suggested values for Alpha as 0.61, and Beta as 0.91.

16 trainings are conducted with tuned hyper parameters. Signatures, topic probability matrix are extracted from the models and support measures for signatures are calculated. Figure 22 shows visualization of signatures generated by one training. Three stable signatures (Signature-0,2,3) and one noise signature (Signature-1) are visible. Both the support measures – support presence and support probability indicated the presence of noise signature – 2 as in Figure 23.

The study of Graudenzi et al. 2021 extracted three stable cluster signatures – cluster SC#1, SC#2, SC#3 [25]. These signatures are considered as reference and compared with the three extracted signatures. Signature-3 has predominant C>T substitutions and visibly like cluster SC#1 Again, Signature-0 is more like cluster SC#2 and Signature-2 is more like cluster SC#3. A quantitative comparison using cosine similarilty between the extracted signatures and reference signatures could not be done due to the difference in the mutation context (96 in [23] and 192 in this study) and time limitation.

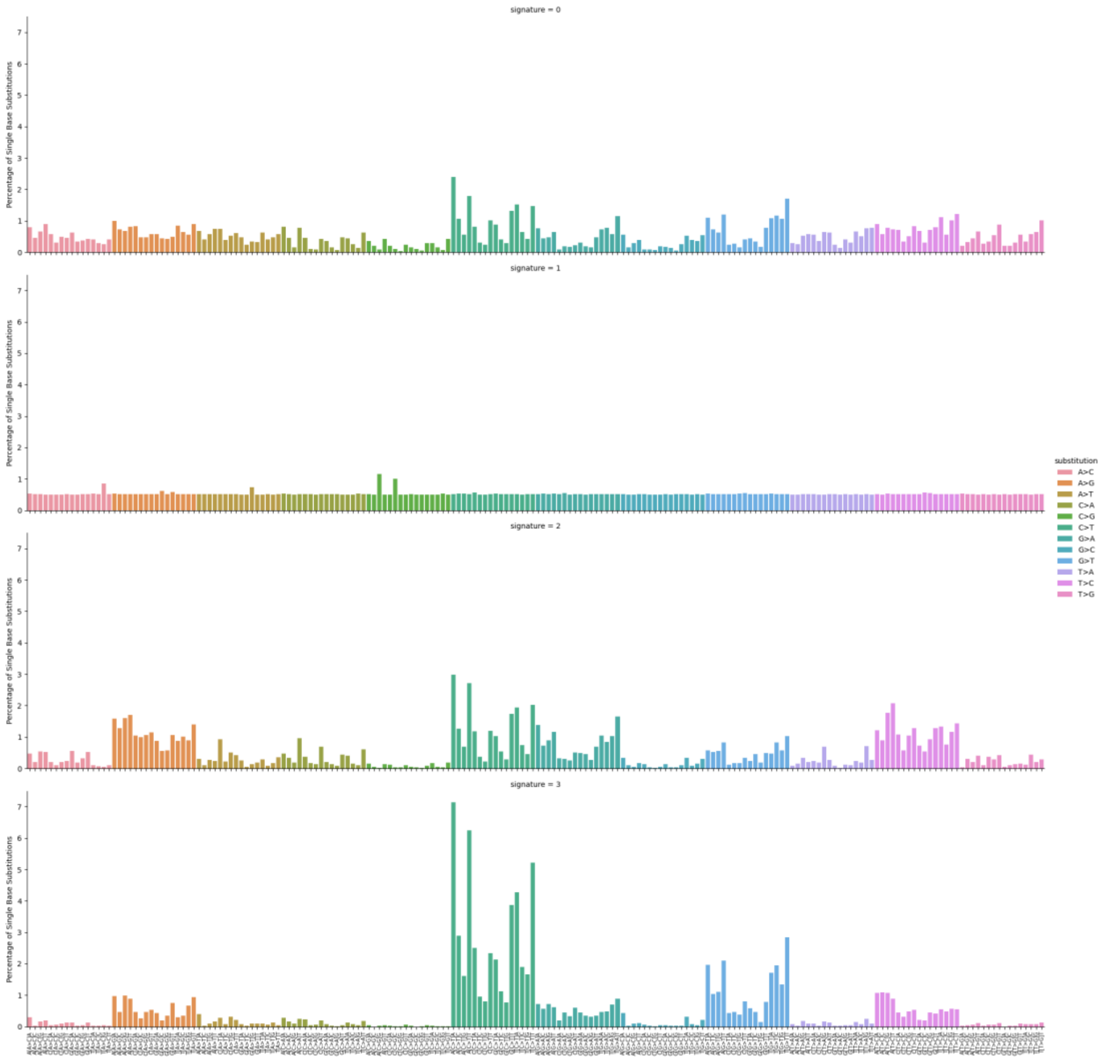


Figure 22: Signatures extracted by a tuned training from SARS-CoV-2 data with 1000 iterations. Note: signature second from the top is noise.

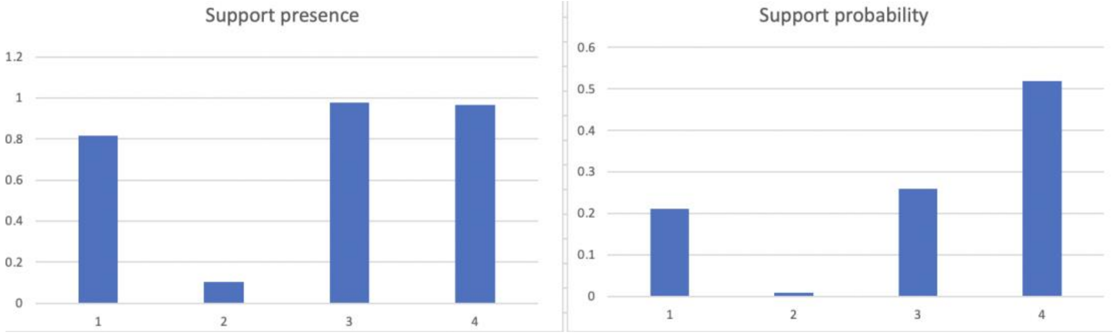


Figure 23: Left – Support presence, right – Support probability calculated for the signatures generated by a tuned training for SARS-CoV-2 mutations

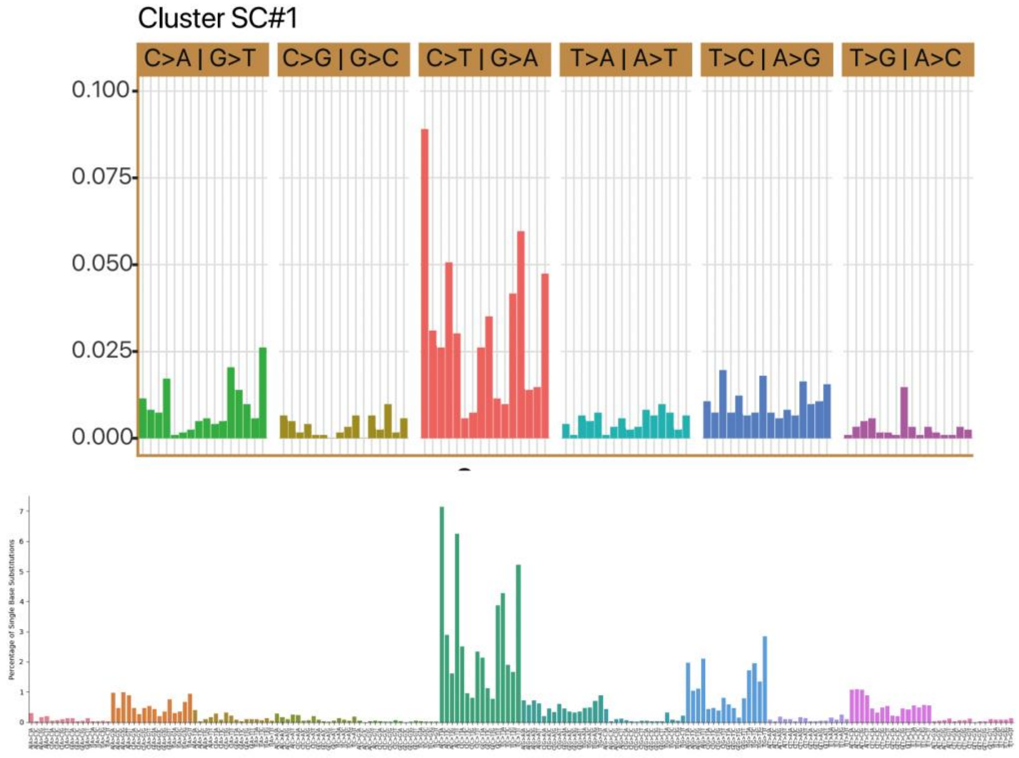


Figure 24: An attempt to visually compare reference signature cluster SC#1 with extracted Signature-3.

Figure 24 is an attempt to visually compare SC#1 reference signature with extracted signature 1. However, since the ordering of mutation types, context and colors are different, it is difficult to compare visually. However, closer look highlights the evident similarity between high values of C>T, G>T, T>C and A>G substitutions.

Therefore, comparison of extracted signatures with reference sequences based on the percentage of contribution of substitution types is performed. The visualizations of reference signature uses a grouping of contexts as into 6 categories: {C>A:G>T, C>G:G>C, C>T:G>A, T>A:A>T, T>C:A>G, T>G:A>C }. For the comparison, similar grouping is performed with the extracted signature and similar color scheme is used to denote context groupings in the visualizations. Comparison shows striking similarity of the extracted signatures with reference signatures. Figures 25, 26, and 27 shows visual comparison between reference signatures and extracted signatures and the similarity found to be quite evident.

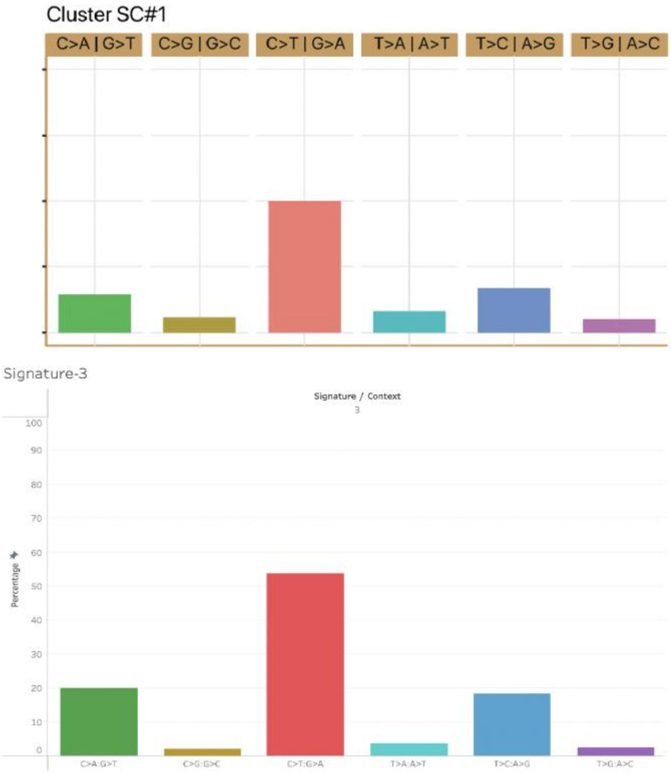


Figure 25: Comparison of Cluster SC#1 of reference signature(top) with extracted Signature-3(bottom)

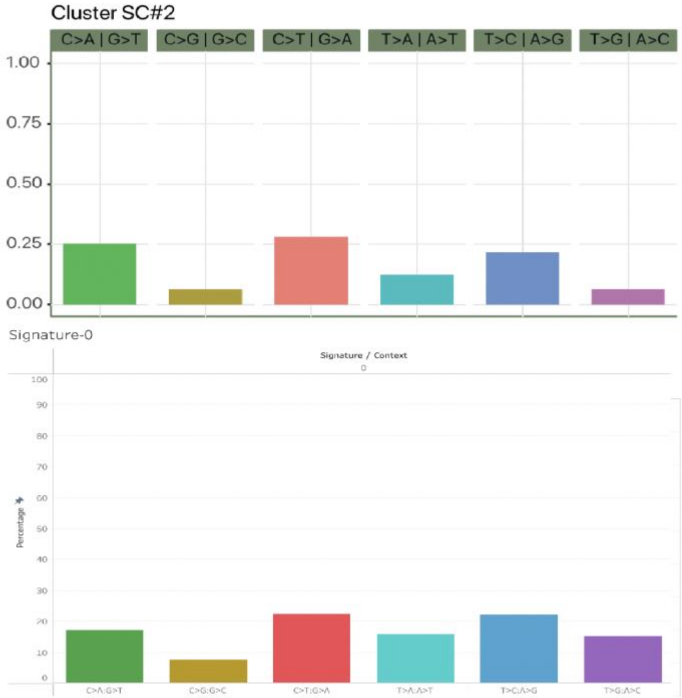


Figure 26: Comparison of Cluster SC#2 of reference signature(top) with extracted Signature-0(bottom)

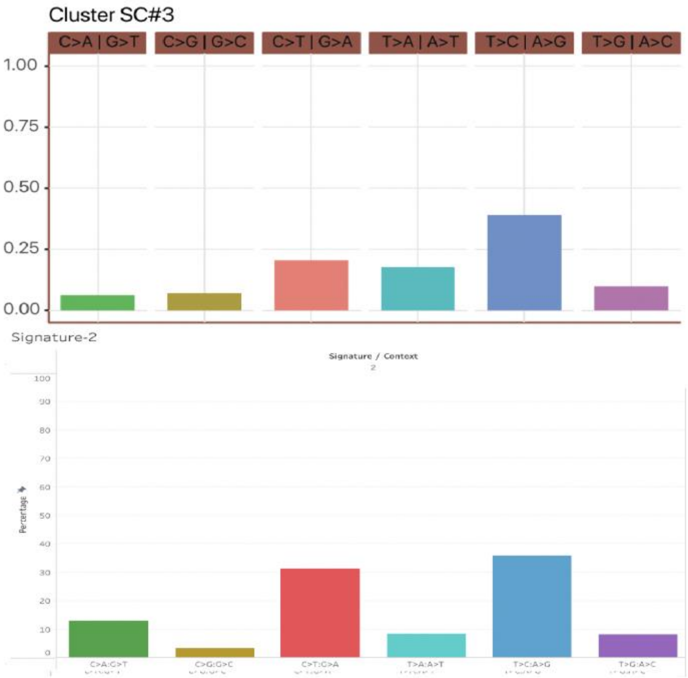


Figure 27: Comparison of Cluster SC#3 of reference signature(top) with extracted Signature-2(bottom)

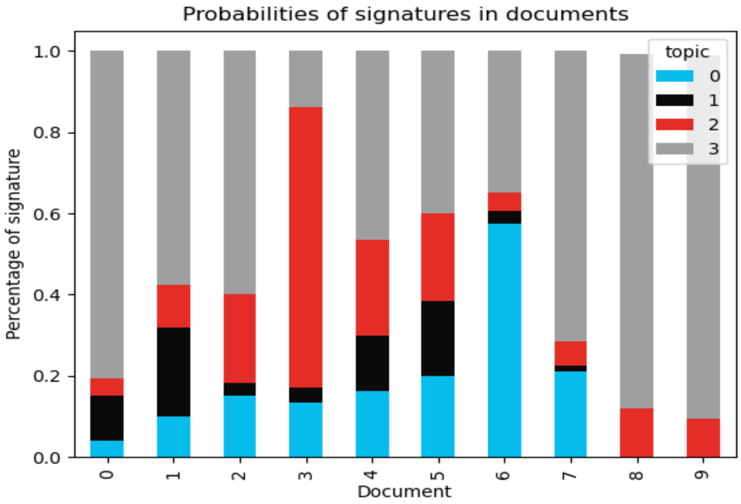


Figure 28: Visualization of document signature probability matrix.

16 tuned training models generated a total of 16x4 = 64 signatures. Signatures bootstrapped by combining signatures with cosine similarity > 0.9 iteratively. This resulted in the extraction of 2 major signatures and one noise signature as shown in Figure 29. Ensemble method generated the signatures as in Figure 30. Both the methods indicate the presence of reference signature cluster SC#1.

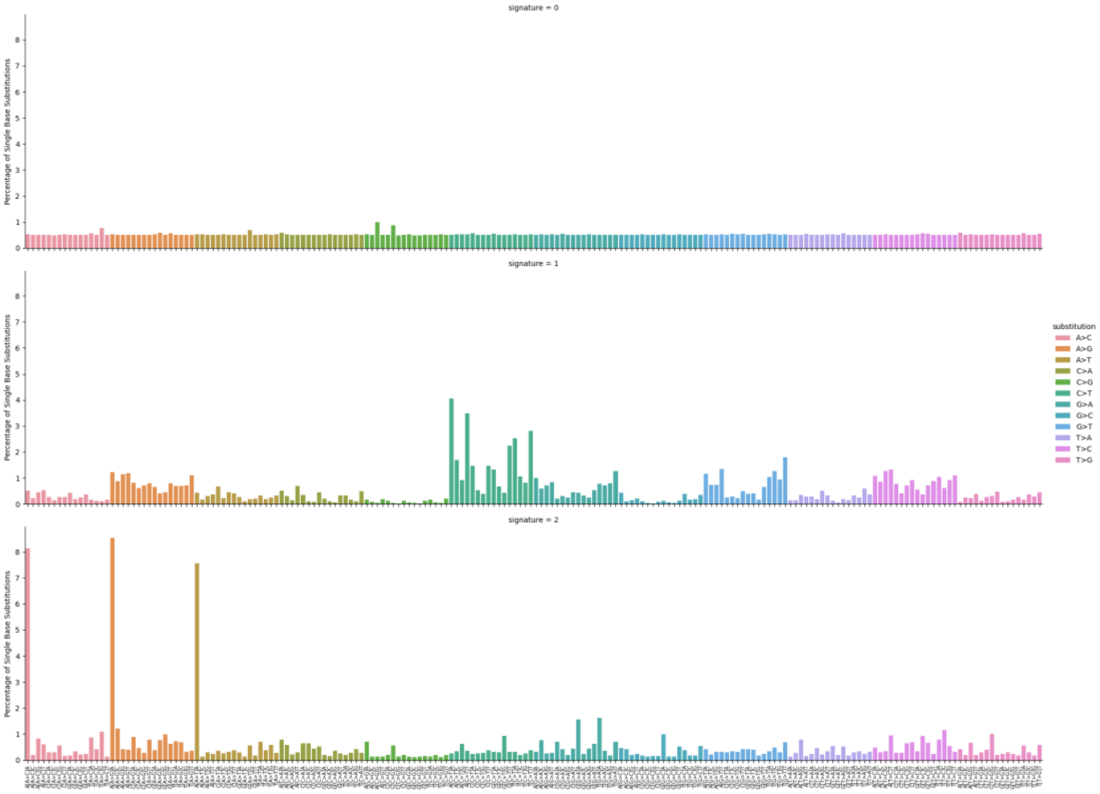


Figure 29: Signatures extracted by bootstrapping of signatures from 16 trained models from SARS-CoV-2 data. Note: signature 1 is noise.

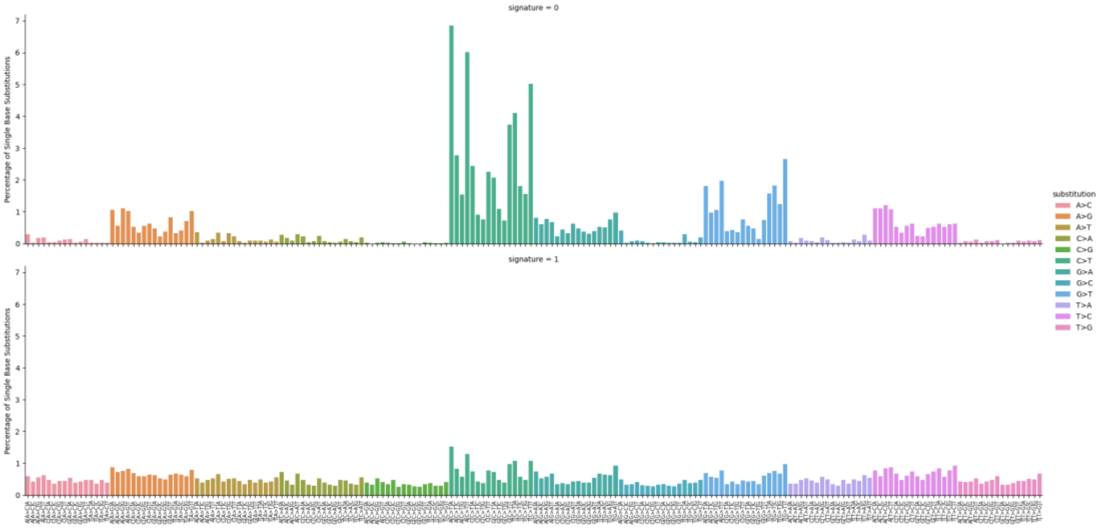


Figure 30: Signatures extracted by ensemble of 16 model training from SARS-CoV-2 data.

An experiment to compare the model performance with different iterations is conducted. Value of passes set as equal to iterations for each model. Models generated with different Iterations = [2, 5, 10, 30, 50, 100, 200, 400, 600, 800, 1000] and Variational Lower Bound is measured and plotted. This provided a graph as in Figure 31. VLB improved sharply till 100 iterations, improved steadily thereafter reaching a plateau at around 400 iterations. Increasing in VLB is slow after 400 until 1000 iterations. Results of this experiment support the selected number of iterations (1000) used for the study of SARS-CoV-2 mutation catalog dataset.

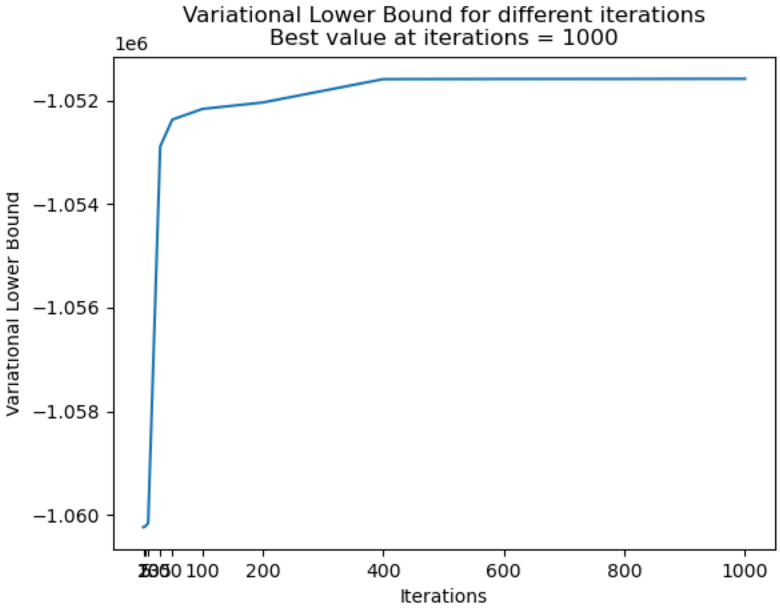


Figure 31: Variation in VLB for different iterations (and passes) for SARS-CoV-2 mutation catalog.

Chapter 5 Conclusion

An experiment computation framework for extracting the mutational signatures from somatic cancer mutations and virus mutations is developed as part of this project. This is an easy to use, scalable platform to perform reproducible experiments and can support future experiments. The features – bootstrapping, ensemble, and support measures are found to be useful in extraction of stable signatures. Multiple experiments conducted with publicly available cancer mutations catalogs and the comparison with the well-established COSMIC signatures [14] suggest that the framework could extract many signatures with high similarity with the reference signatures. Experiments performed with publicly available SARS-CoV-2 virus mutation data and 3 signatures are extracted. Comparison with [25] indicated very high similarity with the reference signatures. Support analyses are performed and found that the signatures are stable. Additionally, a further study could identify matching signatures present in different virus mutations such as influenza, zika etc. Due to time limitation, and unavailability of ready to use mutational catalog for other human viruses, they could not be included in this project. This project is a first step towards such a comprehensive study. Framework is developed in view of future requirement of experiments with multiple virus catalogs and platform can be used to perform these experiments with minor changes in code once the data is available.

The three main challenges faced during this project are - time, computational power, and dataset availability. More comprehensive experiments could be done to extract signatures from all the possible cancer classifications and could be compared with the COSMIC v3.0 reference signatures. COSMIC is constantly updating the signatures and study could be done using the latest signatures. At the time of writing this report, latest is v3.3. Experiments could be done to perform grid search for hyperparameter tuning with a bigger grid so that possibility of reaching global maximum is increased. Further, more experiments to be conducted to identify the number of iterations and passes to be used. System is designed in such a way that we can tune the experiment parameters based on system configuration. Experiment could be done for all cancer types with all the data together with very high iterations and passes and number of models, or for only one cancer type with a smaller number of input data with iteration, passes, number of models as low as 1(a test run). Therefore, a machine with higher configuration would be able to perform the experiments faster and therefore, with higher iterations, passes, and number bootstrapping models could extract more stable signatures with the same framework.

Another interesting direction is to select a subset of records randomly for each model training before the bootstrapping. This will be better when the number of samples is high enough (E.g., Breast cancer). In this approach, rather than taking all the records, each run could be done using subset of random N records (E.g., 200). The computational power saved for processing more records could be used for making a greater number of models so that more stable topics could be generated. However, this approach would not be suitable for types with number of samples is lesser for model to converge.

Variance of the number of mutation samples available for each week of SARS-CoV-2 data is very high. There are weeks with less than 10 samples and greater than 1000 samples. The experiment would be fairer with a comparable number of samples for each week since mutations are aggregated over each week. This point to the requirement of more mutation catalog samples.

More analysis could be done in the support analysis especially on how to calculate the support when models are bootstrapped. Additionally, the framework currently stores most of the results in the local disk. Integration of a database could help in organizing and further analysis of the experiments. Also, the current study is limited to Single Base Substitutions (SBS). More study can be done using DBS, Indels, etc. Another possible direction is to write custom algorithm to perform LDA than using existing package. A custom made LDA will be beneficial in customizing as per the needs of the project.

Finally, but more importantly, more studies need to be performed to analyze the biological significance and application of the results.

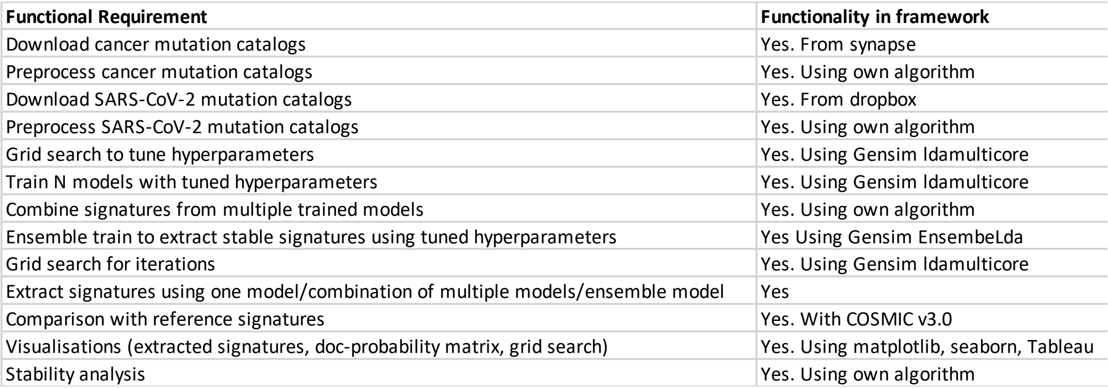
Appendix A Download and use framework

Sample code to download the framework and execute the experiments are given below:

```
# Install dependent packages

# To download cancer mutation catalog
pip install synapseclient

# For LDA implementation
pip install --upgrade genism

# To download source code
git clone https://stgit.dcs.gla.ac.uk/2559542s/MutSigExperiments



# Change directory
cd /MutSigExperiments

# Sample code to run an experiment
# Change experiment parameters for different experiments
MutSigExperiments> python run_experiment.py
```

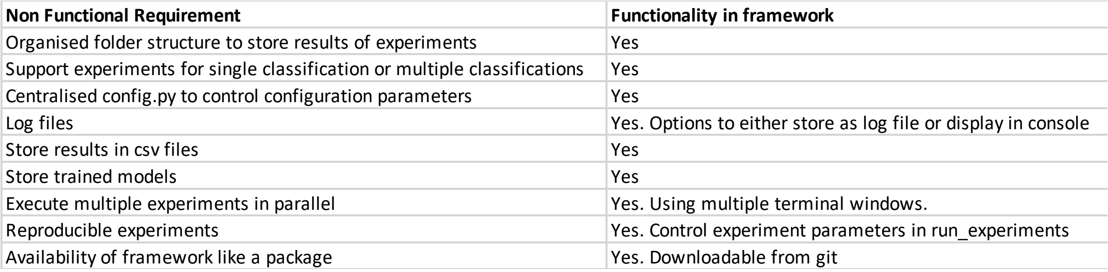
Some sample run_experiments.py files are provided. Iterations and passes set as 1 for test run. Change the iterations and passes parameters to high values to conduct actual experiments.

Appendix B Functional and Non-functional requirements



Functional Requirement	Functionality in framework
Download cancer mutation catalogs	Yes. From synapse
Preprocess cancer mutation catalogs	Yes. Using own algorithm
Download SARS-CoV-2 mutation catalogs	Yes. From dropbox
Preprocess SARS-CoV-2 mutation catalogs	Yes. Using own algorithm
Grid search to tune hyperparameters	Yes. Using Gensim ldamulticore
Train N models with tuned hyperparameters	Yes. Using Gensim ldamulticore
Combine signatures from multiple trained models	Yes. Using own algorithm
Ensemble train to extract stable signatures using tuned hyperparameters	Yes Using Gensim EnsembeLda
Grid search for iterations	Yes. Using Gensim ldamulticore
Extract signatures using one model/combination of multiple models/ensemble model	Yes
Comparison with reference signatures	Yes. With COSMIC v3.0
Visualisations (extracted signatures, doc-probability matrix, grid search)	Yes. Using matplotlib, seaborn, Tableau
Stability analysis	Yes. Using own algorithm

Figure B.1: Functional requirements considered for the framework and their availability



Non Functional Requirement	Functionality in framework
Organised folder structure to store results of experiments	Yes
Support experiments for single classification or multiple classifications	Yes
Centralised config.py to control configuration parameters	Yes
Log files	Yes. Options to either store as log file or display in console
Store results in csv files	Yes
Store trained models	Yes
Execute multiple experiments in parallel	Yes. Using multiple terminal windows.
Reproducible experiments	Yes. Control experiment parameters in run_experiments
Availability of framework like a package	Yes. Downloadable from git

Figure B.2: Non-functional requirements considered for the framework and their availability

Appendix C Extracted mutational signatures

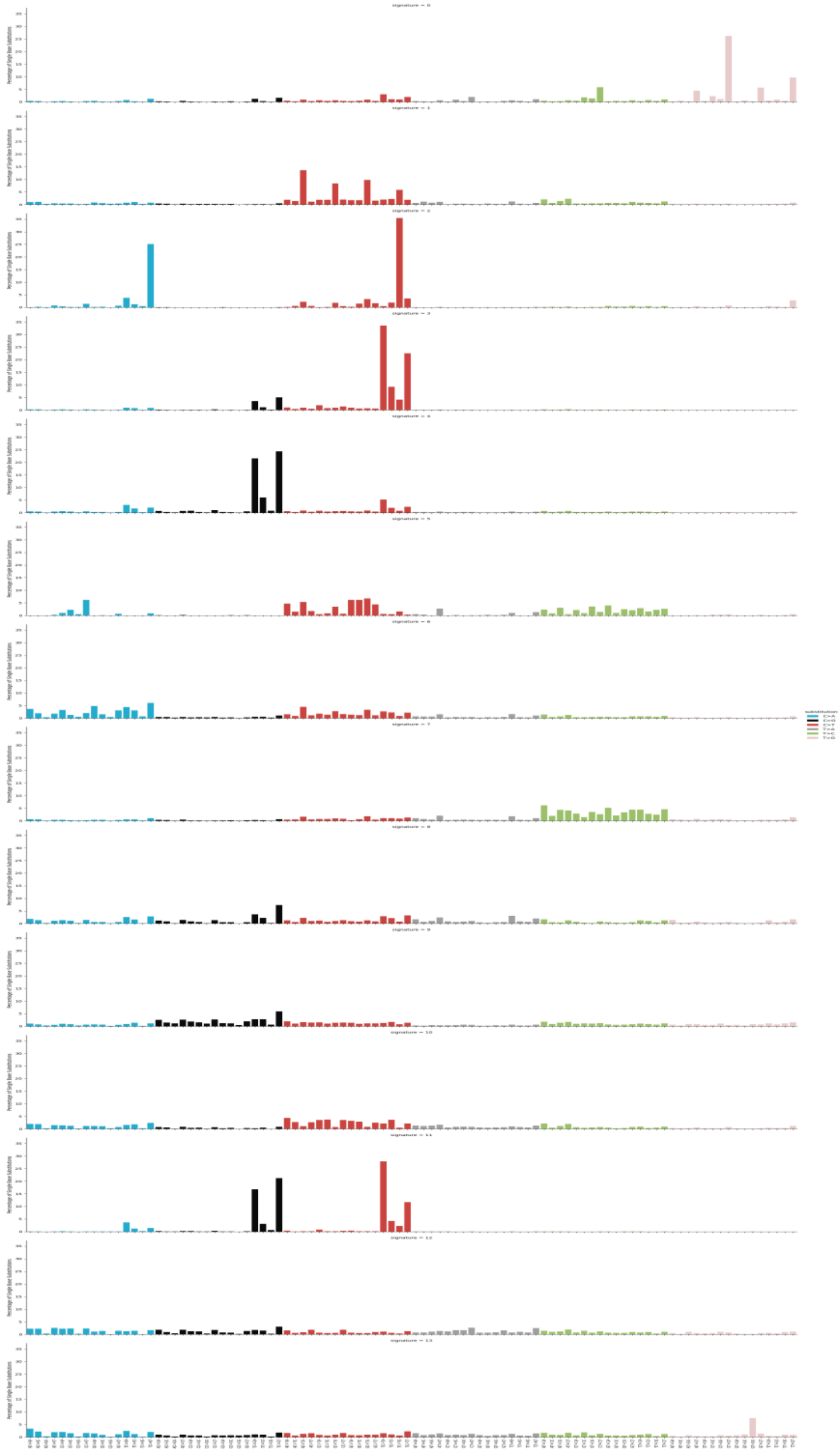


Figure C.1: Visualization of mutational signature set (14 signatures) extracted from breast cancer samples by means of a model trained with tuned hyper parameters. (200 iterations)

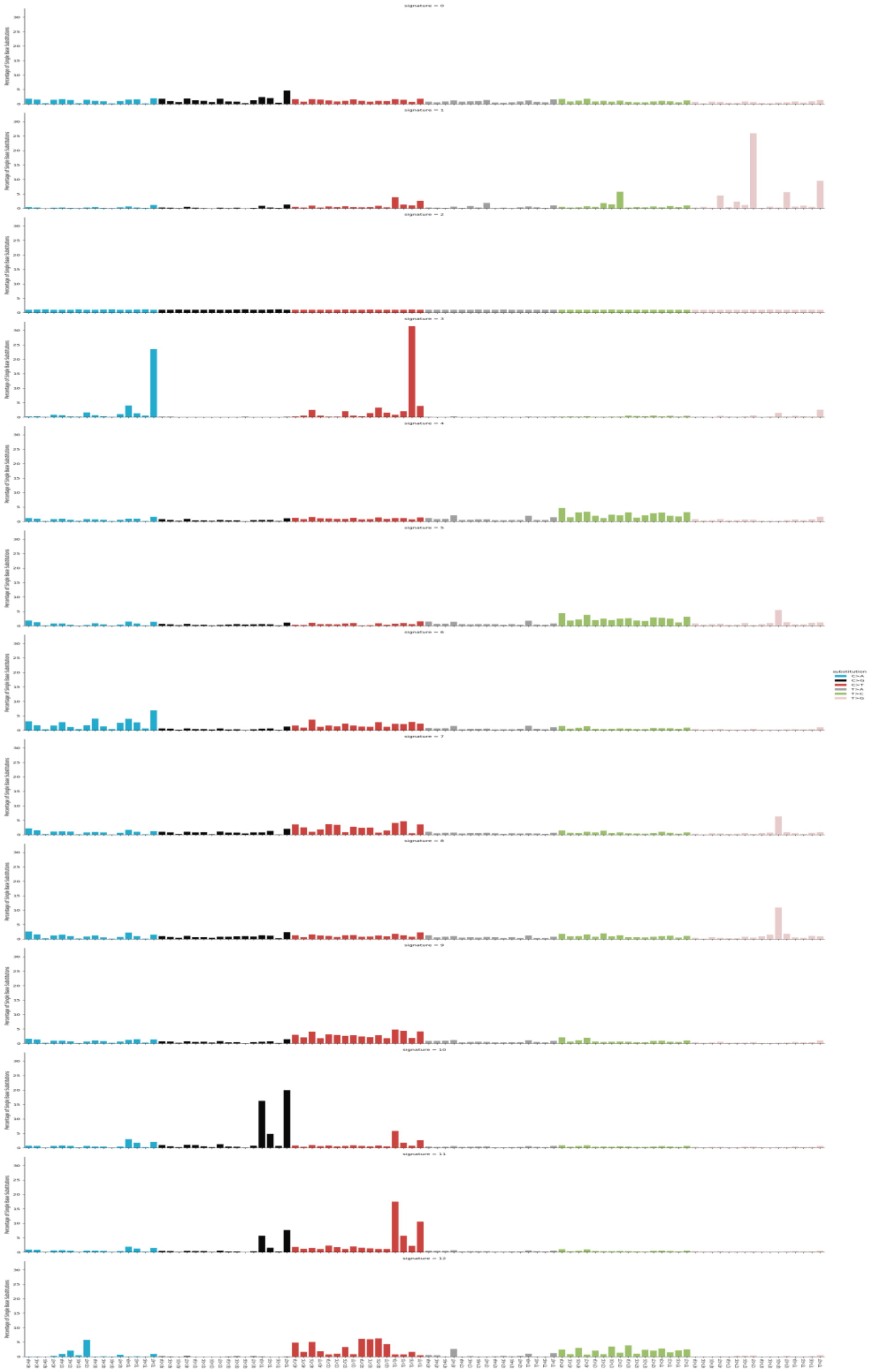


Figure C.2: Visualization of mutational signatures extracted from breast cancer samples using bootstrapping of 16 trainings with tuned hyper parameters. (Note - Signature 2 is noise) (400 iterations)

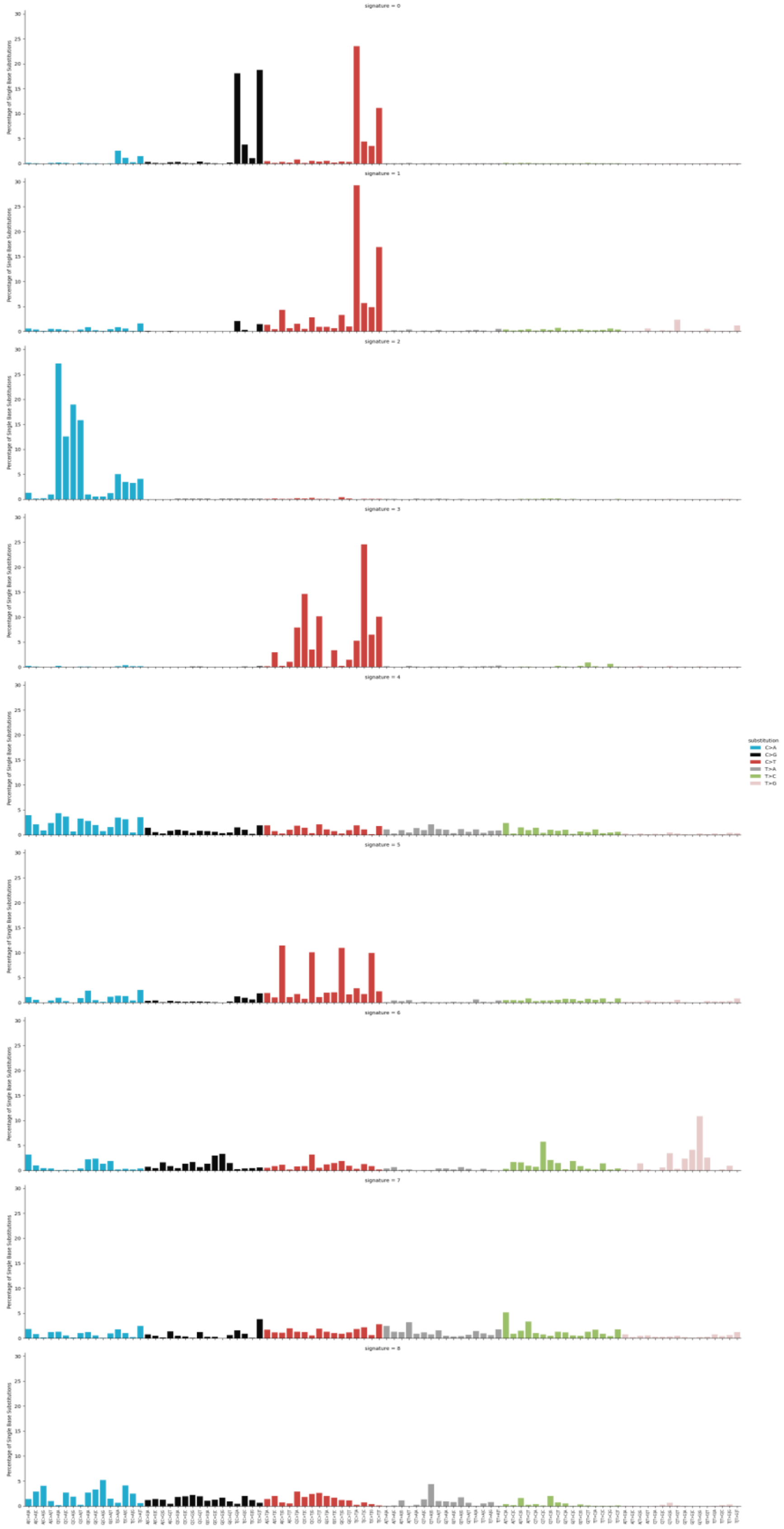


Figure C.3: Visualization of mutational signatures extracted from Head-SCC cancer samples using ensemble LDA with 16 models. (400 iterations)

Bibliography

[1] A. D. Hershey and M. Chase, “INDEPENDENT FUNCTIONS OF VIRAL PROTEIN AND NUCLEIC ACID IN GROWTH OF BACTERIOPHAGE,” *Journal of General Physiology*, vol. 36, no. 1, pp. 39–56, Sep. 1952, doi: 10.1085/jgp.36.1.39.

[2] F. H. CRICK, “On protein synthesis.,” *Symp Soc Exp Biol*, vol. 12, pp. 138–63, 1958.

[3] wikipedia, “DNA and RNA codon tables,” *https://en.wikipedia.org/wiki/DNA_and_RNA_codon_tables*, 2022.

[4] S. Clancy, “Genetic Mutation,” *Nature Education 1(1) https://www.nature.com/scitable/topicpage/genetic-mutation-441/*, 2008.

[5] S. Toyooka, T. Tsuda, and A. F. Gazdar, “The TP53 gene, tobacco exposure, and lung cancer.,” *Hum Mutat*, vol. 21, no. 3, pp. 229–39, Mar. 2003, doi: 10.1002/humu.10177.

[6] W. Peng and B. R. Shaw, “Accelerated deamination of cytosine residues in UV-induced cyclobutane pyrimidine dimers leads to CC-->TT transitions.,” *Biochemistry*, vol. 35, no. 31, pp. 10172–81, Aug. 1996, doi: 10.1021/bi960001x.

[7] S. Nik-Zainal *et al.*, “Mutational processes molding the genomes of 21 breast cancers.,” *Cell*, vol. 149, no. 5, pp. 979–93, May 2012, doi: 10.1016/j.cell.2012.04.024.

[8] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. Campbell, and M. R. Stratton, “Deciphering signatures of mutational processes operative in human cancer.,” *Cell Rep*, vol. 3, no. 1, pp. 246–59, Jan. 2013, doi: 10.1016/j.celrep.2012.12.008.

[9] L. B. Alexandrov *et al.*, “Signatures of mutational processes in human cancer,” *Nature*, vol. 500, no. 7463, pp. 415–421, Aug. 2013, doi: 10.1038/nature12477.

[10] L. B. Alexandrov *et al.*, “Clock-like mutational processes in human somatic cells.,” *Nat Genet*, vol. 47, no. 12, pp. 1402–7, Dec. 2015, doi: 10.1038/ng.3441.

[11] Y. Shiraishi, G. Tremmel, S. Miyano, and M. Stephens, “A Simple Model-Based Approach to Inferring and Visualizing Cancer Mutation Signatures,” *PLoS Genet*, vol. 11, no. 12, p. e1005657, Dec. 2015, doi: 10.1371/journal.pgen.1005657.

[12] S. Nik-Zainal *et al.*, “Landscape of somatic mutations in 560 breast cancer whole-genome sequences,” *Nature*, vol. 534, no. 7605, pp. 47–54, Jun. 2016, doi: 10.1038/nature17676.

[13] R. A. Rosales, R. D. Drummond, R. Valieris, E. Dias-Neto, and I. T. da Silva, “signeR: an empirical Bayesian approach to mutational signature discovery,” *Bioinformatics*, vol. 33, no. 1, pp. 8–16, Jan. 2017, doi: 10.1093/bioinformatics/btw572.

[14] L. B. Alexandrov *et al.*, “The repertoire of mutational signatures in human cancer,” *Nature*, vol. 578, no. 7793, pp. 94–101, Feb. 2020, doi: 10.1038/s41586-020-1943-3.

[15] M. R. Stratton, “Exploring the genomes of cancer cells: progress and promise.,” *Science*, vol. 331, no. 6024, pp. 1553–8, Mar. 2011, doi: 10.1126/science.1204040.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 993–1022, Mar. 2003.

[17] J. G. Tate *et al.*, “COSMIC: the Catalogue Of Somatic Mutations In Cancer,” *Nucleic Acids Res*, vol. 47, no. D1, pp. D941–D947, Jan. 2019, doi: 10.1093/nar/gky1015.

[18] Y. Shu and J. McCauley, “GISAID: Global initiative on sharing all influenza data – from vision to reality,” *Eurosurveillance*, vol. 22, no. 13, Mar. 2017, doi: 10.2807/1560-7917.ES.2017.22.13.30494.

[19] National Human Genome Research Institute (NHGRI)., “Human Genome Project Completion: Frequently Asked Questions,” 2022.

[20] International Cancer Genome Consortium *et al.*, “International network of cancer genome projects.,” *Nature*, vol. 464, no. 7291, pp. 993–8, Apr. 2010, doi: 10.1038/nature08987.

[21] D. Lee and H. S. Seung, “Algorithms for Non-negative Matrix Factorization,” in *Advances in Neural Information Processing Systems*, 2000, vol. 13. [Online]. Available: https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf

[22] X. Zou, M. Owusu, R. Harris, S. P. Jackson, J. I. Loizou, and S. Nik-Zainal, “Validating the concept of mutational signatures with isogenic cell models,” *Nat Commun*, vol. 9, no. 1, p. 1744, Dec. 2018, doi: 10.1038/s41467-018-04052-8.

[23] T. Matsutani, Y. Ueno, T. Fukunaga, and M. Hamada, “Discovering novel mutation signatures by latent Dirichlet allocation with variational Bayes inference,” *Bioinformatics*, vol. 35, no. 22, pp. 4543–4552, Nov. 2019, doi: 10.1093/bioinformatics/btz266.

[24] D. Fantini, V. Vidimar, Y. Yu, S. Condello, and J. J. Meeks, “MutSignatures: an R package for extraction and analysis of cancer mutational signatures,” *Sci Rep*, vol. 10, no. 1, p. 18217, Dec. 2020, doi: 10.1038/s41598-020-75062-0.

[25] A. Graudenzi, D. Maspero, F. Angaroni, R. Piazza, and D. Ramazzotti, “Mutational signatures and heterogeneous host response revealed via large-scale characterization of SARS-CoV-2 genomic diversity.,” *iScience*, vol. 24, no. 2, p. 102116, Feb. 2021, doi: 10.1016/j.isci.2021.102116.

[26] Radimrehurek.com, “Gensim: topic modelling for humans,” *https://radimrehurek.com/gensim_3.8.3/models/wrappers/ldamallet.html*, 2022.

[27] Radimrehurek.com, “Gensim: topic modelling for humans,” *https://radimrehurek.com/gensim/models/ldamulticore.html*, 2022.

[28] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet Processes,” *J Am Stat Assoc*, vol. 101, no. 476, pp. 1566–1581, Dec. 2006, doi: 10.1198/016214506000000302.

[29] radimrehurek.com, “Ensemble Latent Dirichlet Allocation,” *https://radimrehurek.com/gensim/models/ensemblelda.html*, 2022.

[30] P. V’kovski, A. Kratzel, S. Steiner, H. Stalder, and V. Thiel, “Coronavirus biology and replication: implications for SARS-CoV-2,” *Nat Rev Microbiol*, vol. 19, no. 3, pp. 155–170, Mar. 2021, doi: 10.1038/s41579-020-00468-6.

[31] K. Chan *et al.*, “An APOBEC3A hypermutation signature is distinguishable from the signature of background mutagenesis by APOBEC3B in human cancers,” *Nat Genet*, vol. 47, no. 9, pp. 1067–1072, Sep. 2015, doi: 10.1038/ng.3378.

[32] S. Nik-Zainal *et al.*, “The genome as a record of environmental exposure,” *Mutagenesis*, p. gev073, Oct. 2015, doi: 10.1093/mutage/gev073.