
DSB 2024 Montpellier
- March 14-15
Workshop Data Structures in Bioinformatics

Data Structures in Bioinformatics
DSB 2024

14-15 March 2024

Montpellier, France

Location and dates

DSB 2024 takes place in Montpellier, South of France, on the Campus St Priest, in the Montpellier University Amphitheater "Jean-Jacques Moreau", Building #2 on March 14-15, 2024.

Lunches and social events are located in Building #5.

Organizers

From team "Methods & Algorithms for Bioinformatics":

- Jordan Moutet (LIRMM - CNRS et Univ. Montpellier)
- Eric Rivals (LIRMM - CNRS et Univ. Montpellier)
- Nikolai Romashchenko (LIRMM - CNRS et Univ. Montpellier)
- Pengfei Wang (LIRMM - CNRS et Univ. Montpellier)

and the LIRMM Communication unit:

- Virginie Fêche
- Elena Demchenko

Sponsors

We are grateful to the sponsors of the 2024 edition:

- The Int'l Training Network (ITN) ALPACA, European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956229.
- The **GDR Bioinformatique Moléculaire** from CNRS

Work in progress for peta-scale sequence exploration

Chikhi Rayan[1]

[1] *Institut Pasteur, Paris, France*

Abstract

Petabytes of valuable sequencing data reside in public repositories, doubling in size every two years. They contain a wealth of genetic information, in particular about viruses, which can help us monitor spillovers and anticipate future pandemics. In this talk, I will present some of the analysis routes and computational tools that are available, or in development, to explore such data. We recently developed a cloud infrastructure, Serratus (Edgar et al, Nature, 2022), to perform petabase-scale sequence alignment. It enabled the discovery of 10x more RNA viruses than previously known, including a new family of coronaviruses. Serratus pioneered peta-scale biological data analysis, yet there is much more to be accomplished in this field. In particular, the development of k-mer methods is of special interest given their simplicity and efficiency.

Keywords: big data, cloud, assembly, indexing

r-indexing without backward searching

Depuydt Lore[1], Goga Adrián[2], Ahmed Omar[3], Baláž Andrej[2], Brown Nathaniel[3], Petescia Alessia[2], Zakeri Mohsen[3], Fostier Jan[1], Gagie Travis[4], Langmead Ben[3], Manzini Giovanni[5], Navarro Gonzalo[6], and Prezza Nicola[7]

[1] *Ghent University - imec, Belgium*
[2] *Comenius University in Bratislava, Slovakia*
[3] *Johns Hopkins University, United States*
[4] *Dalhousie University, Canada*
[5] *University of Pisa, Italy*
[6] *University of Chile, Chile*
[7] *Ca' Foscari University of Venice, Italy*

Abstract

Since the release of BWA-MEM [1], the search for maximal exact matches (MEMs) has been vital in bioinformatics. With the rising popularity of pan-genomes, driven by the exponential growth in genome sequencing, efficiently identifying MEMs within these large datasets remains a key research focus. One standard approach for MEM discovery involves traversing the suffix-tree of the reference. While efficient hashing strategies enable fast navigation along edges (i.e., without looking at most of the characters), challenges arise when getting stuck mid-edge, in which case we need to figure out how far we got and where to go next. Although the conventional solution involves following suffix links to locate suitable nodes for continuation, this iterative process can pose computational overhead. More importantly, the memory requirements for storing suffix trees of large pan-genomes are impractical, even if only suffixes at Burrows-Wheeler transform (BWT) run boundaries are considered. Recently, r-index-based algorithms like MONI [2] have emerged as alternatives for MEM discovery within reference pan-genomes. These algorithms offer a reduced memory footprint (more efficiently proportional to the number of runs in the BWT of the pan-genome). Additionally, they leverage techniques like storing thresholds or using longest common extension queries to efficiently identify new continuation points in the BWT when matches cannot be extended any further. However, they still rely on character-by-character processing of patterns using the LF operation, which can be time-consuming for long reads. Here, we propose a novel index for MEM-finding that occupies comparable space to the r-index augmented for the same task, yet achieves logarithmic time complexity per edge we would descend in the conceptual suffix tree. Our approach eliminates the need for backward stepping or suffix link traversal, thus combining the strengths of suffix tree and r-index-based methodologies while overcoming their limitations.

Keywords: MEM finding, pangenomes, r-index, suffix trees

References

[1] Li H. (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997v2 [q-bio.GN]

[2] Massimiliano Rossi, Marco Oliva, Ben Langmead, Travis Gagie, and Christina Boucher. MONI: A Pangenomic Index for Finding Maximal Exact Matches. Journal of Computational Biology. Feb 2022. 169-187. http://doi.org/10.1089/cmb.2021.0290

Brisk: Exact resource-efficient dictionary for k-mers

Smith Caleb[1], Martayan Igor[1], Dufresne Yoann[2], and Limasset Antoine[1]

[1] *Centre de Recherche en Informatique, Signal et Automatique de Lille - UMR 9189, France*
[2] *Institut Pasteur, Paris, France*

Abstract

The rapid evolution of DNA sequencing has led to an unprecedented surge in the generation of genomic datasets, with modern sequencers now capable of outputting ten terabases per run. However, the community faces an immense challenge in effectively indexing and studying this huge amount of data. Kmer indexing has proven pivotal in handling extensive datasets in a wide variey of applications such as alignment,compression dataset,comparisons, correction, assembly or quantification. Developping efficient and scalable kmer indexing methods is a growing subject. However state of the art structures are predominantly static in nature, requiring complete index reconstruction when incorporating novel data. More recently the need for dynamic indexing structure was identified. However, most existing solutions are pseudo-dynamic, requiring substantial updates to justify the costs of adding new datasets. In practice, applications mostly rely on regular hashtables to associate data to their kmers, resulting in a high kmer encoding rate ranging between 4 and 8 bytes per kmer. In this work, we present Brisk, a drop-in replacement for most kmer dictionary usages. This novel hashmap-like data structure offers exceptional throughput while drastically reducing memmory usage over state of the art synamic associative indexes, especially for large kmer sizes. To do so we rely on hierachical minimizer indexing and memory efficient superkmer representation and introduce novels techniques to quicly probe kmers among a set of superkmers and to handle duplicated minimizers. We are confident that the methodologies developed in this work represent a significant step forward in the creation of efficient and scalable k-mer dictionaries, facilitating their everyday use in genomic data analysis.

Keywords: Indexing, Kmer, Dynamic, High performance computing

Metagenomic classification with maximal exact matches in KATKA kernels and minimizer digests

Draesslerová Dominika[1], Ahmed Omar[2], Gagie Travis[3], Jan Holub[1], Langmead Ben[2], Manzini Giovanni[4], and Navarro Gonzalo[5]

[1] *Czech Technical University in Prague, Czech Republic*
[2] *Johns Hopkins University, United States*
[3] *Dalhousie University, Halifax, Canada*
[4] *University of Pisa, Italy*
[5] *University of Chile, Santiago, Chile*

Abstract

For metagenomic classification, we are given a phylogenetic tree and a collection of reads and, for each read, asked to guess a small subtree from which it was drawn. Although the most popular classifiers, such as Kraken, work with k-mers, recent research indicates that working with maximal exact matches (MEMs) lead to better classifications. For example, we can build an augmented FM-index over the the genomes in the phylogenetic tree concatenated in left-to-right order; for each MEM in a read, find the interval in the suffix array containing the starting positions of that MEM's occurrences in those genomes; find the minimum and maximum values in that interval; and take the lowest common ancestor (LCA) of the genomes containing those positions. This solution is only practical, however, when there are only a few genomes in the phylogenetic tree or they are small.

We consider applying the same solution to three lossily-compressed representations of the genomes' concatenation: first, the KATKA kernel, which discards characters that are not in the first or last occurrence of any K-tuple for a parameter K; second, a minimizer digest; and third, the KATKA kernel of a minimizer digest. With a test dataset, simulated reads and various parameter settings, we checked how many MEM's LCAs were exactly the genome from which the read was generated ("true positives"), for the three compressed representation and for the uncompressed dataset. Surprisingly, we found with for some parameter settings we achieved significant compression while also increasing the fraction of true positives.

Keywords: Metagenomics, taxonomic classification, KATKA, maximal exact matches, string kernels, minimizer digests

Memory-frugal disk-based (phylo-)k-mer filtering for alignment-free phylogenetic placement

Romashchenko Nikolai[1], Linard Benjamin[2], Pardi Fabio[1], and Rivals Eric[1]

[1] *Méthodes et Algorithmes pour la Bioinformatique, LIRMM, CNRS, France*
[2] *Unité de Mathématiques et Informatique Appliquées de Toulouse, France*

Abstract

Phylogenetic placement enables phylogenetic analysis of large collections of newly sequenced DNA, when de novo tree inference is too unreliable or inefficient. Assuming a set of reference sequences and a high-quality reference tree, the idea is to seek most likely placements for the new sequences in that tree. Recent developments include alignment-free methods that rely on inference of phylo-k-mers, i.e. k-mers that could likely be present in hypothetical sequences, evolutionary close to the reference ones. We cover the inference of such k-mers and k-mer filtering, a method for identifying the most informative ones based on a measure derived from information theory. We discuss a memory-frugal disk-based algorithm for phylo-k-mer inference and filtering, which allows for streaming the most informative k-mers with minimal RAM. This algorithm enables representing large phylogenies with phylo-k-mers, even when these collections do not fit in memory.

Keywords: kmers, filtering, external memory, phylogenetic placement

Pangenomic k-mer distribution with low memory cost

Rouzé Timothé[1], Limasset Antoine[2], and Chikhi Rayan[1]

[1] *G5 Sequence Bioinformatics, Institut Pasteur, France*
[2] *Centre de Recherche en Informatique, Signal et Automatique de Lille - UMR 9189, France*

Abstract

In pangenomic studies, estimating the number of shared k-mers among a large number of input genomes is crucial. Such analysis reveals insights into the species' genome organization and influences how the pangenome is utilized for specific analyses. Although straightforward, this process is highly resource-intensive; for example, managing billions of k-mers in human or large microbial pangenomes can require hundreds of gigabytes of memory. To address this, we introduce a novel structure, the Aggregating Counting Bloom Filters [1], enabling high-precision analysis with significantly reduced memory usage. We implemented the structure in a tool, K-LEB (K-mer Layers Estimation using Bloom filters), developed in Rust. It incorporates scalable sketching techniques to further reduce the computational load, making the analysis adaptable and efficient. We demonstrate its effectiveness with benchmarks on bacterial and human pangenomes. K-LEB is open source and available on GitHub at `www.github.com/TimRouze/KLEB`.

Keywords: k-mers, Bloom, filters, Pangenomics

References

[1] Camille Marchet, Antoine Limasset. "Scalable sequence database search using partitioned aggregated Bloom comb trees", Bioinformatics (39):252–259, 2023, `https://doi.org/10.1093/bioinformatics/btad225`.

A compact embedding-based indexing for accurate and rapid classification in bacterial pangenomics

Jorge Avila Cartes Raghuram Dandinasivara Luca Denti
Simone Ciccolella Gianluca Della Vedova Paola Bonizzoni
Alexander Schönhuth

Species identification is of great importance in areas such as agriculture, food processing, and healthcare. The continually growing genomics databases, especially with the increased focus on bacterial pangenomes in clinical microbiology, have exceeded the capabilities of conventional tools like BLAST. In particular, the need for rapid identification of taxonomy classification from draft assemblies faces two common challenges: species mislabeling and outliers, i.e. cases where the assembly contains sequences that do not fit well within any known taxonomy category.

To address this, we introduce `PANSPACE`, a learning framework designed to compress (draft) assemblies a n-dimensional space, commonly known as embedding. This deep learning framework exploits the frequency matrix of the Chaos Game representation of DNA (FCGR) [3] by defining meaningful architectures that exploits k-mer subsets sharing the suffix. Two ways of creating embeddings are explored, (1) when labels are not considered in the training (autoencoders) [2], and (2) when the labels are considered (metric learning)[5]. Embeddings are indexed, and queries are based on their Euclidean distance. In addition, we use confident learning to identify outliers and possible mislabeled assemblies with the embedding representation in the index, which are validated by the Average Nucleotide Identity distance (ANI). `PANSPACE` distance correlates with ANI.

We evaluate our results on three gold-standard datasets. Compared to the state-of-the-art tools, `PANSPACE` achieved comparable classification results in the mode (1), while in mode (2) it beats its competitors. `PANSPACE` index can achieve a 200× reduction in disk space in the same dataset (661k bacterial assemblies) than [1]. And is 3× faster than the closest tool [4] (with 58k bacterial assemblies in their index).

Availability: `PANSPACE` is available at https://github.com/pg-space/panspace. The index is built on top of FAISS[1], which is an index used for dense vectors to perform similarity search between embeddings.

[1] https://github.com/facebookresearch/FAISS

References

[1] K. Břinda, L. Lima, S. Pignotti, N. Quinones-Olvera, K. Salikhov, R. Chikhi, G. Kucherov, Z. Iqbal, and M. Baym. Efficient and robust search of microbial genomes via phylogenetic compression. *bioRxiv*, pages 2023–04, 2023.

[2] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[3] H. J. Jeffrey. Chaos game representation of gene structure. *Nucleic acids research*, 18(8):2163–2170, 1990.

[4] J. Lumpe, L. Gumbleton, A. Gorzalski, K. Libuit, V. Varghese, T. Lloyd, F. Tadros, T. Arsimendi, E. Wagner, C. Stephens, et al. Gambit (genomic approximation method for bacterial identification and tracking): A methodology to rapidly leverage whole genome sequencing of bacterial isolates for clinical identification. *Plos one*, 18(2):e0277575, 2023.

[5] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Exploiting uniqueness: seed-chain-extend alignment with Elastic Founder Graphs

Rizzo Nicola[1], Càceres Manuel[1], and Mäkinen Veli[1]

[1] *University of Helsinki, Finland*

Abstract

Sequence-to-graph alignment is a central challenge of computational pangenomics: even though aligning sequences and matching patterns in labeled graphs are quadratically hard problems, pangenomic aligners such as GraphAligner (Rautiainen and Marschall, Gen. Biol. 2020), minigraph (Li et al., Gen. Biol. 2020), and others, have been developed based on the heuristic seed-and-extend or seed-chain-extend solution to alignment. Indexable Elastic Founder Graphs (Indexable EFGs) are a specific class of acyclic graphs that break the hardness of pattern matching by exploiting unique substrings (Equi et al., Algorithmica 2023). In this work, we implement a complete alignment pipeline tying together indexable EFGs and seed-chain-extend solutions. To do so, we deal with: efficient indexable EFG construction, also handling ambiguous nucleotides; computing a good subset of Maximal-Exact-Match seeds; chaining these seeds on the Elastic Degenerate String relaxation of the EFG; extending the resulting alignment with the banded alignment of GraphAligner. We perform preliminary experiments at the scale of a human chromosome and discuss the results.

This project received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956229, and from the Academy of Finland grants No. 352821 and 328877.

Keywords: sequence to graph alignment, colinear chaining, elastic founder graphs

The Backpack Quotient Filter: a dynamic and space-efficient data structure for querying k-mers with abundance.

Levallois Victor[1], Andreace Francesco[2], Le Gal Bertrand[1], Dufresne Yoann[2], and Peterlongo Pierre[1]

[1]*Scalable, Optimized and Parallel Algorithms for Genomics, France*
[2]*SeqBio, France*
[3]*Architectures matérielles spécialisées pour l'ère post loi-de-Moore, France*

Abstract

Genomic data sequencing has become indispensable for elucidating the complexities of biological systems. As databases storing genomic information, such as the European Nucleotide Archive, continue to grow exponentially, efficient solutions for data manipulation are imperative. One fundamental operation that remains challenging is querying these databases to determine the presence or absence of specific sequences and their abundances within datasets.

This paper introduces a novel data structure indexing k-mers, the Backpack Quotient Filter (BQF), which serves as an alternative to the Counting Quotient Filter (CQF). The BQF offers enhanced space efficiency compared to the CQF while retaining key properties, including abundance information and dynamicity, with a negligible false positive rate. The approach involves a redefinition of how abundance information is handled within the structure, alongside with an independent strategy for space-efficiency.

We show that the BQF uses 4x less space than the CQF on some of the most complex data to index: metagenomics sequences. We also show that space efficiency increases as the amount of data to be indexed grows, which is in line with the initial scaling up objective.

Keywords: Indexing, kmer, Quotient Filter, Metagenomics

Interpolating and Extrapolating Node Counts in Colored Compacted de Bruijn Graphs for Pangenome Growth Comparison

Luca Parmigiani[1] and Stoye Jens[1]

[1] *Bielefeld University, Germany*

Abstract

A pangenome is a collection of taxonomically related genomes, often from the same species, serving as a representation of their genomic diversity. The study of pangenomes, or pangenomics, aims to quantify and compare this diversity, which has significant relevance in fields such as medicine and biology.

Originally conceptualized as sets of genes, pangenomes are now commonly represented as sequence graphs. These graphs consist of nodes representing genomic sequences and edges connecting consecutive sequences within a genome. Among possible sequence graphs, a common option is the compacted de Bruijn graph. In our work, we focus on the colored compacted de Bruijn graph, where each node is associated with a set of colors that indicate the genomes traversing it.

In response to the evolution of pangenome representation, we introduce a novel method for comparing pangenomes by their growth in terms of node number, targeting two main challenges: the variability in node counts arising from graphs constructed with different numbers of genomes, and the large influence of rare genomic sequences. We propose an approach for interpolating and extrapolating node counts in colored compacted de Bruijn graphs, adjusting for the number of genomes. To tackle the influence of rare genomic sequences, we apply Hill's numbers, a well-established diversity index previously utilized in ecology and metagenomics for similar purposes, to proportionally weight both rare and common nodes according to the frequency of genomes traversing them.

Keywords: Comparative Pangenomics, de Bruijn Graphs

Strangepg: Toward Pangenome Scale Graph Visualization

Bonnet Konstantinn[1,2] and Marschall Tobias[1,2]

[1] *Center for Digital Medicine, Heinrich Heine University, Germany*
[2] *Institute for Medical Biometry and Bioinformatics, Germany*

Abstract

Visualizing large graphs in the million node scale and beyond remains a challenge and is relevant to multiple fields of study. In pangenomics, as databases are continuously enriched by new and high quality assemblies, there is a growing need for tools and methods tailored to handle extremely large volumes of data. As pangenome sizes multiply, available techniques quickly hit operational limits in terms of processing time and memory. In particular, visualizing graphs in real time, intuitively and interactively, is useful for analysis and interpretation, yet computationally prohibitive at such a scale. Indeed, this commonly implies computing layouts, rendering and user interaction on the data as a whole, making even relatively small graphs and annotations in the order of tens of thousands of nodes a difficult challenge. Here we propose strangepg, a novel tool aiming to address these limitations and render visualization feasible on commodity hardware. This is achieved by employing graph coarsening, whereby only a rough representation of the overall graph is shown and processed, and any further detail is unraveled interactively by the user. To maximize efficiency, most of the computational effort is offloaded to a one-time preprocessing step, yielding an indexed graph and a coarsening hierarchy which the visualizer can then trivially query. Finally, the transparent use of external memory to store data exceeding available RAM, ie. using disk space as additional cache, mitigates the high memory requirements. We demonstrate its scalability to pangenome graphs in the hundred million node count and beyond. strangepg is implemented in C in a largely self-contained and highly modular, portable and extensible manner, with the goal of allowing easy substitution of the layouting and coarsening algorithms for ones more suitable to specific applications.

Keywords: pangenome, visualization, variation graph, scalability, external memory, coarsening, layouting

Towards an edit distance between pangenome graphs

Dubois Siegfried[1], Lemaitre Claire[1], Faraut Thomas[2], and Zytnicki Matthias[3]

[1] *Inria Rennes – Bretagne Atlantique, France*
[2] *Génétique Physiologie et Systèmes d'Elevage, France*
[3] *Unité de Mathématiques et Informatique Appliquées de Toulouse, France*

Abstract

A pangenome graph is a sequence graph that aims to represent variations among a collection of genomes in a single data structure. Each genome is segmented and embedded as a path in the graph with its successive nodes corresponding to contiguous segments on the associated genome. Building such graphs relies on alignment heuristics, and thus gives different graphs from the same input data depending on the chosen method, or the set of parameters. In this work, we would like to question to what extent the construction method influences the resulting graph and therefore to what extent the resulting graph reflects genuine genomic variations. We present here an algorithm that analyzes the differences in segmentation across two pangenome graphs, the segmentation being the way the genomes are split into nodes inside the graph structure. We define elementary operations, fusion and fission, that enables to transform one graph into another. Our algorithm provides a dissimilarity measure between each pair of variation graphs: the minimal number of elementary operations. It enables both to quantify the impact of the graph construction method and its parameters and to pinpoint specific areas of the graph and genomes that are impacted by the changes in segmentation. We applied our method on graphs from 21 yeast telomere-to-telomere phased genomes assemblies with the two current state-of-the-art pangenome graph builders, minigraph-cactus and pggb. We show that, with a fixed set of genomes, changing the reference in minigraph-cactus mattered much more than shuffling the order of insertion of the other genomes, and that comparing two minigraph-cactus graphs with different references can result in a higher dissimilarity than comparing a minigraph-cactus graph and the pggb graph.

Keywords: variation graph, pangenome, edit distance

An incremental algorithm for computing the set of all period sets

Rivals Eric[1]

[1] *Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, France*

Abstract

Overlaps between strings are crucial in many areas of computer science, such as bioinformatics, code design, and stringology. A self overlapping string is characterized by its periods and borders. A period of a string u is the starting position of a suffix of u that is also a prefix u, and such a suffix is called a border. Each word of length, say $n > 0$, has a set of periods, but not all combinations of integers are sets of periods. The question we address is how to compute the set, denoted Gamma(n), of all period sets of strings of length n. Computing the period set for all possible words of length n is clearly prohibitive. The cardinality of Gamma(n) is exponential in n. One dynamic programming algorithm exists for enumerating Gamma(n), but it suffers from an expensive space complexity. After recalling some combinatorial properties of period sets, we will present a novel algorithm that computes Gamma(n) from Gamma(n-1), for any length $n > 1$. The period set of a string u is a key information for computing the absence probability of u in random texts. Moreover, computing Gamma(n) is useful for assessing the significance of word statistics, like the number of k-mers shared between two texts, or the number of missing k-mers in one text. Besides applications, investigating Gamma(n) is interesting per se as it unveils combinatorial properties of string overlaps.

Keywords: overlap, periodicity, enumeration, algorithm, combinatorics, string, word, text, statistics

Automated design of efficient search schemes for lossless approximate pattern matching

Renders Luca[1], Depuydt Lore[1], Rahmann Sven[2,3], and Fostier Jan[1]

[1] *Ghent University, Belgium*
[2] *Saarland University, Saarbrücken, Germany*
[3] *Center for Bioinformatics, Germany*

Abstract

We present a novel method for the automated design of search schemes for lossless approximate pattern matching. Search schemes are combinatorial structures that define a series of searches, each of which specifies lower and upper bounds on the number of errors in each part of a partitioned search pattern, and the processing order of these parts. Collectively, these searches guarantee that all approximate occurrences of a search pattern up to a predefined number of k errors are identified. Because generating efficient search schemes is increasingly computationally expensive for larger k, search schemes have been proposed in literature for only up to k=4 errors.

To design search schemes allowing more errors, we combine a greedy algorithm and a new Integer Linear Programming (ILP) formulation. Efficient, ILP-optimal search schemes for up to k=7 errors are proposed and shown to outperform alternative strategies, both in theory and in practice. Additionally, we propose a technique to dynamically select an appropriate search scheme given a specific search pattern. These combined approaches result in reductions of up to 53

We introduce Hato, an open-source software tool (AGPL-3.0 license) to automatically generate search schemes. It implements the greedy algorithm and solves the ILP formulation using CPLEX. Furthermore, we present Columba 1.2, an open-source lossless read mapper (AGPL-3.0 license) implemented in C++. Columba can identify all approximate occurrences of 100 000 Illumina reads (150 bp) in the human reference genome in 24 s (maximum edit distance of 4) and in 75 s (edit distance of 6) using a single CPU core, thereby outperforming existing state-of-the-art tools for lossless approximate matching by a large margin.

Keywords: Search Schemes, Integer Linear Programming, Approximate Pattern Matching, Read Alignment

Mathematical model of phylogenetic compression

Hendrychová Veronika[1] and Břinda Karel[1]

[1] *Inria Rennes, Bretagne Atlantique, France*

Abstract

Comprehensive genome collections play a pivotal role in life sciences research. However, their exponential growth outpaces the development of computational capacities, rendering genome storage and analysis increasingly challenging. For instance, the proportion of data searchable using the Basic Local Alignment Search Tool (BLAST) and its successors has been decreasing exponentially over time. While substantial efforts have recently been devoted to the development of highly optimized alignment-based and k-mer based approaches, these have provided rather partial improvements than a systematic solution of the underlying scalability challenge. Recent work introducing so-called phylogenetic compression has shown that by using evolutionary history to guide existing algorithms and data structures, we can improve state-of-the-art methods for compression and search of large and diverse bacterial genome collections by one to two orders of magnitude [1]. However, despite the clear performance improvement with phylogenetic compression, its theoretical foundations are yet to be established. In this talk, we develop the first formal framework to mathematically study the compression capabilities of phylogenetic compression. To do so, we select one specific protocol of phylogenetic compression and also formalize data compression as a general optimization problem. We demonstrate that, although the compression problem itself might be NP-hard, when input data are modeled by simplified, yet realistic, evolutionary models, phylogenetic compression can provide an optimal solution in polynomial time. Finally, we show that the developed framework accurately models the compression improvement observed in practical applications.

Keywords: phylogenetic compression, compressive genomics, compression

References

[1] K. Břinda et al., "Efficient and Robust Search of Microbial Genomes via Phylogenetic Compression," bioRxiv, Apr. 2023, doi: 10.1101/2023.04.15.536996.

Counting multiple-merger tree structures emerging in Population Genetics

Wirtz Johannes[1]

[1] *Centre d'Ecologie Fonctionnelle et Evolutive UMR, Montpellier, France*

Abstract

Tree structures have always been key tools in both Phylogenetics and Population Genetics to represent evolutionary relations between individuals and species. Depending on the way a tree is combinatorially defined and constructed, the number a(n) of existing tree objects for a given number n of leaves (i.e. individuals/species) may differ. In many cases, to include the most relevant ones in applications, techniques from analytic combinatorics can be employed to determine the asymptotic behaviour of a(n) as n tends to infinity, as well as to estimate certain tree properties such as height or balance. We will discuss several results obtained from this approach on Cayley trees, Steiner trees and the class of "labeled increasing trees", the latter two being of relevance within the context of a range of non-standard Coalescent models.

Keywords: Tree Structures, Enumerative Combinatorics

PlasBin-flow on Pangenome graphs: improving bacterial plasmid binning in short-read assemblies

Sgro Mattia[1], Bonizzoni Paola[1], Chauve Cedric[2], Tomas Vinar[3], and Brejová Brona[3]

[1] *University of Milano-Bicocca, Italy*
[2] *Simon Fraser University, Canada*
[3] *Comenius University in Bratislava, Slovakia*

Abstract

Motivation: Identifying plasmids in sequenced bacterial isolates is a crucial task in microbial genomics helping to monitor spread of antimicrobial resistance. Short-read genome assemblies typically consist of many contigs of variable lengths, which makes it difficult to identify sets of contigs belonging to individual plasmids. We refer to this problem as "plasmid binning". De novo methods for this problem exploit contig features such as length, coverage, circularity, or GC-content, as well as their connections in the assembly graphs. On the other hand, referenced-based strategies make use of homology to databases of known plasmids. Plasbin-flow is a hybrid method that defines plasmid bins as subgraphs of the assembly graph, identified through a MILP model that combines both contig features and plasmid database. Results: PlasBin-flow is sensitive to the quality of the underlying genome assembly graph. In this work, we propose the use of a pangenome graph, built from assembly graphs produced by different software tools from the same sample. This pangenome graph leverages similarities between contigs from different assemblies while also retaining the information on contigs that appear only in one of the input assemblies. Our new tool, Pan-Plasbin-flow, first builds the pangenome graph using "nf-core/pangenome" pipeline, modified to retain information from input assembly graphs. It then uses a modified MILP model from Plabin-flow to identify plasmid bins. Preliminary results on assemblies built by Unicycler and Skesa show increased recall compared to the results based on single assemblies, leading to more true plasmid contigs correctly detected.

Part of this work has been done during a secondment of M. Sgrò, P. Bonizzoni, T. Vinar and B. Brejova at SFU, Vancouver, in collaboration with C. Chauve. This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 872539.

Keywords: Plasmid binning, Antimicrobial Resistance, Pangenome Graphs

Tinted de Bruijn Graphs for efficient read extraction from sequencing datasets

Vandamme Lea[1], Cazaux Bastien[1], and Limasset Antoine[1]

[1] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France*

Abstract

The study of biological sequences often relies on using reference genomes, yet achieving accurate assemblies remains challenging. Consequently, de novo analysis directly from raw reads, without pre-processing, is frequently more practical. We recognize a very commonly shared need across various applications: identifying reads containing a specific k-mer in a dataset. This k-mer-to-reads association would be pivotal in multiple contexts, including genotyping, bacterial strain resolution, profiling, data compression, or error correction. While this challenge appears similar to the extensively researched colored de Bruijn graph problem, resolving it at the read level would be prohibitively resource-intensive in practical applications. In this work, we demonstrate its tractable resolution by leveraging certain assumptions for sequencing dataset indexing. To tackle this challenge, we introduce the Tinted de Bruijn Graph concept, a nuanced version of the colored de Bruijn graph where each read in a sequencing dataset represents a unique source. We developed K2R, a highly scalable index that implement such search efficiently within this framework. K2R's performance, in terms of index size, memory footprint, throughput, and construction time, is benchmarked against leading methods, including hashing techniques (e.g., Short Read Connector) and full-text indexing (e.g., Spumoni and Movi), across various datasets. K2R consistently outperforms contemporary solutions in most metrics and is the only tool capable of scaling to larger datasets in many scenarios. The K2R index, developed in C++, is open source and available on Github.

Keywords: Indexing, Sequencing datasets, Compression

Conway-Bromage-Lyndon (CBL): an exact, dynamic representation of k-mer sets

Martayan Igor[1], Cazaux Bastien[1], Limasset Antoine[1], and Marchet Camille[1]

[1] *Centre de Recherche en Informatique, Signal et Automatique de Lille - UMR 9189, France*

Abstract

We introduce the Conway-Bromage-Lyndon (CBL) structure, a compressed, dynamic and exact method for representing k-mer sets. Originating from Conway and Bromage's concept, CBL innovatively employs the smallest cyclic rotations of k-mers, akin to Lyndon words, to leverage lexicographic redundancies. In order to support dynamic operations and set operations, we propose a dynamic bit vector structure that draws a parallel with Elias-Fano's scheme. This structure is encapsulated in a Rust library, demonstrating a balanced blend of construction efficiency, cache locality, and compression. Our findings suggest that CBL outperforms existing k-mer set methods, particularly in dynamic scenarios. Unique to this work, CBL stands out as the only known exact k-mer structure offering in-place set operations. Its different combined abilities position it as a flexible Swiss knife structure for k-mer set management.

Keywords: kmers, sets, necklaces, data structures

Finimizers: Variable-length bounded-frequency minimizers for k-mer sets

Biagi Elena[1], Alanko Jarno[1], and Puglisi Simon[1]

[1] *University of Helsinki, Finland*

Abstract

The minimizer of a k-mer is the smallest m-mer inside the k-mer according to some order relation $<$ of the m-mers. Minimizers are often used as keys in hash tables in indexing tasks in genomics. The main weakness of minimizer-based indexing is the possibility of very frequently occurring minimzers, which can slow the query times down significantly. Popular minimizer alignment tools employ various and often wild heuristics as workarounds, typically by ignoring frequent minimizers or blacklisting commonly occurring patterns, to the detriment of other metrics (e.g., alignment recall, space usage, or code complexity). In this paper, we introduce frequency-bounded minimizers, which we call finimizers. The idea is to use an order relation $<$ for minimizer comparison that depends on the frequency of the minimizers within the indexed k-mers. With finimizers, the length m of the m-mers is not fixed, but it is allowed to vary depending on the context, so that the length can increase to bring the frequency down below a user-specified threshold t. Setting a maximum frequency solves the issue of very frequent minimizers and gives us a worst-case guarantee for the query time. We show how to implement a family of finimizer schemes efficiently using the Spectral Burrows-Wheeler Transform (SBWT) (Alanko et al., Proc. ACDA, 2023) augmented with longest common suffix information. In experiments, we explore in detail the special case in which we set $t = 1$. This choice simplifies the index structure and makes the scheme completely parameter-free apart from the choice of k. A prototype implementation of Shortest Unique Finimizers exhibits lookup times close to, and often faster than, state-of-the art minimizer-based schemes.

Keywords: Minimizer, Finimizer, Spectral Burrows, Wheeler transform, SBWT, pseudoalignment, k-mer, de Bruijn graph, compact data structures, pangenomics, metagenomics

EpiSegMix: Discovering chromatin states using a flexible distribution hidden Markov model with duration modeling

Schmitz Johanna[1,2], Aggarwal Nihit[3], Laufer Lukas[3], Walter Jörn[3], Salhab Abdulrahman[3,4], and Rahmann Sven[1]

[1] *Algorithmic Bioinformatics, Center for Bioinformatics, Saarland Informatics Campus, Saarland University, Germany*
[2] *Graduate School of Computer Science, Saarland Informatics Campus, Germany*
[3] *Department of Genetics, Saarland University, Germany*
[4] *Integrated Genomics Services, Sidra Medicine, Qatar*

Abstract

Motivation
Histone marks play an essential role in regulating chromatin dynamics and altering DNA accessibility. For instance, H3K27ac (acetylation of lysine (K) at position 27 in the amino–acid sequence of the histone protein H3) is mainly enriched in active promoters and enhancers, while H3K9me3 labels heterochromatic regions, thereby characterizing chromatin states which correspond to genomic elements with different functional roles. The availability of genome-wide histone profiles allows us to automate chromatin state discovery and subsequent segmentation of the genome using a probabilistic model. A popular probabilistic model for chromatin segmentation is the hidden Markov model (HMM), which captures the combinatorial patterns of multiple histone marks using state-specific multivariate emission probabilities and the spatial relations via transition probabilities. However, existing methods have two limitations. First, they use a fixed probability distribution type to model the read counts of all histone marks, although the read count distributions show different levels of overdispersion and skewness. Second, they have a limited flexibility to model chromatin domains of varying lengths, e.g., short promoters in comparison to long genes.
Method
To address the above limitations, we developed a new chromatin segmentation tool based on a more flexible HMM. We support several distribution types to fit the read count distributions, including the commonly used 2-parametric Negative Binomial distribution and the more flexible 3-parametric Beta Negative Binomial distribution. Due to our flexible framework, the user may specify a different distribution type for each mark. In addition, we change the internal HMM topology to an extended-state HMM, which enables a more flexible state duration modeling.
Results
We shows that the increased flexibility of EpiSegMix allows us to more accurately discover chromatin states that are predictive of cell biology compared to existing methods.

Keywords: HMM, Hidden Markov Model, chromatin segmentation, histone modification, probabilistic model

Generalized uncertainty-aware haplotype quantification with application in HLA typing and virus analysis.

Uzuner Hamdiye[1], Köster Johannes[1], Schadendorf Dirk[1], and Paschen Annette[1]

[1] *University of Duisburg-Essen, Essen, Germany*

Abstract

We present Orthanq (ORThogonal evidence HAplotype Quantification), an uncertainty-aware framework for haplotype quantification at subclonal resolution. Our approach for haplotype quantification is based on a Bayesian latent variable model that uses variant allele frequencies (VAFs) in sequencing data. It firstly starts with finding the most plausible set of haplotypes using maximum likelihood estimates of VAFs via a linear program. Secondly, a Bayesian model is employed. The model uses the posterior allele frequency distribution of individual variants as provided by Varlociraptor. Using candidate haplotypes found in the first step, all possible explanations of observed VAFs are explored via combinations of latent haplotype fractions. The exploration involves a recursion with exponential runtime complexity. Therefore, in practice, we prune the search space using representatives from haplotype equivalence classes, i.e. haplotypes that have similar set of variants. Valid solutions may only contain up to one haplotype from each equivalence class. Orthanq currently makes use of SNVs and small indels, and will be extended towards multiple nucleotide variant support. The proposed model relies on pangenome read alignments to capture the most relevant genetic variation. Orthanq can be applied for HLA typing and virus lineage quantification. Using benchmark datasets, we show that Orthanq performs same or better prediction than state-of-the-art HLA typers. We also evaluate Orthanq on virus lineage quantification by creating simulated samples that contain SARS-CoV-2 lineages. Orthanq allows to track its decisions down to individual variants that can be explored via comprehensive visualizations. The model will also help solving problems associated with heterogeneity of tumors when determining sensitivity of subclones to neoantigen based immunotherapy. Orthanq can be reached under https://orthanq.github.io.

Keywords: Haplotype quantification, HLA typing, Bayesian latent variable model