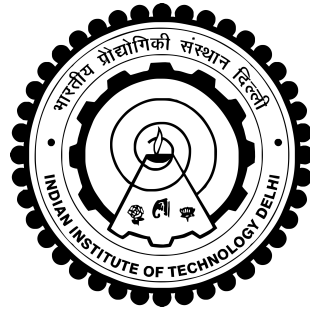


LEARNING ON LARGE DATASETS USING BIT-STRING TREES

PRASHANT GUPTA



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI

April 2023

©Indian Institute of Technology Delhi (IITD), New Delhi, 2023

LEARNING ON LARGE DATASETS USING BIT-STRING TREES

by

PRASHANT GUPTA

DEPARTMENT OF ELECTRICAL ENGINEERING

Submitted

*in fulfillment of the requirements of the degree of Doctor of Philosophy
to the*



INDIAN INSTITUTE OF TECHNOLOGY DELHI

April 2023

Certificate

This is to certify that the thesis entitled “**Learning On Large Datasets Using Bit-String Trees**”, being submitted by **Prashant Gupta** for the award of the degree of **Doctor of Philosophy** to the Department of Electrical Engineering, Indian Institute of Technology Delhi, is a record of bonafide work done by him under my supervision and guidance. The matter embodied in this thesis has not been submitted to any other University or Institute for the award of any other degree or diploma.

Dr. Jayadeva

Professor

Department of Electrical Engineering,
Indian Institute of Technology Delhi,
Hauz Khas, New Delhi - 110016,
India.

Dr. Vibhor Kumar

Associate Professor

Department of Computational Biology,
Indraprastha Institute of Information Technology,
Okhla Phase III, Delhi - 110020,
India.

Dr. Debarka Sengupta

Associate Professor

Department of Computer Science & Engineering,
Department of Computational Biology,
Head, Center for Artificial Intelligence,
Indraprastha Institute of Information Technology,
Okhla Phase III, Delhi - 110020,
India.

(Adj.) Associate Professor

Institute of Health & Biomedical Innovation,
QUT, Australia

Acknowledgments

I would like to thank my supervisors Prof. Jayadeva (IITD), Assoc. Prof. Debarka Sengupta (IIITD), Assoc. Prof. Vibhor Kumar (IIITD), Research Committee Members Prof. Indra Narayan Kar (IITD), Prof. Shouri Chatterjee (IITD), and Asst. Prof. Vivekanandan Perumal (IITD); faculty members with whom I worked - Retd. Prof. Suresh Chandra (IITD), Assoc. Prof. Gaurav Ahuza (IIITD), Retd. Prof. Munishwar Nath Gupta (IITD), and Prof. Rajeev Narang (AIIMS); friends and colleagues Aashi Jindal, Dr. Sumit Soman, Dr. Udit Kumar, Aashish Rajiv, Dr. Mayank Sharma, Dr. Shruti Sharma, Dr. Himanshu Pant, Devesh Bajpai, Ritika Bajpai, Sanjay Pandey; department staff members Rakesh Kumar, Yatindra Mani, Mukesh, and Ritwik Pahari; my parents, my parents-in-law, my wife Aashi Jindal and other family members Priyanka Gupta, Sahil Jindal, and Khushboo Mehrotra for supporting me through this journey.

I would like to extend my special thanks to my wife Aashi Jindal for being the part of both journeys, academic and life, and supporting me through the ups of downs.

(Prashant Gupta)

Abstract

Similarity preserving hashing finds widespread application in nearest-neighbor search. The widely used form of similarity preserving hashing is space-partitioning-based hashing. Many space partitioning-based hashing techniques generate bit codes as hash codes. Although Binary Search Trees (BSTs) can be used for storing bit codes, their size grows exponentially with code length. In practice, such a tree turns out to be highly sparse, increasing the *miss-rate* of nearest neighbor searches. To tackle sparsity and memory issues of BST, we first developed Compressed BST of Inverted hash tables (ComBI), a geometrically motivated compression technique for BSTs. ComBI enables fast and approximate nearest neighbor searches without a significant memory footprint over BSTs. We show, that approximate search in ComBI is competitive with an exact search algorithm in retrieving the nearest neighbors in a hamming space. On a database containing ~ 80 million samples, ComBI yields an average precision of 0.90, at $\sim 4X$ - $\sim 296X$ improvements in run-time across different code lengths when compared to Multi-Index Hashing (MIH), a widely used exact search method. On a database consisting of 1 billion samples, this value of precision (0.90) is reached at $\sim 4X$ - $\sim 19X$ improvements in run-time. Next, the ComBI has been shown as a search engine for single-cell RNA sequencing (scRNA-seq) data, and its performance is compared with the state-of-the-art scRNA-seq search engine method, Cellfishing.jl, which is based on the MIH. The ComBI outperforms Cellfishing.jl in multiple accounts. The achieved speed-up in the search is around ~ 2 - ~ 13 .

We next shift our attention to using similarity preserving hashing to build a classifier. The learned structure of hashing algorithms is suitable to be combined with a Bayes' classifier. We explored the construction of three basic space-partitioning-based hashing algorithms and identified their pros and cons. This motivated us to build a tree-based hashing classifier. We present Guided Random Forest (GRAF), a tree-based ensemble hashing classifier that realizes global partitioning by extending the idea of building oblique decision trees with localized partitioning. We show that GRAF bridges the gap between decision trees and boosting algorithms. Experiments indicate that it reduces the generalization error bound. Results on 115 benchmark datasets show that GRAF yields comparable or better results on a majority of datasets. We also build an unsupervised version of GRAF, Unsupervised GRAF (uGRAF), to perform guided hashing. The GRAF fundamentally works by generating more hyperplanes in the region of high data complexity and this phenomenon is represented by the number of planes required to classify a sample correctly. This measure can be used for importance sampling. In the next part of the thesis, this direction

is explored to build a data approximator using GRAF. An extensive empirical evaluation with simulated and UCI datasets was performed to establish the theory. The proposed methodology is compared with the two state-of-the-art importance sampling algorithms. An analogy between Support Vector Machine (SVM) and the samples marked by GRAF as of high importance is also developed.

We then show that the learned neighborhood of a sample can be used to estimate the confusion around the sample in a scalable manner. We utilized uGRAF and ComBI to estimate the per-sample classifiability. An empirical evaluation of estimated values is presented. We show how per-sample classifiability can be used to estimate cancer patient survivability.

Cancer is a disease of the genome. Genomic changes resulting in cancer can be inherited, brought on by environmental carcinogens, or may result from random replication errors. Mutations continue to spread after the induction of carcinogenicity and significantly change cancer genomes. Most cancer-related somatic mutations are indistinguishable from germline variants or other non-cancerous somatic mutations, even though only a small subset of driver mutations have been identified and characterized thus far. Thus, such overlap makes it difficult to understand many harmful but unstudied somatic mutations. The main bottleneck results from patient-to-patient variation in mutational profiles, which makes it challenging to link particular mutations with a particular disease outcome. This thesis introduces a newly developed method called Continuous Representation of Codon Switches (CRCS). This deep learning-based approach enables us to produce numerical vector representations of genetic changes, enabling a variety of machine learning-based tasks. We show how CRCS can be used in three different ways. First, we show how it can be used to find cancer-related somatic mutations without matched normal samples. Second, the suggested method makes it possible to find and study driver genes. Finally, we created a numerical representation of mutations by combining a sequence classifier with CRCS. These representations are used to score individual mutations in a tumor sample using per-sample classifiability, which was found to be predictive of patient survival in Bladder Urothelial Carcinoma (BLCA), Hepatocellular Carcinoma (HCC), and Glioblastoma Multiforme (GBM). Taken together, we propose CRCS as a valuable computational tool for analysis of the functional significance of individual cancer mutations.

सार

समानता को संरक्षित रखने वाली हैशिंग प्रक्रिया का निकटतम-पड़ोसी को खोजने में व्यापक अनुप्रयोग मिलता है। समानता-संरक्षण हैशिंग प्रक्रियाओं का अधिकतम उपयोग विस्तार-विभाजन आधारित प्रणालियों में होता है। कई विस्तार-विभाजन आधारित हैशिंग प्रणालियाँ, हैश संकेतावली के रूप में अंश संकेतावली उत्पन्न करती है। हालांकि द्विआधारी खोज वृक्ष (BST) का उपयोग अंश संकेतावलीओं को संगृहित करने के लिए किया जा सकता है, लेकिन इस प्रकार के वृक्षों का आकार संकेतावलीओं की लंबाई के साथ तेजी से बढ़ता है। व्यवहार में ऐसे वृक्ष अत्यधिक विरल होती है, जिससे निकटतम-पड़ोसियों को खोजने की क्षति-दर में वृद्धि होती है। BST की विरलता और स्मृति की समस्या के समाधान के लिए हमने उलटे हैश तालिका की सम्पीडित द्विआधारी खोज वृक्ष (ComBI) को विकसित किया, जो BST की ज्यामितीय रूप से प्रेरित संपीड़न तकनीक है। ComBI, BST के उल्लेखनीय स्मृति पदचिन्ह के बिना तेज़ और अनुमानित निकटतम-पड़ोसियों की खोज करने में सक्षम है। इस शोध प्रबंध में हम दिखाते हैं कि हैमिंग विस्तार में ComBI द्वारा अनुमानित रूप से खोजे हुए निकटतम पड़ोसी, यथार्थ पड़ोसियों की खोज करने वाली कलन विधियों के साथ प्रतिस्पर्धी है। व्यापक रूप से उपयोग होने वाले यथार्थ खोज की विधि, बहु सूचकांक हैशिंग (MIH) की तुलना में ComBI ~८ करोड़ प्रतिदर्श वाले आंकडाकोष में ०.९ की औसतन परिशुद्धता विभिन्न लम्बाइयों की संकेतावलीओं पर लगभग ~४ से ~२९६ गुना अधिक तीव्रता से प्राप्त करता है। १ अरब प्रतिदर्श वाले आंकडाकोष में यह परिशुद्धता MIH की तुलना में लगभग ~४ से ~१९ गुना तीव्रता से प्राप्त हो जाती है। इसके पश्चात, ComBI को एकक कोशिका RNA अनुक्रमण (scRNA-seq) के आंकड़ों द्वारा सामान कोशिकाओं की खोज करने वाले खोज-यन्त्र के रूप में प्रस्तुत किया गया और इसके प्रदर्शन की तुलना अत्याधुनिक scRNA-seq खोज-यन्त्र, Cellfishing.jl से की गयी जो की MIH पे आधारित है। ComBI कई खातों में Cellfishing.jl से बेहतर प्रदर्शन करता है और इस खोज के अभ्यास में लगभग ~२ से ~१३ गुना की तीव्रता भी प्राप्त करता है।

इसके पश्चात हम अपना ध्यान समानता संरक्षण करने वाली हैशिंग विधियों का उपयोग करके एक वर्गीकर्ता को विकसित करने में करते हैं। इस प्रकार के वर्गीकर्ताओं के निर्माण के लिए हैशिंग कलन विधियों द्वारा सीखी गयी संरचना बयस' वर्गीकर्ताओं (bayes' classifier) के साथ संयुक्त रूप में उपयोग की जा सकती है। इस शोध प्रबंध में हमने तीन मूलभूत विस्तार-विभाजन आधारित हैशिंग कला विधियों का उपयोग करके वर्गीकर्ताओं का निर्माण किया और उनके गुंडों और अवगुंडों का विश्लेषण किया है। यह विश्लेषण हमें एक वृक्ष आधारित हैशिंग वर्गीकर्ता के निर्माण के लिए प्रेरित करता है। यह शोध प्रबंध वृक्ष आधारित समवेत हैशिंग वर्गीकर्ता का उल्लेख करता है जिसे मार्गदर्शित आकस्मिक वर्गीकर्ता (GRAF) के नाम से सम्बोधित किया जाता है। GRAF एक परोक्ष विभाजन तकनीक है जो की वृक्ष आधारित वर्गीकरणों का निर्माण सार्वत्रिक विभाजनों के आधार पे करती है। इस प्रक्रिया में अन्य स्थानीय विभागों का पुनः वर्गीकरण सार्वत्रिक विभाजनों के आधार पे होता है जो की वर्तमान तकनीकों से भिन्न है। हम यह भी दिखाते हैं की GRAF, निर्णय वृक्ष (decision tree) और वर्धन कला विधियों (ensemble algorithms) के मध्यांतर को पूर्ण करता है। प्रयोगों से यह संकेत भी मिलता है कि GRAF सामान्यीकरण त्रुटियों की सीमा को भी कम करता है। ११५ मापदंड आंकड़ा समूहों के परिणाम दिखाते हैं की GRAF अधिकतर आंकड़ा समूहों पे उत्तीर्ण परिणाम देता है। इस शोध प्रबंध में अनिरीक्षित GRAF (uGRAF) को भी विकसित किया गया है। मौलिक रूप से GRAF उच्च आंकड़ा जटिल क्षेत्र में अधिक अधिसमतल उत्पन्न करता है और इस घटना को एक प्रतिदर्श को शुद्ध रूप से वर्गीकृत करने के लिए आवश्यक समतलों की संख्या द्वारा दर्शाया जाता है। इस माप का उपयोग महत्वपूर्ण प्रतिदर्श के चयन के लिए किया जा सकता है। इस प्रक्रिया का उपयोग आंकड़ा समूहों के सन्निकटन के लिए भी किया जा सकता है। इस सिद्धांत को स्थापित करने के लिए कृत्रिम और UCI आंकड़ा समूहों के साथ एक व्यापक

अनुभवजन्य मूल्यांकन किया गया है। प्रस्तावित कार्यप्रणाली की तुलना अत्याधुनिक महत्वपूर्ण प्रतिदर्श के चयन करने वाली कला विधियों से की गयी है। इस शोध प्रबंध में शह सदिश प्रतिदर्श (SVs) और GRAF के द्वारा चिन्हित महत्वपूर्ण प्रतिदर्श के मध्य एक समानता भी स्थापित की गयी है।

इसके पश्चात हम दिखाते हैं की प्रतिदर्श के सीखे हुए पड़ोस का उपयोग एक मापनीय तरीके से उसके परिवेश में भ्रम का अनुमान लगाने के लिए किया जाता है। प्रति-प्रतिदर्श वर्गीकरणीयता का अनुमान लगाने के लिए हमने uGRAF और ComBI का उपयोग किया। अनुमानित मूल्यों का एक अनुभवजन्य मूल्यांकन प्रस्तुत किया गया है। हम दिखाते हैं कि कैसे प्रति-प्रतिदर्श वर्गीकरण क्षमता का उपयोग कर्क रोगी की उत्तरजीविता का अनुमान लगाने के लिए किया जा सकता है।

कर्क रोग एक सनजीन का रोग है। सनजीन के उत्परिवर्तन जिनके परिणामस्वरूप कर्क रोग हो सकता है उन्हें विरासत में प्राप्त किया जा सकता है, पर्यावरणीय तत्वों से अधिग्रहित किया जा सकते हैं या यादृच्छिक प्रतिकृति त्रुटियों के परिणामस्वरूप हो सकता है। कार्सिनोजेनेसिस (carcinogenicity) के प्रवेश होने के बाद उत्परिवर्तन फैलता रहता है और कर्क सनजीन में अत्यधिक बदलाव आता है। अधिकांश कर्क रोग सम्बन्धी दैहिक उत्परिवर्तन, जनन रेखा सम्बन्धी उत्परिवर्तन या अन्य गैर-दैहिक उत्परिवर्तन से अविभेद्य होते हैं। आज तक केवल एक छोटे से उत्परिवर्तन उपसम्मुचय का विश्लेषण किया गया है। उत्परिवर्तनो का प्रयोगशाला सत्यापन अत्यधिक कठिन और श्रमिक कार्य है। उत्परिवर्तन पार्श्वचित्र में रोगीयों के मध्य अत्यधिक भिन्नता यह समस्या को और प्रख्यात करती है और रोग विशेष उत्परिवर्तनो को पहचानना और मुश्किल हो जाता है। इसीलिए बहुत सारे हानिकारक उत्परिवर्तन आज भी अज्ञात है। यह शोध प्रबंध एक नयी विधि कोडन स्विचेस का सतत प्रतिनिधित्व (CRCS) का परिचय देता है। यह डीप-लर्निंग (Deep learning) आधारित दृष्टिकोण सनजीन उत्परिवर्तनो के संख्यात्मक सदिश प्रतिनिधित्व का उत्पादन करने में सक्षम बनाता है। यह प्रतिनिधित्व हमें विभिन्न प्रकार के मशीन लर्निंग (machine learning) आधारित कार्यों को करने के सक्षम बनाता है। हम दिखाते हैं कि कैसे CRCS को तीन अलग-अलग तरीकों से इस्तेमाल किया जा सकता है। सबसे पहले हम यह दिखाते हैं की CRCS तकनीक का उपयोग उपयुक्त प्रकृतिस्थ प्रतिदर्श की अनुपस्थिति में दैहिक उत्परिवर्तनो की पहचान करने में किया जा सकता है। दूसरा, प्रस्तुत की गई विधि चालक जीनों को खोजना और उनका अध्ययन करना संभव बनाती है। अंत में, हमने CRCS के साथ एक अनुक्रम वर्गीकारक को जोड़कर उत्परिवर्तनों का एक संख्यात्मक प्रतिनिधित्व बनाया। ये प्रतिनिधियों का उपयोग प्रति प्रतिदर्श के व्यक्तिगत उत्परिवर्तनों के आंकलन में किया जा सकता है। यह आंकलन ब्लैडर यूरोथेलियल कार्सिनोमा (BLCA), हेपेटोसेलुलर कार्सिनोमा (HCC), और ग्लियोब्लास्टोमा मल्टीफॉर्म (GBM) के रोगियों की जीवन सम्भावना का पूर्वानुमान लगाने में सक्षम था। संछेप में, हम CRCS को व्यक्तिगत कर्क रोग के उत्परिवर्तनों को कार्यात्मक महत्व के विश्लेषण के लिए एक मूल्यवान अभिकलनीय उपकरण के रूप में प्रस्तावित करते हैं।

Contents

| | |
|---|--------------|
| Certificate | i |
| Acknowledgements | iii |
| Abstract | v |
| Hindi Abstract | vii |
| List of figures | xv |
| List of tables | xxv |
| List of algorithms | xxvii |
| List of abbreviations | xxix |
| 1 Introduction | 1 |
| 1.1 Scope and objectives | 2 |
| 1.2 Space-partitioning-based algorithms | 4 |
| 1.2.1 Space-partitioning-based hashing and nearest neighbor search | 4 |
| 1.2.2 Local space-partitioning-based algorithms - A case for classification | 6 |
| 1.3 Genomics of cancer | 7 |
| 1.3.1 <i>Central dogma</i> of molecular biology | 7 |
| 1.3.2 Variations in DNA: Cause of cancer | 7 |
| 1.3.3 Mutational landscape of cancer | 9 |
| 1.4 Organization of the thesis | 10 |
| 1.5 Conclusion | 12 |
| 2 The ComBI: A bit-string tree for fast approximate search in hamming space | 13 |
| 2.1 Introduction | 13 |

| | | |
|----------|--|-----------|
| 2.2 | The Compressed BST of Inverted hash tables (ComBI) | 15 |
| 2.2.1 | Motivation | 16 |
| 2.2.2 | Details of ComBI | 17 |
| 2.2.2.1 | Construction of ComBI | 18 |
| 2.2.2.2 | Search in ComBI | 19 |
| 2.2.2.3 | Search in ComBI is approximate | 20 |
| 2.2.2.4 | Convergence of search in ComBI | 21 |
| 2.2.2.5 | Online construction of ComBI | 22 |
| 2.2.2.6 | Scaling ComBI on a large data. | 24 |
| 2.3 | Implementation, Experiments and Results | 26 |
| 2.3.1 | Bit code generation | 26 |
| 2.3.2 | ComBI implementation details | 26 |
| 2.3.3 | Dataset description | 26 |
| 2.3.4 | Experiment design | 27 |
| 2.3.5 | Performance metric | 28 |
| 2.3.5.1 | Nearest samples in a hamming space | 28 |
| 2.3.5.2 | False discovery rate | 28 |
| 2.3.5.3 | Speedup | 29 |
| 2.3.6 | Performance comparison | 29 |
| 2.3.6.1 | Speed-up analysis | 29 |
| 2.3.6.2 | Quality of approximate search | 30 |
| 2.3.7 | Comments on performance tuning | 34 |
| 2.3.8 | Reduction in memory usage in comparison to IBST | 34 |
| 2.3.9 | ComBI as single cell search engine | 35 |
| 2.3.9.1 | Pre-processing and hashing of gene expressions | 35 |
| 2.3.9.2 | Experimental setup for comparison | 36 |
| 2.3.9.3 | Results | 37 |
| 2.4 | Discussion | 38 |
| 2.5 | Conclusion | 39 |
| 3 | Generalized hashing classifier | 41 |
| 3.1 | Introduction | 41 |
| 3.2 | Hashing classifier | 42 |
| 3.2.1 | General idea | 43 |

| | | |
|----------|--|-----------|
| 3.2.1.1 | Choice of \mathcal{A} | 44 |
| 3.2.1.2 | Choice of \mathcal{B} | 44 |
| 3.2.2 | Some sample hashing classifier | 46 |
| 3.2.2.1 | Sketching-based classifiers | 46 |
| 3.2.2.2 | Projection hash-based classifiers | 50 |
| 3.2.2.3 | Binary hashing-based classifiers | 51 |
| 3.2.3 | Need for a tree arrangement | 53 |
| 3.2.4 | Conclusion | 56 |
| 4 | The GRAF: A bit-string tree as a hashing classifier | 57 |
| 4.1 | Introduction | 57 |
| 4.2 | Related Work | 59 |
| 4.3 | Guided Random Forest (GRAF) | 60 |
| 4.4 | GRAF as \mathcal{A} - \mathcal{B} formulation | 65 |
| 4.4.1 | Choice of \mathcal{A} | 65 |
| 4.4.2 | Choice of \mathcal{B} | 65 |
| 4.5 | Implementation details | 66 |
| 4.5.1 | Heuristic for region search | 67 |
| 4.5.2 | CPU vs GPU implementation | 68 |
| 4.5.3 | Time Complexity | 68 |
| 4.5.3.1 | Training time complexity of a tree | 69 |
| 4.5.3.2 | Testing time complexity of a tree | 70 |
| 4.5.4 | Model Size | 70 |
| 4.5.5 | Space Complexity | 71 |
| 4.6 | Relationship of GRAF with boosting | 71 |
| 4.7 | Feature selection using GRAF | 71 |
| 4.8 | The Unsupervised GRAF (uGRAF) | 73 |
| 4.9 | Simulation Study | 74 |
| 4.10 | Results | 79 |
| 4.10.1 | Data generation with Weka for Bias-variance tradeoff | 79 |
| 4.10.2 | Bias-variance tradeoff | 81 |
| 4.10.3 | Performance comparison on UCI datasets | 83 |
| 4.11 | Conclusion | 95 |

| | | |
|----------|--|------------|
| 5 | Utilization of neighborhood learned by the bit-string trees | 97 |
| 5.1 | Introduction | 97 |
| 5.2 | GRAF as data approximator | 98 |
| 5.2.1 | Empirical convergence of sensitivity scores | 104 |
| 5.3 | Unsupervised GRAF and ComBI for classifiability computation | 105 |
| 5.3.1 | Per sample classifiability computation | 106 |
| 5.3.2 | Differentiation between sensitivity and classifiability | 107 |
| 5.3.3 | Use-case of per-sample classifiability | 108 |
| 5.4 | Conclusion | 109 |
| 6 | Learning nucleotide sequence context of cancer mutations and its applications | |
| | in survivability | 111 |
| 6.1 | Introduction | 111 |
| 6.2 | Datasets, Methods, and Experiments | 113 |
| 6.2.1 | Description of datasets | 113 |
| 6.2.2 | Pruning of the coding variants | 114 |
| 6.2.3 | Codon switch sequences | 115 |
| 6.2.4 | Continuous embedding of codon switches | 116 |
| 6.2.5 | Cross-chromosome sequence similarity analysis | 118 |
| 6.2.6 | Variant classification | 118 |
| 6.2.7 | Other methods for mutation annotation | 120 |
| 6.2.8 | Other available embeddings | 120 |
| 6.2.9 | Other available architectures | 120 |
| 6.2.10 | Comparing cBioPortal predictions with dbSNP predictions | 122 |
| 6.2.11 | Classifiability for survival analysis | 122 |
| 6.3 | Results | 123 |
| 6.3.1 | Learning numeric representation of mutations | 123 |
| 6.3.2 | CRCS exposes inherent diversity of chromosomes | 125 |
| 6.3.3 | Classifying cancerous and non-cancerous mutations | 127 |
| 6.4 | Comparison of CRCS-based approach with the existing best practice architectures | 130 |
| 6.4.1 | BLAC score assists in driver gene exploration | 131 |
| 6.4.2 | BLAC enable survival risk stratification in different cancer types | 136 |
| 6.5 | Discussion | 138 |
| 6.6 | Conclusion | 140 |

| | | |
|----------|--|------------|
| 7 | Conclusions and future work | 141 |
| 7.1 | Future work | 143 |
| 7.1.1 | ComBI & uGRAF for clustering in hamming space | 143 |
| 7.1.2 | Pan chromosome BLAC | 144 |
| 7.1.3 | Better architectures to handle extreme length variations | 144 |
| 7.1.4 | New horizon to learn better embedding | 144 |
| 7.1.5 | Extended switch dictionary and their embedding | 145 |
| 7.1.6 | Application of CRCS in influenza and other diseases | 147 |
| | List of publications | 163 |
| | Brief biodata of author | 165 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Example of space-partitioning-based hash functions. A) Projection hash which assigns hash code by quantizing the randomly projected values. B) Sketching, which assigns hash code by thresholding every feature. C) Binary hash randomly divides the space and assigns bit code to every sample based on their location in the space. | 4 |
| 1.2 | Space-partitioning strategy example. Local partitioning axis aligned splits - This strategy followed by Random Forest (RF), Extremely Randomized Trees (ET), kd-tree etc. Local partitioning oblique splits - This strategy followed by Oblique Tree (OT), RPTrees etc. Global partitioning oblique splits - This strategy is followed by hashing techniques and Guided Random Forest (GRAF), the classification algorithm proposed in this thesis. There is another strategy to have global partitioning axis aligned splits - This strategy is followed by sketching. | 6 |
| 1.3 | Type of DNA mutations based on amino acid changes. | 8 |
| 1.4 | Mutational landscape of cancer is sparse. There are ~ 3 billion base pairs in the human genome. Every one of them can mutate, causing a very large event space. However, the number of mutations that get mutated in an individual's lifetime is relatively small. Also between two individuals, there is much less overlap on the acquired mutations, thus making the mutation frequency in population very low. | 9 |
| 1.5 | Organization of thesis. The chapter name, chapter flow, and headlines of every chapter in the thesis. | 11 |
| 2.1 | Visualization of the resulting view after merger of regions. A) Geometrical representation of space partitioning via LSH. B,C) 1-NN approximation of space via IBST, and ComBI, respectively. | 18 |
| 2.2 | Tree representations A) IBST. B) ComBI. | 19 |

| | | |
|------|--|----|
| 2.3 | The methodology of online ComBI construction. The highlighted nodes are affected during insertion. A) Insertion of first bit code. B) Update in ComBI. | 22 |
| 2.4 | Distribution of hamming distances (HD) of returned neighbors by ComBI from the farthest exact NN. FDR represents the fraction of neighbors returned by ComBI that are not part of the exact NN set. Each violin plot corresponds to the configuration shown in Table 2.2 for the combination of bits and NNs for the SIFT-1B data set. | 32 |
| 2.5 | Visualization of approximate search. Visualization of top 10 nearest neighbors returned by ComBI and MIH on 5 random samples for 256 bits long bit code. . . . | 33 |
| 2.6 | Convergence on hyperparameters on 80-M tiny dataset. Impact of tunable parameters of ComBI (T and m), on precision and time, for 100 NNs on the 80M-tiny image data set. | 34 |
| 2.7 | Convergence on hyperparameters on SIFT-1B dataset. Impact of tunable parameters of ComBI (T and m), on precision and time, for 100 NNs on the SIFT-1B dataset. | 35 |
| 2.8 | Performance of ComBI and Cellfishing.jl on baron2016 dataset. The dataset has 8569 samples. With increasing length of bit code and higher number of tables ComBI has better performance. For 64, 128, and 256 bits ComBI has ~ 2 , ~ 2 , and ~ 4 times speed-up in search time with 4 tables, respectively. Similarly, ComBI has speed-up of ~ 2 , ~ 3.8 , and ~ 8 for 64, 128, and 256 bits with 16 tables, respectively. | 36 |
| 2.9 | Performance of ComBI and Cellfishing.jl on plass2018 dataset. The dataset hash 21612 samples. With increasing length of bit code and higher number of tables ComBI has better performance. For 64, 128, and 256 bits ComBI has ~ 1.24 , ~ 1.68 , and ~ 4 times speed-up in search time with 4 tables, respectively. Similarly, ComBI has speed-up of ~ 2.5 , ~ 5.3 , and ~ 12.7 for 64, 128, and 256 bits with 16 tables, respectively. | 37 |
| 2.10 | Performance of ComBI and Cellfishing.jl on shekhar2016 dataset. The dataset hash 27499 samples. With increasing length of bit code and higher number of tables ComBI has better performance. For 64, 128, and 256 bits ComBI has ~ 2 , ~ 1.4 , and ~ 5.6 times speed-up in search time with 4 tables, respectively. Similarly, ComBI has speed-up of ~ 2 , ~ 5 , and ~ 12.3 for 64, 128, and 256 bits with 16 tables, respectively. | 38 |

| | | |
|-----|---|----|
| 3.1 | Components of a hashing classifier. In summary, first, select a hashing function, then select the search technique, and finally identify the appropriate method to interpret the bins to perform the task. | 43 |
| 3.2 | Example hashing classifiers. An illustration of components to build hashing classifiers for three possible hashing methods, namely projection-based hashing, sketching, and binary hashing. The projection-based hashing can be converted into a classifier by choosing BST as the search technique. The bins can be interpreted with weighted average or exponential decay. Similarly, a binary search tree or hamming tree can be employed to convert the sketching technique into a classification. The bins from the sketching technique can be interpreted with the weighted average of exponential decay. The binary hash can be associated with a hamming tree for nearest neighbor search, and exponential decay can be employed to interpret bins. | 46 |
| 3.3 | Qualitative comparison of hashing classifiers. Among the three example classifier, the binary hash-based hashing classifier has the most desired properties. Sketching-based hashing classifier is at the second. Projection-based hashing classifier is the most undesirable. | 54 |
| 4.1 | An overview of the creation of high variance instances in GRAF. Every instance consists of sub-spacing the dataset in a uniformly sampled feature space. A random hyperplane is generated for the sub-spaced samples. It assigns a bit 0/1 to every sample. A pure (impure) region is a region containing all (some) samples of the same class. Amongst these regions, the most impure region affects the generation of the next hyperplane. This hyperplane is extended to the other region as well, if it improves the purity of subsequent regions in that space. This generation of hyperplanes is continued until all regions are maximally purified. At an intermediate stage, regions are either pure or impure. To increase the confidence of classification, the above process is repeated to create L high variance instances. | 58 |
| 4.2 | The division of space in GRAF is represented by a tree. A region containing a subset of samples is defined by its unique combination of hyperplanes. However, these hyperplanes may affect the formation of other regions. The process terminates once space is maximally divided such that the impurity in any region cannot be reduced any further. Every resultant region corresponds to a leaf node in the tree, represented by a dot in the figure. (A triangle denotes an impure region that may be dichotomized further.) | 67 |

| | | |
|-----|--|----|
| 4.3 | A heuristic for faster run-time of GRAF. The perpendicular distance of the mean point from plane A (d_1) is greater than Radius of Influence (ROI). Hence, Plane A does not dichotomize the region. The perpendicular distance of the mean point from plane B (d_2) is less than ROI. Hence, plane B may dichotomize the region. If the perpendicular distance is equal to ROI, it is considered as not dichotomized. | 68 |
| 4.4 | The performances and model size comparison of methods on simulated binary and multiclass examples with high concept complexity. The high concept complexity means that all the features are independent of each other. The number of features varies from 3 to 15. A, B) For both binary and multiclass examples, GRAF has the highest values of Cohen’s kappa coefficients, closely followed by Oblique Tree (OT). C, D) However, for similar performance measures, the overall model size of OT is much higher when compared with GRAF. | 77 |
| 4.5 | The performances and model size comparison of methods on simulated binary and multiclass examples with low concept complexity. The low concept complexity means that only a few features are relevant and independent. The number of features varies from 3 to 15. A, B) For both binary and multiclass examples. In these settings performances of all methods are comparable. C, D) The trend in the model size is the same as the high concept complexity datasets. | 78 |
| 4.6 | The run-time complexity analysis of high concept complexity datasets. The training and testing time of different methods is compared on a simulated dataset. A, B) GRAF’s GPU implementation significantly reduces the training time for both binary and multiclass examples. C, D) GRAF’s testing time is comparable with other methods. | 79 |
| 4.7 | The run-time complexity analysis of low concept complexity datasets. The training and testing times of different methods are compared on a simulated dataset projected by using a random matrix. A, B) The GPU implementation of GRAF significantly reduces its training time for both binary and multiclass examples. C, D) The testing time of GRAF is comparable with other methods. | 80 |
| 4.8 | Bias-variance analysis with an increasing number of estimators (trees) in a classifier. For both binary A - C) and multi-class D - F) datasets with 10 centroids, the number of estimators is increased from 2 to 150, while fixing the number of dimensions to be sampled ($M = n/2$). As the number of estimators is increased, bias, error, and variance rapidly saturate. | 82 |

- 4.9 **Bias-variance analysis with an increasing number of dimensions (features) selected from a given feature space in a classifier.** For both binary **A - C)** and multi-class **D - F)** datasets with 10 centroids, M is increased from 2 to 10, while fixing the number of estimators to be assembled ($L = 100$). For GRAF, when the dimension of the sub-space is large enough to distinguish samples of different classes, bias and variance saturate and converge to their minimum. With increasing dimensionality of the sub-space, misclassification error continues to decrease and rapidly saturates to its minimum. 83
- 4.10 **Bias-variance analysis with increasing samples in a training set.** For both binary **A - C)** and multi-class **D - F)** datasets with 10 centroids, the number of samples is increased from 200 to 2500, while fixing the number of dimensions to be sampled ($M = n/2$) and the number of estimators as $L = 100$. As the cardinality of the training set is increased, bias-variance continues to decrease, and the misclassification error continues to decrease and may saturate to its minimum. 84
- 4.11 **One-sided paired Wilcoxon signed-rank test on Cohen's kappa score.** Each method is paired with every other method, and p-value was computed for the null hypothesis 'left method = right method'. Null hypothesis is rejected in favour of hypothesis 'left method > right method', if the corrected p-value is below a certain significance level. The method on the left side (of comparison) is placed on the x-axis, and the method on the right side is placed on the y-axis. Each cell represents the corrected p-value. Hence, every column represents the significance of the kappa score for a method when compared with other methods. Suppose the corrected p-value is less than a certain significance level in a cell. In that case, the null hypothesis is rejected, and the method on the x-axis will be assumed to perform better than the corresponding method on the y-axis. The numerals in the x-axis represent the average Friedman ranking of the method. 86

| | | |
|-----|--|-----|
| 5.1 | Assessment of performance of GRAF's sensitivity on simulated binary and multi-class datasets. A, B, and E) represent simulated datasets with binary classes. B, D, and F) represent simulated multi-class datasets. The classes are arranged in different patterns, concentric circles, pie-charts, and XOR representations, in A-B), C-D), and E-F) , respectively. For each of these datasets, the distribution of sensitivities computed using GRAF has been shown in column <i>Sensitivity</i> . A point with higher sensitivity indicates that it is more important for data approximation. The other columns U25%, P25%, and S25%, compare the performances of data approximation using only 25% of the total samples, sampled using a uniform distribution, distribution defined by GRAF's sensitivity, and the points with the highest values of sensitivities, respectively. The regions with the most confusion are best approximated using points with the highest sensitivities. . | 100 |
| 5.2 | Performance evaluation of Random Forest (RF) and GRAF , with increasing fraction of samples used for training, sampled according to uniform distribution (U), their sensitivities (P), and their decreasing order of sensitivities (S). The points sampled using a distribution defined by their sensitivities perform comparable or better when compared with points sampled using a uniform distribution. Also, as points are added in the decreasing order of their sensitivities, the accuracy on the test set converges and reaches its maximum with only a fraction of points with high sensitivities. The trends in results are similar, irrespective of the method used for classification. | 101 |
| 5.3 | An analogy between support vectors and points with high sensitivities. The distribution of probabilities (5.4) associated with support vectors has been compared with that of a fraction of points with high sensitivities, and the distribution of probabilities is associated with all points. It can be concluded that points with higher sensitivities coincide with the support vectors with higher values of weights. | 102 |
| 5.4 | Convergence of sensitivity values. Change in sensitivity score almost reaches 0 as the number of hyperplanes increases. | 105 |
| 5.5 | Per-sample classifiability on simulated dataset As expected the samples near the decision boundary have lower classifiability while the inner sample has higher classifiability. | 108 |

| | | |
|-----|--|-----|
| 6.1 | Variant distribution in COSMIC data (v89). Single base substitutions are the most frequent type of mutations in the database. While the complex mutations are the rarer ones. | 115 |
| 6.2 | Filtering criteria to handle computation overhead. A) mRNAs whose switch sequences were ≤ 1500 long were kept for the analysis. B) Genes that have ≥ 200 mutations were kept for analysis. | 119 |
| 6.3 | An overview of learning Continuous Representation of Codon Switches (CRCS). A) The procedures include two steps: i) choosing variants that are located in exon regions; and (ii) creating the codon switch sequence. A codon switch is described as a directional pair of codons that includes an alternative codon (a sequence derived from an interest genome) and a reference codon (a sequence derived from the reference genome). Included is a toy example that shows how to build codon switch sequences. A codon switch sequence's index in the codon switch dictionary is indicated by the number next to it. B) A center codon switch is selected probabilistically. Two types of tuples are built for the chosen center codon switch. Tuples belonging to a center codon switch's context window are marked with a 1; a few codon switches from outside the context are also selected; their tuples are marked with a 0; C) A classifier is trained to classify these tuples. Input layer weights of this network behave as codon switch embeddings. D) tSNE plot of learned embeddings. E) Distribution of different codon switches on the tSNE plots. Interestingly, similar codon switches tend to cluster far from opposite codon switches ($G > A$ and $A > G$, $G > T$ and $T > G$, $A > C$ and $C > A$, $C > T$ and $T > C$). | 124 |
| 6.4 | CRCS embeddings reveals exclusive nature of chromosomes. A) tSNE projections of the embeddings learned independently for all the chromosomes. The embeddings are clearly segregated, indicating heterogeneity in nucleotide sequence patterns. B) Spearman correlation of unigram frequencies across chromosomes. Chromosomes are found to give rise to some tight clusters. C) Spearman correlation of bigram frequencies in chromosomes. D) Chromosomes are described as trigrams. Chromosomal similarities fade away with an increase in the sequence length. . . . | 126 |

| | | |
|-----|--|-----|
| 6.5 | <p>Classification of cancerous and non-cancerous variants. A) Deep learning architecture, used for CRCS-based classification of ExAC/COSMIC variants. B) Precision-Recall (PR) curve for the BLAC after 200 epochs. The red and green curves indicate the performance of SIFT and Polyphen2, respectively. Validation performances were measured on fake alteration classes, constructed by randomly splitting cancer/non-cancer alterations into two equal-size groups. The black dashed line represents the performance of the fake test set created from COSMIC data. Similarly, the blue dashed line is for ExAC data. Both PR curves thus obtained, as expected, collapsed on the 0.5 precision line. C) Boxplots depict the distribution of prediction scores (probability of being a cancer alteration), assigned to the ExAC and COSMIC alterations, in the validation set (across all folds). D) Similar trends are observed for non-pathogenic dbSNP alterations and mutations found in cancer patients from Met and cBioPortal. Scores on these datasets were predicted using the model trained on the full dataset.</p> | 128 |
| 6.6 | <p>Evaluation of model trained on chromosome X against chromosome 22. As expected, model performance deteriorated. This reduction in performance is due to the fact that the nucleotide distribution in a chromosome is different. Thus a model trained on one chromosome can not be used on the other chromosome without re-training/fine-tuning. Also, the complexity of every chromosome is different, thus same deep learning architecture may not suitable for other chromosomes.</p> | 131 |
| 6.7 | <p>Performance comparison of BLAC scores with other deep learning architectures. A) Precision-Recall plot of the predictions obtained from the model trained with CRCS embeddings and dna2vec embeddings. B) Comparison of the distribution of scores obtained from the model. dna2vec does not have any discriminating power (Mann-Whitney U-test P-value = 1). C) Precision-Recall plot of the predictions obtained from other deep learning models, DanQ, DeepSea, and HeartENN. Compared to our proposed model trained with CRCS, other models have inferior performance. D) Comparison of the distribution of scores obtained by models. DanQ does not have any discriminating power (Mann-Whitney U-test P-value = 1). Other models have a different distribution of scores on ExAC and COSMIC. Mann-Whitney U-test P-value for DeepSea and HeartENN is 2.7×10^{-20} and 9.08×10^{-9} respectively. Our model with CRCS has the most differentiating power (Mann-Whitney P-value is almost near 0).</p> | 132 |

- 6.8 **Driver gene analysis and exploration. A)** Boxplots show the distribution of prediction scores assigned to ExAC and COSMIC alterations for the known driver genes from the validation set (across all folds). In the figure, 5 stars represent a P -value less than $5e^{-15}$. Values in the range $[5e^{-15}, 5e^{-12})$ are represented by 4 stars. Similarly, values in the range of $[5e^{-12}, 5e^{-9})$, $[5e^{-9}, 5e^{-6})$, and $[5e^{-6}, 5e^{-2})$ are represented by 3, 2, and 1 stars, respectively. **B)** Heatmap shows the genes (in black) that have been marked significant most frequently, across cancer types. For a given cancer type in cBioPortal, a gene was marked significant if the BLAC scores of the reported mutations were significantly elevated as compared dbSNP variants. The colors in the top row show the organ of cancer. Gene marked with * are known driver genes. **C)** Heatmap depicting the cluster-wise enrichment of the prominent biological functions in the indicated cancer types. Of note, the selected cancer types harbored a number of mutational genes identified using the CRCS-based approach. Cancer types that displayed significantly divergent risk groups include Skin Cutaneous Melanoma (SKCM), Lung Adenocarcinoma (LUAD), and Undifferentiated Endometrial Carcinoma (UEC). The scale bar represents the negatively log-transformed (base 10) P -values. 133
- 6.9 **BLAC score distribution of remaining driver genes.** Boxplots show the distribution of the prediction scores assigned to ExAC and COSMIC alterations for the remaining known driver genes from the validation set (across all folds), except the top 10. Top 10 values are present in Figure 6.8. Stars have the same meaning as in Figure 6.8. 134
- 6.10 **Heatmap depicting the cluster-wise enrichment of the prominent biological functions in the indicated cancer types.** Of note, the selected cancer types harbored the number of mutational genes identified using BLAC. Cancer types include Skin Cutaneous Melanoma (SKCM), Lung Adenocarcinoma (LUAD), Undifferentiated Endometrial Carcinoma (UEC), Lung Squamous Cell Carcinoma (LUSC), Head-Neck Squamous Cell Carcinoma (HNSC), Urothelial Bladder Carcinoma (BLCA), and Non-small-cell Lung Carcinoma (NSCLC). The scale bar represents the negatively log-transformed (base 10) P -values. 135

| | | |
|------|--|-----|
| 6.11 | Overview of survivability analysis using classifiability. On the combined dataset of dbSNP and COSMIC, attention vectors were extracted from the BLAC network. These scores are used for classifiability computation. Then groups pertaining to COSMIC were extracted along with their classifiability score and used for survivability analysis. | 137 |
| 6.12 | Survival risk stratification based on classifiability. Patients with lower average BLAC scores in Bladder Urothelial Carcinoma (BLCA), a subtype of bladder cancer, has better survival. Similar trends are also visible in Hepatocellular Carcinoma (HCC), a subtype of brain cancer, and Glioblastoma Multiforme (GBM), a subtype of lung cancer. | 138 |
| 7.1 | Extended codon switch dictionary. The strategy presented here facilitates the representation of any kind of mutation via these elements. Since switches are constructed on a codon, only three types of substitutions are considered: i) Single base substitution. ii) Double base substitution. iii) Triple base substitution. With the introduction of \$, the representation can handle insertions, deletions, and complex indels. | 146 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | Dataset Description for empirical evaluation of ComBI. | 27 |
| 2.2 | The column of <i>time(s)</i> contains an average of the nearest neighbor search time for all query samples. For ComBI, the results are reported for a configuration that takes minimum time to achieve a precision of ≥ 0.90 in a hamming space. For MIH, a configuration with the least time is selected. | 30 |
| 2.3 | The column of <i>time(s)</i> contains an average of the nearest neighbor search time for all query samples. For ComBI, the results are reported for a configuration that takes minimum time to achieve a precision of ≥ 0.95 in a hamming space. For MIH, a configuration with the least time is selected. | 31 |
| 2.4 | A comparison of node counts in IBST and ComBI. | 35 |
| 4.1 | A simulation study to discuss the design aspects of GRAF. The number of features varied from 3 to 15. For a given value of the feature, both binary and multiclass examples were generated. For every configuration, 10 different trials were performed to generate samples. The total number of samples vary from ~ 25 - $\sim 115,000$ across all trials. The train-test split consists of 70-30% of the total samples. The total number of principal components which explains 90% of the total variance in the dataset differs when it is projected on a random matrix. | 75 |
| 4.2 | Data statistics of 115 UCI datasets. The total number of samples across all datasets varies from 24 to $\sim 130k$. The count of features across all datasets varies from 3 to 262. | 87 |
| 4.3 | The performance of methods is compared on 115 UCI datasets using Cohen's kappa coefficient. | 91 |

| | | |
|-----|--|-----|
| 5.1 | Equivalence between the reduced training set and support vectors. For a given test set, the SVM model is learned using two different sets. First, an SVM model is trained using all the samples in the training set. Its accuracy on the test set is then evaluated (column <i>% SVM Accuracy</i>), and information about the support vectors is recorded (column <i>% SVs</i>). Separately, an SVM model is trained using points from the reduced training set (column <i>% SVM accuracy on reduced training set</i>). For GRAF and SVIS, the size of the reduced training set is the same as that of support vectors. For NPPS, the reduced training set consists of samples with high heterogeneity values in their neighborhood (column <i>%size of reduced training set</i>). The size of the neighborhood in NPPS is determined by <i>k</i> . An analogy between the reduced training set and support vectors is recorded in column <i>% Overlap with SVs</i> for all three methods. Note that the hyper-parameters for the SVM model in the reduced training set were kept the same as that of the full training set. | 103 |
| 6.1 | Specificity, Sensitivity, and, F1-score values at the threshold of 0.9. This value of threshold was chosen since predictions of all the algorithms are skewed toward high values. These metrics are computed on the predicted scores on mutations reported in ExAC and COSMIC databases. | 130 |
| 7.1 | Frequencies of switches in different variant types formed using ExAC dataset. . . . | 145 |

List of Algorithms

| | | |
|-----|--|-----|
| 2.1 | Construction of Inverted-Hash-Table BST (IBST) | 16 |
| 2.2 | Search in Inverted-Hash-Table BST (IBST) | 17 |
| 2.3 | Construction of Compressed BST of Inverted hash tables (ComBI) by compression of IBST | 20 |
| 2.4 | Search in ComBI | 21 |
| 2.5 | Online Construction of ComBI | 23 |
| 2.6 | ComBI construction in a Compute Cluster | 24 |
| 2.7 | ComBI search in a Compute Cluster | 25 |
| 4.1 | Pseudocode of GRAF | 69 |
| 4.2 | High variance instance of GRAF as boosting | 71 |
| 4.3 | Simulation Data for GRAF benchmarking | 75 |
| 5.1 | Computation of Per-Sample Classifiability using uGRAF and ComBI | 106 |
| 6.1 | Network architecture to learn the Continuous Representation of Codon Switches (CRCS) | 118 |
| 6.2 | Customized neural network for sequence classification - Bidirectional Long Short- Term Memory with Attention & CRCS embeddings (BLAC) | 119 |
| 6.3 | Modified DeepSea Neural Network | 121 |
| 6.4 | Modified DanQ Neural Network | 121 |
| 6.5 | Modified HeartENN Neural Network | 122 |

List of abbreviations

nsr negative sampling rate

ws window size

ADA Adaboost

AP Average Precision

BERT Bidirectional Encoder Representations from Transformers

bi-LSTM Bidirectional Long Short-Term Memory

BLAC Bidirectional Long Short-Term Memory with Attention & CRCS embeddings

BLCA Bladder Urothelial Carcinoma

BST Binary Search Tree

BSTs Binary Search Trees

cBioPortal cBio Cancer Genomics Portal

cfDNA Cell-free DNA

CGI Cancer Genome Interpreter

CNN Convolutional Neural Network

CO2 Continuously Optimized Oblique

ComBI Compressed BST of Inverted hash tables

COSMIC Catalogue Of Somatic Mutations In Cancer

CPU Central Processing Unit

CRCS Continuous Representation of Codon Switches

CSC Cancer-Stem-Cell

ctDNA Circulating Tumor DNA

DBSCAN Density-Based Spatial Clustering of Applications with Noise

dbSNP Single Nucleotide Polymorphism Database

DNA Deoxyribonucleic Acid

DT Decision Trees

eQTL Expression Quantitative Trait Loci

ESCC Esophageal Squamous Cell Carcinoma

ET Extremely Randomized Trees

EWH Error Weighted Hashing

ExAC Exome Aggregation Consortium

FDR False Discovery Rate

GB Gradient Boosting

GBM Glioblastoma Multiforme

GEPSVM Proximal SVM with Generalized Eigenvalues

gnomAD Genome Aggregation Database

GPU Graphics Processing Unit

GRAF Guided Random Forest

GTO Global Tree Optimization

HBST Hamming Distance Embedding BST

HCC Hepatocellular Carcinoma

HD Hamming Distance

HEPTS Hybrid Extreme Point Tabu Search

HNSC Head and Neck Squamous Cell Carcinoma

HWT Hamming Weight Tree

IBST Inverted-Hash-Table BST

indel Insertions & Deletions

intOgen Integrative Onco Genomics

LDA Linear Discriminant Analysis

LGD Local Gene Duplication

LSB Least Significant Bit

LSH Locality Sensitive Hashing

LSTM Long Short-Tem Memory

LUAD Lung Adenocarcinoma

LUSC Lung Squamous Cell Carcinoma

MIH Multi-Index Hashing

MML Maximum Message Length

MPSVM Multi-Surface Proximal SVM

mRNA Messenger RNA

MRPT Multiple Random Projection Trees

MSB Most Significant Bit

NLP Natural Language Processing

NN Nearest Neighbor

NNs Nearest Neighbors

NPPS Neighborhood Property-Based Pattern Selection

NSCLC Non-Small Cell Lung Cancer

OC1 Oblique Classifier 1

OncoKB Oncology Knowledge Base

OOB Out of Bag

OT Oblique Tree

OTs Oblique decision trees

PCA Principal Component Analysis

PolyPhen2 Polymorphism Phenotyping v2

PR Precision-Recall

RAM Random Access Memory

RF Random Forest

ROI Radius of Influence

RP Random Partition

RVFL Random Vector Functional Link Network

scRNA-seq single-cell RNA sequencing

SIFT Sorting Intolerant From Tolerant

SKCM Skin Cutaneous Melanoma

SNV Single Nucleotide Variant

SOM Self-Organizing Maps

SPH Similarity Preserving Hashing

SV Support Vector

SVIS Small Votes Instance Selection

SVM Support Vector Machine

TMB Tumor Mutational Burden

tSNE t-Distributed Stochastic Neighbor Embedding

UCSC University of California, Santa Cruz

UEC Undifferentiated Endometrial Carcinoma

uGRAF Unsupervised GRAF

VCF Variant Call Format

WES Whole Exome Sequencing

WGS Whole Genome Sequencing

XGB Extreme Gradient Boosting (XGBoost)