Abstracts and proceedings of

SeqBIM

2023 - 20-21 nov.
Lille

November 20-21 2023

Lille

cnrs GDR Groupement de recherche
BIM Bioinformatique moléculaire

cnrs GDR Groupement de recherche
IM Informatique Mathématique

Keynote speaker

Birte Kehr

(Algorithmic Bioinformatics, Leibniz Institute for Immunotherapy, Leibniz, Germany)

Genotyping structural variation : from simple to complex

Abstract

Structural variants (SVs) contribute significantly to human genetic diversity, affect the risk of many complex diseases, and can be the cause of rare inherited diseases. By definition, SVs affect at least 50 base pairs of sequence and comprise many types of variants ranging from simple deletions to complex rearrangements that involve multiple DNA segments. Estimates of the number of SVs were recently corrected from thousands to tens of thousands per human genome by using long read data, which demonstrates that the majority of SVs have remained undetected in previous studies on short read data. Given that short read data is still cheaper and much more abundant than long read data, the aim of my research group is to make use of these available masses of data. In my talk I will introduce our approaches for calling SVs using data from short read sequencing of human genomes. In the first part, I will present our approach PopDel for detecting simple deletions in short read data of very many genomes simultaneously and briefly outline our extension of PopDel to simple duplications and inversion. In the second part, I will introduce our unpublished work on genotyping known complex structural variants including corresponding measures of genotype uncertainty.

Keynote speaker

Christopher Quince

(High-Resolution Microbiomics, Earlham Institute, Norwich, United-Kingdom)

Resolving genomes from metagenomes

Vizitig: de Bruijn graph visualization tool

Charles Paperman, Camille Marchet
Univ. Lille

The de Bruijn graph is a fundamental data structure in computational biology, widely used for tasks such as genome assembly, variant calling, and k-mer set representation. Several fast construction methods for de Bruijn graphs (such as BCALM2 [CLM16], Bifrost [HM20] and recent GGCAT [CT23]) have popularized their usage in recent years. Bandage [WSZH15] is the reference software for visualizing de Bruijn graphs in bioinformatics using a graphical user interface (GUI), but is primarily designed to explore genome assemblies. Prior to Bandage, other works relied on tools like Cytoscape or ad-hoc solutions [JLT+18], but none integrated specificities of de Bruijn graphs, such as the possibility to compact the graph into a unitig graph for a more compact representation, or to have a semantic on the nodes' colors (in so-called colored de Bruijn graphs).

This work introduces Vizitig, a tool for visualizing and manipulating de Bruijn graphs. Vizitig is implemented using NetworkX and NetworkDisk, making it compatible with Python ecosystems. This solution also avoids a resource intensive manipulation of large graphs. Users can explore the graph, search for sequences, and extract relevant information, all while benefiting from indexed and thread-safe graph access. We will demonstrate that Vizitig supports both traditional de Bruijn graphs and is actively developing support for colored de Bruijn graphs.

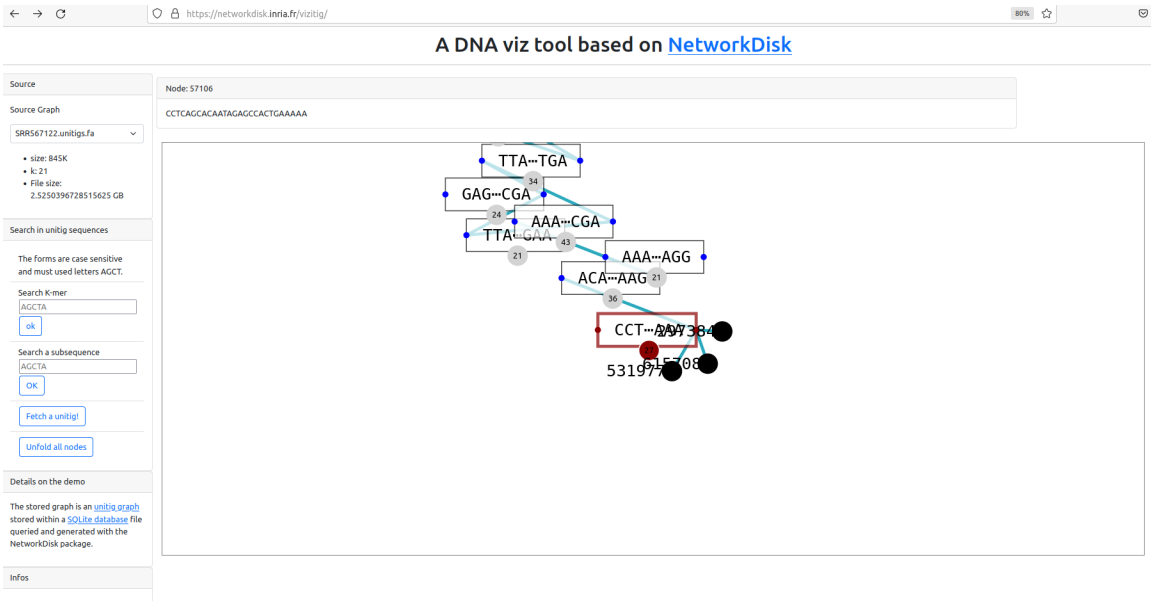

Figure 1: Local nodes of a subgraph in Vizitig in web browser mode.

Vizitig is currently available at https://networkdisk.inria.fr/vizitig/ and open source https://gitlab.inria.fr/pydisk/examples/vizitig.

References

[CLM16] Rayan Chikhi, Antoine Limasset, and Paul Medvedev. Compacting de bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics*, 32(12):i201–i208, 2016.

[CT23] Andrea Cracco and Alexandru I Tomescu. Extremely fast construction and querying of compacted and colored de bruijn graphs with ggcat. *Genome Research*, pages gr–277615, 2023.

[HM20] Guillaume Holley and Páll Melsted. Bifrost: highly parallel construction and indexing of colored and compacted de bruijn graphs. *Genome biology*, 21(1):1–20, 2020.

[JLT+18] Magali Jaillard, Leandro Lima, Maud Tournoud, Pierre Mahé, Alex Van Belkum, Vincent Lacroix, and Laurent Jacob. A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. *PLoS genetics*, 14(11):e1007758, 2018.

[WSZH15] Ryan R Wick, Mark B Schultz, Justin Zobel, and Kathryn E Holt. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, 31(20):3350–3352, 2015.

Extended abstract

SeqBIM

Towards solving the peptidomics problem with ProSpect

Emile Benoist[1]*, Géraldine Jean[1], Hélène Rogniaux[2], Guillaume Fertin[1], Dominique Tessier[2]

[1] *Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France*
[2] *INRAE, PROBE research infrastructure, BIBS facility, F-44300 Nantes, France & INRAE, UR1268 Biopolymères Interactions Assemblages, F-44316 Nantes, France*
***Corresponding author**: emile.benoist@univ-nantes.fr

Abstract

Peptidomics refers to the branch of proteomics dedicated to identifying protein fragments, called *peptides*, that are naturally present in a living system. Identifying such peptides has multiple and important applications, notably in health and agronomy. A major challenge in this area consists in providing a fast and accurate computational method capable of identifying such peptides, even if they carry modifications. Such a method conceptually differs from classical peptide identification methods in proteomics, in which peptides are known to be tryptic because enzymatically digested before entering the mass spectrometer. In peptidomics, we cannot rely on such an assumption, which substantially increases the search space, and thus makes the problem considerably more difficult. Our work consists of the implementation of a new software, ProSpect, aimed at identifying peptides in the context of peptidomics. ProSpect is able to compare (i) a set of spectra (obtained by the analysis of a peptides sample by mass spectrometry) against (ii) an entire protein database (i.e., not previously *in silico* digested). Its core algorithm is somewhat similar to the one developed in SpecGlobX, a software whose objective is to interpret given PSMs (Peptide-Spectrum-Matches) in classical proteomics. However, it is adaptated from the proteomics to the peptidomics context, and most importantly, ProSpect has been fine-tuned and optimized in many ways, so as to be able to run in reasonable time while provinding accurate results.

Keywords

Peptidomics — Mass Spectrometry — Peptide Identification — Running time — Optimization — Dynamic programming

1. Introduction

Peptidomics refers to the research area aimed at characterizing and identifying peptides (i.e., small protein fragments) as well as their post-translational modifications (PTMs), in a given organism or a given tissue. Identifying such peptides is of great importance towards better understanding living organisms, and has many applications, notably in health, where peptides can act as pathology biomarkers in body fluids such as urine or blood [1], or in agronomy, where peptides are both known to be pathogen or pathogen-resistant in plants [2].

A widely used technique, when it comes to identifying peptides or proteins, is

mass spectrometry (see e.g. [3]). A mass spectrometer is used to measure the *mass-to-charge* ratio of molecules (in our case, peptides) resulting in a set of MS1 spectra. Note that the mass-to-charge ratio can be converted into a mass knowing the charge; thus, for simplicity we will thereafter speak of *mass* instead of mass-to-charge ratio.

Because the mass of a peptide is not enough to identify it, peptides are fragmented, and the masses of the resulting fragments are then measured so as to obtain a set of MS2 spectra. In summary, each MS2 spectrum carries information, in the form of peaks, about the amino acid sequence of a given peptide. The objective is then, starting from a biological sample, to provide one (or several) relevant *Peptide-Spectrum Match(es)* (PSM(s)) for as many MS2 spectra as possible. For simplicity, in the following, the term *spectra* will be used instead of MS2 spectra. To identify the sequence corresponding to a spectrum, a protein database is generally used, based on the assumption that the peptide we are looking for comes from a protein of the database (possibly carrying modifications such as PTMs). When the objective is to identify *proteins* in a sample, these proteins are first enzymatically digested into peptides, using a specific enzyme, before peptides enter the mass spectrometer. Trypsin is the most commonly used enzyme, and has the effect of cutting proteins after each K (lysine) and R (arginine) amino acid, resulting in *tryptic* peptides. This allows to directly compare spectra with tryptic peptide sequences obtained by *in silico* digestion of the proteins from the database.

However, the above description of peptide identification does not work *per se* for peptidomics. The obvious reason is that the sought peptides, which enter the mass spectrometer, are no longer tryptic. Otherwise stated, a spectrum can correspond to *any* sequence, at *any* location along the proteins of the database. This makes the problem much more complex, and calls for new tools, as existing ones cannot be used as such in peptidomics. Indeed, an intuitive but inapplicable method would be to replace the above described *in silico* digestion of the database by generating all (i.e. non necessarily tryptic) possible peptides. However, this would considerably increase the number of candidate peptides and lead to unacceptable running times.

Hence, in order to identify peptides from a biological sample in the peptidomics context, we implemented ProSpect (PROteome-Scale Search for SPECTra). The idea behind ProSpect is to directly compare each spectrum s with all protein sequences of the database, so as to find at which position(s) s best matches these sequences. To do that, we started from the core algorithm of SpecGlobX, a software we previously designed [4] (see also `https://github.com/bibs-lab/SpecGlobX`). SpecGlobX being dedicated to classical proteomics, we had to adapt it to the present context. We must indeed take into account that, in ProSpect, the spectra are compared to protein sequences instead of (tryptic) peptide sequences, and that the objective is not to interpret how a *given sequence* matches a given spectrum s, but rather to *find* at which location(s) on the protein sequences s best matches. The second major difference between ProSpect and SpecGlobX is that SpecGlobX takes as input PSMs that have *already been computed* by a peptide identification method (chosen by the user), whereas ProSpect has to determine PSMs by itself. This implies that ProSpect has to deal with a much larger search space than SpecGlobX. The following numerical illustration shows that ProSpect requires refined algorithmic design efforts and technical optimizations compared to SpecGlobX: in

practice, SpecGlobX is able to interpret one million PSMs in approximately 10 minutes with one thread on a standard computer. Assuming SpecGlobX would have to determine PSMs by itself *only considering tryptic peptides*, its execution time would then exceed 6,000 hours (based on a sample of 47,000 spectra to match against the human proteome, which contains 20,000 proteins and on average 40 tryptic peptides per protein).

2. Method

We first give insight on how the main algorithm within SpecGlobX works. Due to space constraints, it will not be fully described: some features will be omitted, and we refer the interested reader to [4] for a full and detailed description of SpecGlobX.

2.1 SpecGlobX in a Nutshell

SpecGlobX is a standalone Java application which, given a list of PSMs (where a PSM is a pair consisting of a spectrum s and its best putative peptide π), provides for each PSM an interpretation and a score. Here, an interpretation corresponds to the best way to match s to π, allowing for mass and/or sequence changes (notably, PTMs); while the score provided by SpecGlobX reflects the quality of the interpretation (a higher score means a better interpretation). For this, SpecGlobX relies on a dynamic programming algorithm using a 2-dimensional matrix M, whose width and height respectively correspond to the number of peaks in s and to the sequence length of π plus one. A cell $M[i][j]$ of M corresponds to the $(i-1)^{th}$ amino acid of π (which we will denote aa_{i-1}) and the j^{th} peak of s (which we will denote p_j). M is then filled from left to right and from top to bottom, and each cell value $M[i][j]$ is computed so as to contain the best interpretation score between the first j peaks of s and the first $(i-1)$ amino acids from π. The scores of the first row are initialized to 0 and the scores of the first column cells $M[i][0]$ are initialized to $-4i$ in order to penalize an interpretation that does not start at the beginning of π. Note that, excepted for the first row and first column, each cell score $M[i][j]$ is computed, based on the score of a cell c located to its left and in the previous row. Once M is completely filled, the best interpretation for PSM (s, π) is found by looking at the cell c^* belonging to the last row and that contains the highest score. Once c^* is identified, the corresponding interpretation is provided by tracing back, one by one, all cells that led to c^*.

2.2 ProSpect: Main Features

Moving from the proteomics context towards the peptidomics context of peptidomics implied adaptating SpecGlobX and improving it, so as to obtain and fast and accurate tool, namely ProSpect. We start by presenting the two main adaptations we applied to SpecGlobX.

First, in matrix M, not only the scores of the last row are taken into account, but *any* score at *any* position can be considered as a good interpretation for a spectrum s, as long as the corresponding score is high enough. More precisely, for each spectrum, ProSpect will output b best-scored interpretations corresponding to PSMs, where b is a user-defined parameter. To do that, we implemented a dynamic binary heap H of size b, which is updated each time a newly computed score is strictly better than the worst score in H. Each update is then achieved in a time which is logarithmic

in b. The second adaptation comes from the following observation: in SpecGlobX, we only consider the last row in M, because the goal is to fully match peptide π to spectrum s. In ProSpect, however, we are looking for peptides that can correspond to s *at any position along the proteins*. Hence, instead of initializing the first column elements $M[i][0]$ with score $-4i$ as we did in SpecGlobX, we simply initialize the first column of M with 0s, as all starting points can be relevant and thus, no penalty should be applied.

We now present some of the fine-tuned technical improvements we applied to ProSpect so as to achieve reasonable execution time. Due to space constraints, only two of them will be briefly described.

Recall that in SpecGlobX, the score of each cell $M[i][j]$ of M is computed based on the score of a previous cell in M. There are several possible scenarios, and one of them consists of applying a penalty of 4 on the score of the cell above it ($M[i][j] = M[i-1][j] - 4$). This scenario is extremely frequent, as it happens when the amino of row i (aa_{i-1}) is not found in s at the position of the peak corresponding to column j (p_j). We call such a scenario a "not found". In ProSpect, only the score of the cells that *do not* correspond to a "not found" scenario are computed. This allows us to spare many uninformative computations.

The second improvement concerns preprocessing of the spectra. When a spectrum s is preprocessed, it is possible to detect that some of its peaks are useless, as they do not support proof of the presence of an amino acid in s. Such a peak corresponds to a column in matrix M that only contains "not found" scenarios, and would thus only contain very low scores. Hence, during this preprocessing step, these columns are removed from M. In practice, on average, this divides the number of columns in M by 3, which leads to a further acceleration of the running time.

3. Experimental Results

In order to assess scalability and accuracy of ProSpect, we ran two types of experiments. We executed these experiments on a laptop equipped with an Intel processor (2.6 GHz) and 8GB of RAM dedicated to the Java Virtual Machine, running under Windows 10 in a multithreaded environment (11 threads).

The first test we performed consisted in running ProSpect so as to evaluate its running time. For this, we used a standard dataset generated from HEK293 cells [5] and downloaded from PRIDE (PXD001468), which approximately contains 47,000 spectra. The protein database we used corresponds to the human proteome and consists of about 20,000 proteins (UniProt UP000005640). The running time needed by ProSpect to perform an all-to-all comparison on this dataset is roughly 9h. As a comparison, recall that if the fine-tuned improvements presented in the previous section had not been performed, then the execution time on the same dataset, but *only considering tryptic peptides*, would have exceeded 6,000h.

The second dataset we used is to assess accuracy of our method. It is composed of a small set of 694 spectra coming from the Cytochrome c bovine purified protein (CYC_BOVIN), most of them corresponding to tryptic peptides. The protein database contains the reference proteome of a remote species (here, *E. coli*), together with the CYC_BOVIN protein (4,404 proteins altogether). We compared ProSpect to

two other tools, namely SpecOMS coupled with SpecGlobX, as well as MS-GF+. SpecOMS and MS-GF+ are both peptide identification tools (see resp. [6] and [7] for a detailed description). Since SpecOMS returns PSMs, we used its output as input for SpecGlobX. Table 1 gives the number of correct PSMs provided by each of these three methods, where by "correct" here we mean that the PSM corresponds to a peptide coming from the CYC_BOVIN protein, and not from *E. coli*. As can be seen, even without *a priori* (i.e. without knowing that most peptides are tryptic), ProSpect outperforms the two other methods, and notably provides 20% more correct PSMs than its best competitor.

Table 1. Number of correct PSMs provided by each of the three methods in our second experiment

SpecOMS/SpecGlobX	MS-GF+	ProSpect
228	192	**273**

Acknowledgments

This work was supported by ANR DeepProt (ANR-18-CE45-044).

References

[1] Letícia de Almeida Brondani, Ariana Aguiar Soares, Mariana Recamonde-Mendoza, Angélica Dall'Agnol, Joíza Lins Camargo, Karina Mariante Monteiro, and Sandra Pinho Silveiro. Urinary peptidomics and bioinformatics for the detection of diabetic kidney disease. *Scientific Reports*, 10:1242, 2020.

[2] Koji Yamaguchi and Tsutomu Kawasaki. Pathogen- and plant-derived peptides trigger plant immunity. *Peptides*, 144:170611, 2021.

[3] E. Ingvar, F. Kristian, M. Lennart, and M. Svein-Ole. *Computational Methods for Mass Spectrometry Proteomics*. Wiley, 2008.

[4] Grégoire Prunier, Mehdi Cherkaoui, Albane Lysiak, Olivier Langella, Mélisande Blein-Nicolas, Virginie Lollier, Emile Benoist, Géraldine Jean, Guillaume Fertin, Hélène Rogniaux, and Dominique Tessier. Fast alignment of mass spectra in large proteomics datasets, capturing dissimilarities arising from multiple complex modifications of peptides. *bioRxiv*, 2023.

[5] Joel M. Chick, Deepak Kolippakkam, David P. Nusinow, Bo Zhai, Ramin Rad, Edward L. Huttlin, and Steven P. Gygi. A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides. *Nature Biotechnology*, 33:743–749, 2015.

[6] Matthieu David, Guillaume Fertin, Hélène Rogniaux, and Dominique Tessier. SpecOMS: A full open modification search method performing all-to-all spectra comparisons within minutes. *Journal of Proteome Research*, 16(8):3030–3038, 2017.

[7] Sangtae Kim and Pavel A. Pevzner. MS-GF+ makes progress towards a universal database search tool for proteomics. *Nature Communications*, 5:5277, 2014.

Abstract

SeqBIM

Towards an edit distance between pangenome graphs

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

[1] *Univ Rennes, Inria, CNRS, IRISA, Rennes, F-35000, France*
[2] *MIAT, Université de Toulouse, INRAE, 31320 Castanet-Tolosan, France*
[3] *GenPhySE, Université de Toulouse, INRAE, ENVT, 31320 Castanet-Tolosan, France*
***Corresponding author**: siegfried.dubois@inria.fr

Abstract

A variation graph is a data structure that aims to represent variations among a collection of genomes. It is a sequence graph where each genome is embedded as a path in the graph with the successive nodes, along the path, corresponding to successive segments on the associated genome sequence. Shared subpaths correspond to shared genomic regions between the genomes and divergent path to variations: this structure features inversions, insertions, deletions and substitutions. The construction of a variation graph from a collection of chromosome-size genome sequences is a difficult task that is generally adressed using a number of heuristics such as those implemented in the *state-of-the-art* pangenome graph builders *minigraph-cactus* [1] and *pggb* [2]. The question that arises is to what extent the construction method influences the resulting graph and therefore to what extent the resulting graph reflects genuine genomic variations.

We propose to address this question by constructing an edition script between two variation graphs built from the same set of genomes which provides a measure of similarity, and more importantly that enables to identify discordant regions between the two graphs. We proceed by comparing, for each genome, the two corresponding paths in the two graphs which correspond to two possibly different segmentations of the same genomic sequence. As such, for each interval defined by the nodes of the path of the genome in the first graph, we define a set of relations with the nodes of the second graph, such as equalities, prefix and suffix overlaps... which allows for a calculation of how many elementary operations, such as fusions and divisions of nodes, are required to go from one graph to another.

We tested our method on variation graphs constructed using both simulated dataset as well as a real dataset made of 15 yeast telomere-to-telomere phased genome assemblies [3], with *minigraph-cactus* as the graph construction tool. This tool builds iteratively the variation graph, starting from a genome taken as a reference and incorporating each genome in the order provided by the user. In this work, we compared by pairs the graphs constructed from the same set of genomes but using different incorporation orders. After the application of our algorithm, we get a measure of the similarity between each pair of variation graphs in the form of a

distance, that enables both to quantify the impact of the order of genomes in the graph construction, and to pinpoint the specific areas of the graph and genomes that are impacted by the changes in segmentation. Ongoing work includes being able to compare graphs issued from different variation graphs construction tools.

Availability: This algorithm is implemented as a Python tool: https://github.com/Tharos-ux/pancat

References

[1] Glenn Hickey, Jean Monlong, Jana Ebler, Adam Novak, Jordan M. Eizenga, Yan Gao, Human Pangenome Reference Consortium, Tobias Marschall, Heng Li, and Benedict Paten. Pangenome Graph Construction from Genome Alignment with Minigraph-Cactus, April 2023.

[2] Erik Garrison and Andrea Guarracino. Unbiased pangenome graphs. *Bioinformatics*, 39(1):btac743, January 2023.

[3] Telomere-to-telomere assemblies of 142 strains characterize the genome structural landscape in Saccharomyces cerevisiae | Nature Genetics.

Abstract

SeqBIM

Réduction de l'espace utilisé par le Counting Quotient Filter

Victor Levallois[1]*, Yoann Dufresne[2], Pierre Peterlongo[1]

[1] *équipe Genscale, Inria, Rennes, France*
[2] *équipe Seqbio, Institut Pasteur, Paris, France*
***Corresponding author**: victor.levallois@inria.fr*

Abstract

Motivation

Depuis qu'il est possible de séquencer l'ADN, les bases de données publiques ne cessent de croître exponentiellement (33Po de données normalisées pour SRA en 2023). Dans l'objectif de pouvoir interroger ces très gros ensembles de données, plusieurs structures de données approximées sont utilisées. Parmi elles, les filtres de bloom [1], sont massivement utilisés, ainsi que les Counting Bloom Filters (CBF) [2] qui indexent conjointement des k-mers (mots de taille k) et leurs abondances. Depuis 2017, les Counting Quotient Filter (CQF) [3] ont été introduits pour concurrencer directement les CBF. Entre autres avantages, le CQF utilise moins d'espace que les CBF lorsque le taux de faux positifs est sous 1.56%.

Contribution

Dans ce travail, nous présentons une implémentation d'une nouvelle variante de CQF appelée Backpack Quotient Filter (BQF). Le BQF change la façon de stocker l'abondance des k-mers. L'idée est basée sur le schéma Fimpera [4] qui indexe un k-mer en stockant la liste des s-mers plus petits qui le composent. Le stockage de ces éléments plus petits autorise l'utilisation de fonctions de hachages nécessitant moins de bits, libérant ainsi plusieurs bits par élément, exploités pour stocker l'abondance. Contrairement à la structure originale, le BQF stocke ainsi l'information d'abondance, sur un unique slot par élément. Ainsi le BQF stocke la même information que le CQF sur moins d'espace. Ceci se fait au prix de la création de faux positifs dits de *construction*, inférieurs à 10^{-11}% dans nos expérimentations.

References

[1] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. 13(7):422–426.

[2] Li Fan, Pei Cao, J. Almeida, and A.Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. 8(3):281–293. Conference Name: IEEE/ACM Transactions on Networking.

[3] Prashant Pandey, Michael A. Bender, Rob Johnson, and Rob Patro. A general-purpose counting filter: Making every bit count. In *Proceedings of the 2017 ACM*

International Conference on Management of Data, SIGMOD '17, pages 775–787. Association for Computing Machinery.

[4] Lucas Robidou and Pierre Peterlongo. fimpera: drastic improvement of approximate membership query data-structures with counts. *Bioinformatics*, 39(5):btad305, 2023.

Abstract

Mapping-friendly Sequence Reductions to process compressed genomic data

Roland Faure[1,2]*, Baptiste Hilaire[1,] Dominique Lavenier[1]

[1]*Univ. Rennes, Inria RBA, CNRS UMR 6074, Rennes, France*
[2]*Evolution Biologique et Ecologie, Université libre de Bruxelles (ULB), Brussels, Belgium*
***Corresponding author**: roland.faure@irisa.fr*

Abstract

Efficiently managing vast DNA datasets necessitates the development of highly effective sequence compression techniques to reduce storage and computational requirements. We propose here to explore the potential of a lossy compression technique, Mapping-friendly Sequence Reductions (MSRs).

MSRs were introduced in [1] as a generalization of homopolymer compression to improve the accuracy of alignment tools. Essentially, MSRs deterministically transform sequences into shorter counterparts, in such a way that if an original query and a target sequence align, their reduced forms will align as well. While homopolymer compression is one example of an MSR, numerous others exist, potentially offering substantial sequence length reduction—such as retaining only bases between 'A' and 'T' (on average, 1 base out of 16): AACAGTGACACTAAACT → GCC. These rapid computations yield lossy representations of the originals. Notably, the reduced sequences can be stored, aligned, assembled, and indexed much like regular sequences.

MSRs could be used to improve the efficiency of taxonomic classification tools, by indexing and querying reduced sequences. Our experimentation with a toy example, a mixture of 10 *E. coli* strains, demonstrates that this approach can yield greater precision than indexing and querying a reduced portion of k-mers (typically minimizers). Specifically, using the reduction described above, 76% of reduced 31-mers were unique, whereas only 47% of not-reduced 31-mers were.

In our presentation, we will also explore other tasks that could benefit from sequence reduction, such as mapping, genome assembly, and structural variant detection.

References

[1] Blassel L, Medvedev P, Chikhi R. Mapping-friendly sequence reductions: Going beyond homopolymer compression. iScience. 2022 Oct 13;25(11):105305.

DeTox: A pipeline for the Detection of Toxins in venomous organisms

Ringeval Allan[1]*, Farhat Sarah[1], Fedosov Alexander[1,2], Gerdol Marco[3,4], Greco Samuele[3], Mary Lou[1], Modica Maria Vittoria[4], Puillandre Nicolas[1].

[1]*Institut Systématique Evolution Biodiversité (ISYEB), Muséum National d'Histoire Naturelle, CNRS, Sorbonne Université, EPHE, Université des Antilles, Paris, France.*

[2]*Department of Zoology, Swedish Museum of Natural History, Stockholm, Sweden.*

[3]*Department of Life Sciences, University of Trieste, Trieste, Italy*

[4]*Department of Biology and Evolution of Marine Organisms (BEOM), Stazione Zoologica Anton Dohrn, Roma, Italy*

*__Corresponding author__: allan.ringeval@mnhn.fr

Abstract

Venomous organisms have independently evolved the ability to produce toxins 101 times during their evolutionary history, resulting in over 200,000 venomous species. Collectively, these species produce millions of toxins, making them a valuable resource for bioprospecting and understanding the evolutionary mechanisms underlying genetic diversification. RNA-seq is the preferred method for characterizing toxin repertoires, but the analysis of the resulting data remains challenging. While early approaches relied on similarity-based mapping to known toxin databases, recent studies have highlighted the importance of structural features for toxin detection. The few existing pipelines lack an integration between these complementary approaches, and tend to be difficult for non-experienced users.

To address these issues, we developed DeTox, a user-friendly, all-in-one tool for toxin research. It combines rapidity of execution, parallelization, and customization of parameters. DeTox was tested on published transcriptomes from gastropod mollusks, cnidarians, and snakes, retrieving most putative toxins from the original articles and identifying additional peptides as potential toxins to be confirmed through manual annotation and eventually proteomic analysis. By integrating a structure-based search with similarity-based approaches, DeTox allows the comprehensive characterization of toxin repertoire in poorly-known taxa. The effect of the taxonomic bias in existing databases is minimized, as mirrored in the detection of unique and divergent toxins that would have been overlooked by similarity-based methods. DeTox streamlines toxin identification, providing a valuable tool for efficient identification of venom components that could enhance venom research in neglected taxa.

Abstract

SeqBIM

Developing phylogeny-colored de Bruijn graphs for bacterial gene birth detection

Arya Kaul[1,2]*, Karel Břinda[2], Michael Baym[1]

[1] *Department of Biomedical Informatics, Harvard Medical School, United States of America*
[2] *GenScale, Inria/IRISA Université de Rennes, France*
***Corresponding author**: arya_kaul@g.harvard.edu

Abstract

Understanding the mechanisms of gene birth is key for explaining the extremely diverse gene content in bacteria.[1]. However, despite its importance, the identification of specific gene birth events is challenging due to the large genomic variation across bacterial species and their rapid evolution. Recent works have proposed two general mechanisms by which a bacterial species may acquire novel genes: horizontal gene transfer and structural variation.[2] However, while gene birth via horizontal gene transfer has been extensively studied, studies of the contributions of structural variation are hindered by the difficulty in distinguishing it from the evolution of pre-existing or horizontally acquired genes.[3, 4, 5]

In this talk, we will describe our ongoing work on the role of structural variation in bacterial gene birth. First, we will present our central hypothesis that large deletions may lead to novel fusion genes, and that this may represent a major mechanism of gene birth in bacteria. Then, we will introduce two alignment-based approaches to identify fusion gene birth in long-term evolutionary experiments (i.e., within a single species) and across species that are highly related. Finally, we will present a novel variant of colored de Bruijn graphs (dBG), called phylogeny-colored dBG, to identify gene birth events within large genome repositories and at various time scales.

References

[1] James O McInerney. Prokaryotic pangenomes act as evolving ecosystems. *Molecular Biology and Evolution*, 40(1), October 2022.

[2] Itamar Sela, Yuri I. Wolf, and Eugene V. Koonin. Assessment of assumptions underlying models of prokaryotic pangenome evolution. *BMC Biology*, 19(1), February 2021.

[3] Todd J. Treangen and Eduardo P. C. Rocha. Horizontal transfer, not duplication, drives the expansion of protein families in prokaryotes. *PLoS Genetics*, 7(1):e1001284, January 2011.

[4] Evgeni Bolotin and Ruth Hershberg. Horizontally acquired genes are often shared between closely related bacterial species. *Frontiers in Microbiology*, 8, August 2017.

[5] Caroline M. Weisman, Andrew W. Murray, and Sean R. Eddy. Many, but not all, lineage-specific genes can be explained by homology detection failure. *PLOS Biology*, 18(11):e3000862, November 2020.

Extended abstract

SeqBIM

Approximate Cartesian Tree Matching: An approach Using Swaps *

Bastien Auvray[1]*, Julien David[1, 2], Richard Groult[1, 3], Thierry Lecroq[1, 3]

[1] *CNRS NormaSTIC FR 3638, Caen, Le Havre, Rouen, France*
[2] *Normandie University, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France*
[3] *Univ Rouen Normandie, INSA Rouen Normandie, Université Le Havre Normandie, Normandie Univ, LITIS UR 4108, F-76000 Rouen, France*
***Corresponding author**: bastien.auvray@etu.univ-rouen.fr

Abstract

Cartesian tree pattern matching consists of finding all the factors of a text that have the same Cartesian tree than a given pattern. There already exist theoretical and practical solutions for the exact case. To the best of our knowledge, no result is known for approximate pattern matching with Cartesian trees. We consider Cartesian tree pattern matching with one swap: given a pattern and a text we propose two algorithms that find all the factors of the text that have the same Cartesian tree of the pattern after one transposition of two adjacent symbols.

Keywords

Cartesian tree — Approximate pattern matching — Swap

1. Introduction

In general terms, the pattern matching problem consists of finding one or all the occurrences of a pattern in a text. When both the pattern and the text are strings, the problem has been extensively studied and has received a huge number of solutions [2]. Searching time series or list of values for patterns representing specific fluctuations of the values requires a redefinition of the notion of pattern. The question is to deal with the recognition of peaks, breakdowns, or more features. For those specific needs, one can use the notion of Cartesian tree.

Cartesian trees have been introduced by Vuillemin in 1980 [3]. Recently, Park *et al.* [4] introduced a new metric of generalized matching, called Cartesian tree matching. It is the problem of finding every factor of a text which has the same Cartesian tree as that of a given pattern. Cartesian tree matching can be applied, for instance, to finding patterns in time series such as share prices in stock markets or gene sample time data.

However, in real life applications data are often noisy, and it is thus important to find factors of the text that are similar, to some extent, to the pattern. We showcase the first results in this setting by considering approximate Cartesian tree pattern matching with one transposition (aka swap) of one symbol with the adjacent symbol. Swaps are common in real life data, and it seems natural to consider them in the

*Part of this work has been presented in [1]

Cartesian pattern matching framework.

2. Basics

We consider sequences of integers with a total order $<$. We also assume all elements to be distinct.

2.1 Cartesian tree

Given a sequence x of length n, the Cartesian tree $C(x)$ associated to that sequence is recursively defined as follows:

- if x is empty, then $C(x)$ is the empty tree;
- if $x[1 \dots n]$ is not empty and $x[i]$ is the smallest value of x, $C(x)$ is the Cartesian tree with i as its root, the Cartesian tree of $x[1 \dots i-1]$ as the left subtree and the Cartesian tree of $x[i+1 \dots n]$ as the right subtree.

2.2 Cartesian tree matching

The Cartesian tree matching (CTM) problem consists of finding all factors of a text which share the same Cartesian tree as a pattern. In order to solve this problem efficiently without building every Cartesian tree, Park *et al.* [4] introduced a linear representation that has a one-to-one mapping with these trees called the parent-distance representation. Park *et al.* gave linear-time solutions for single and multiple pattern Cartesian tree matching, utilizing this parent-distance representation and existing classical string algorithms.

2.3 Approximate Cartesian tree matching

We use the notion of swap (or transposition) on sequences to define an approximate version of Cartesian tree matching. We seek to characterize what happens in the Cartesian tree framework when we allow up to one transposition of two adjacent symbols in a sequence, which amounts to moving the leftmost descendant of the right subtree to a rightmost position in the left subtree at the swap position, or vice versa. In order to do so, we also define the reverse parent-distance representation, computed similarly to the parent-distance, albeit as if read from right to left.

3. Results and Discussion

3.1 Characterization of the parent-distance tables

We describe how the parent-distances $\overrightarrow{PD}_x$ and $\overleftarrow{PD}_x$ of a sequence x of length n are modified into tables $\overrightarrow{PD}_y$ and $\overleftarrow{PD}_y$ when there is one swap between x and y. Figure 1 sums up the different parts of the parent-distance tables that we characterize. We show that the green zones are equal, the blue zones are equal, and taht values in the red zones differ by at most 1. Moreover, the eight values $\overrightarrow{a_x}, \overrightarrow{a_y}, \overrightarrow{b_x}, \overrightarrow{b_y}$... found at the positions of the swap verify a simple set of rules. Figure 2 provides a simple, thorough example.

With this newfound information, we devise a parent-distance based algorithm to solve the approximate CTM problem that has a $\Theta(mn)$ worst-case time complexity and a $\Theta(m)$ space complexity.

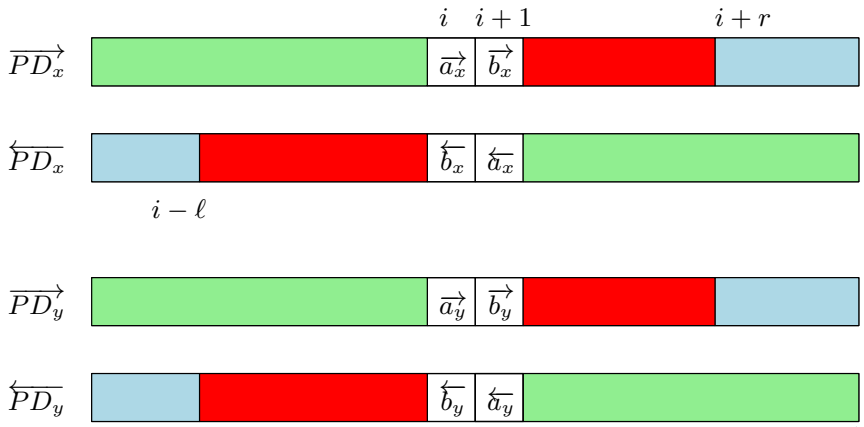

Figure 1. The different zones of both parent-distance tables. The left (resp. right) green zones of $\overrightarrow{PD}_x$ and $\overrightarrow{PD}_y$ are equal. Likewise for the blue zones. Each value in the red zones is either equal to or differs by 1 with its counterpart (+1 or -1 depending on the side).

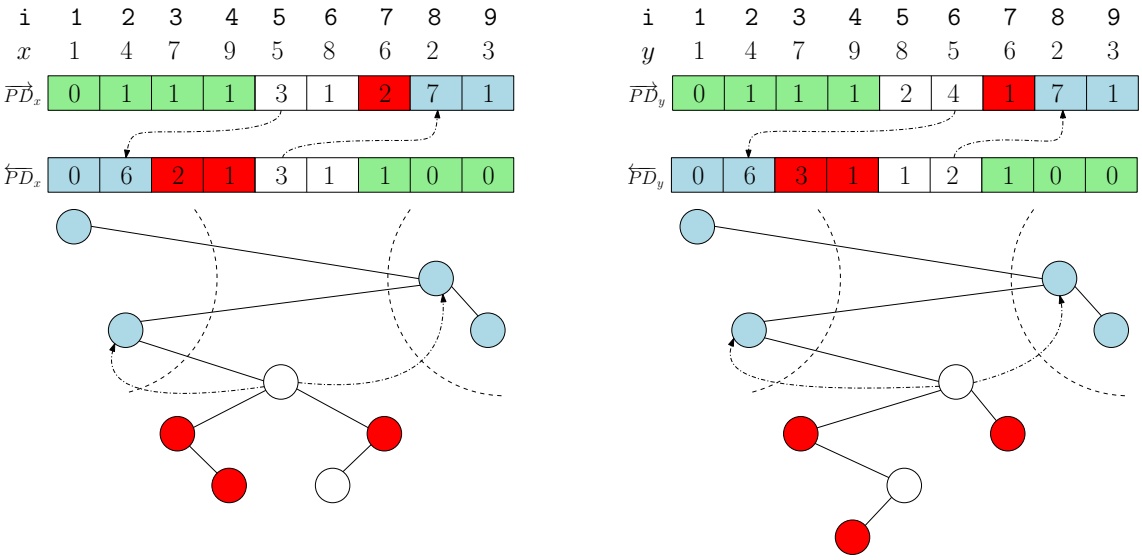

Figure 2. A detailed example of the properties mentioned above. The elements at position 5 and 6 were swapped.

3.2 Swap Graph

We define the swap graph of Cartesian trees for a given n as a graph where:

- The vertices are the Cartesian trees of size n
- There is an edge between two vertices x and y if x and y are one swap away from each other.

Figure 3 shows the swap graphs for n smaller than 4.

We show that the number of neighbours $|ng(T)|$ a single tree T has is bounded. We have :

$$n-1 \leq |ng(T)| \leq \lceil 3(n-1) - 2(\log_2(n+1) - 1) \rceil$$

From that, we can also obtain a lower bound on the graph's diameter.

We propose a second, Aho-Corasick based algorithm, similar to the multiple pattern matching solution given by Park *et al.* [4]. Given a sequence, we first compute

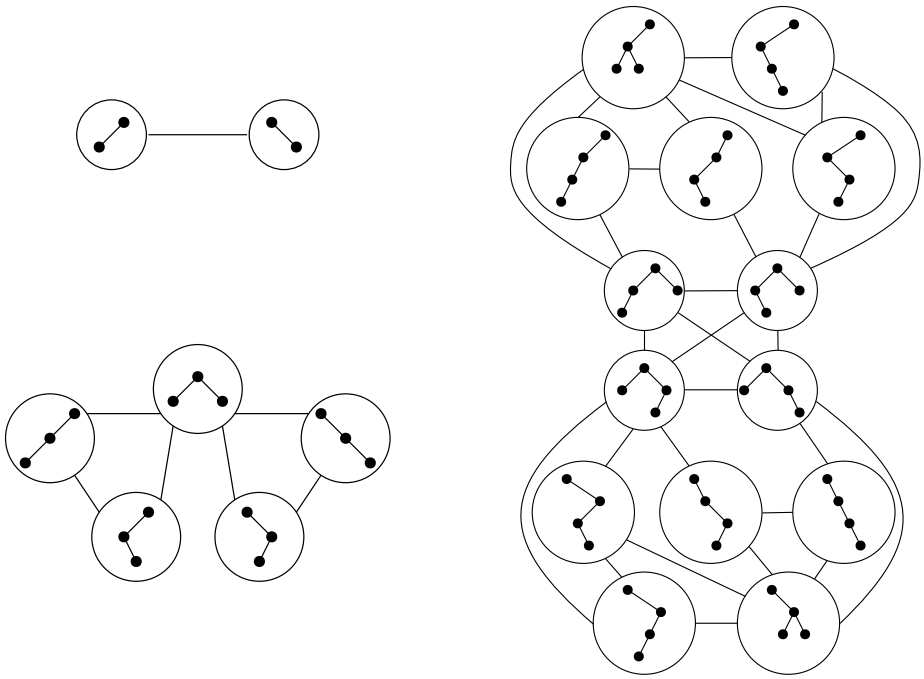

Figure 3. Swap graphs of Cartesian trees of size $2, 3$ and 4.

its set of neighbours, then the set of all their parent-distance tables and finally the automaton that recognizes this set of tables. This algorithm has a $\mathcal{O}((m^2+n)\log m)$ worst-case time complexity and a $\mathcal{O}(m^2)$ space complexity.

3.3 Perspectives

Generalizing our results to sequences with a partial order instead of a total one, obtaining a more general method (where the number of swaps could be given as a parameter), considering other errors than swaps, other representations than the parent-distance or using indexing methods are all options worth pondering.

The analysis of both algorithms should be improved. An amortized analysis could be done on the first algorithm and on the computation of the set of parent-distance tables for the second algorithm.

Acknowledgments

This work was supported by the NormaSTIC Federation https://www.normastic.fr/.

References

[1] Bastien Auvray, Julien David, Richard Groult, and Thierry Lecroq. Approximate cartesian tree matching: An approach using swaps. In Franco Maria Nardini, Nadia Pisanti, and Rossano Venturini, editors, *String Processing and Information Retrieval - 30th International Symposium, SPIRE 2023, Pisa, Italy, September 26-28, 2023, Proceedings*, volume 14240 of *Lecture Notes in Computer Science*, pages 49–61. Springer, 2023.

[2] S Faro and T Lecroq. The exact online string matching problem: a review of the most recent results. *ACM Comput. Surv.*, 45(2):13, 2013.

[3] Jean Vuillemin. A unifying look at data structures. *Commun. ACM*, 23(4):229–239, 1980.

[4] Sung Park, Amihood Amir, Gad Landau, and Kunsoo Park. Cartesian tree matching and indexing. In *CPM*, volume 16, pages 1–14, Pisa, Italy, 2019.

Abstract

Mathematical model of phylogenetic compression

Veronika Hendrychová[1]*, Karel Břinda[1]*

[1]*Inria, IRISA, Univ. Rennes, 35042 Rennes, France*

***Corresponding authors**: veronika.hendrychova@etudiant.univ-rennes1.fr, karel.brinda@inria.fr

Abstract

Comprehensive genome collections play a pivotal role in life sciences research. However, their exponential growth exceeds the pace of development of computational capacities, which renders genome storage and analysis increasingly difficult [6]. For instance, the proportion of data searchable using the Basic Local Alignment Search Tool (BLAST) [1] and its successors has been decreasing exponentially over time [5]. While substantial efforts have recently been devoted to the development of highly optimized alignment-based [3, 7] and k-mer based approaches [9], these provide one-time improvements rather than a systematic solution of the underlying scalability challenge.

One fundamental way to address the data explosion has been proposed via so-called compressive genomics [8, 10], a paradigm exploiting the geometrical structure of genomic data to design entropy-scaling algorithms that would be sub-linear in space and time. However, despite its theoretical benefits, compressive genomics has not yet been fully implemented on a large scale in usable tools, due to the underlying practical challenges such as how to efficiently identify redundancies across large and diverse genome collections, especially those arising in bacterial genomics [2, 4].

A recent work introducing a so-called phylogenetic compression, inspired by compressive genomics, has shown that, by using evolutionary history to guide existing algorithms and data structures, we can improve the state-of-the-art methods for compression and search of large and diverse bacterial genome collections by 1–2 orders of magnitude [5]. However, in spite of the clear performance improvement of phylogenetic compression, its theoretical foundations, including a mapping to the previously developed theory of entropy-scaling algorithms, are yet to be established.

In this talk, we develop a formal framework to study the compression capabilities of phylogenetic compression. To do so, we select one protocol of phylogenetic compression and formalize data compression as an optimization problem. We show that despite the fact that the problem itself might be NP-hard (as it can be reformulated as a specific variant of the Traveling Salesman Problem), when input data are modeled by a simplified, yet realistic evolutionary models, phylogenetic compression solves this problem optimally in polynomial time using algorithms that are efficient in practice. Finally, using a series of computational experiments, we demonstrate that these theoretical results well correspond to the compression performance observed in practical applications.

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, Oct 1990.

[2] P. Bradley, H. C. den Bakker, E. P. C. Rocha, G. McVean, and Z. Iqbal. Ultra-fast search of all deposited bacterial and viral genomic data. *Nat. Biotechnol.*, 37(2):152–159, Feb 2019.

[3] B. Buchfink, C. Xie, and D. H. Huson. Fast and sensitive protein alignment using diamond. *Nat. Methods*, 12(1):59–60, Jan 2015.

[4] G. A. Blackwell et al. Exploring bacterial diversity via a curated and searchable snapshot of archived dna sequences. *PLoS Biol.*, 19(11):e3001421, Nov 2021.

[5] K. Břinda et al. Efficient and robust search of microbial genomes via phylogenetic compression. *bioRxiv*, Apr 2023. doi: 10.1101/2023.04.15.536996.

[6] P. Muir et al. The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biol.*, 17:53, Mar 2016.

[7] H. Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, Sep 2018.

[8] P.-R. Loh, M. Baym, and B. Berger. Compressive genomics. *Nat. Biotechnol.*, 30(7):627–630, Jul 2012.

[9] C. Marchet, C. Boucher, S. J. Puglisi, P. Medvedev, M. Salson, and R. Chikhi. Data structures based on k-mers for querying large collections of sequencing data sets. *Genome Res.*, 31(1):1–12, Jan 2021.

[10] Y. W. Yu, N. M. Daniels, D. C. Danko, and B. Berger. Entropy-scaling search of massive biological data. *Cell Syst*, 1(2):130–140, Aug 2015.

Abstract

SeqBIM

Kmer2Reads an associative index for Third Generation Sequencing data

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

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

***Corresponding author**: lea.vandamme@univ-lille.fr and antoine.limasset@univ-lille.fr

Abstract

Studying biological sequences typically involves using a reference genome, but obtaining accurate assemblies from sequencing data can be challenging due to genomic repeats, errors, and biases. Hence, working directly with raw data output by sequencers, without pre-processing, can be preferable. Our objective is to develop multifaceted indexes able to identify reads containing a specific k-mer in a given dataset. Popular indexes, dubbed colored de Bruijn graphs associate the k-mer origin among thousand of datasets. However they are not able to index each reads separately. Additionally, with a large number of colors, such as in the human genome, colored de Bruijn graphs are not able to scale up.

To address this challenge, we present K2R, which leverages redundancy in the data to limit memory usage and rely on very efficent compression technique. Specifically, we use super-k-mers to reduce the number of entries in our structures and employ the concept of color to minimize memory impact of repetitive k-mer data. Furthermore, we will present the results that do not include low abundance k-mers (present in a single read) and their impact on query results.

We present the main results obtained by comparing K2R with state-of-the-art methods such as hashing methods [1] and full-text indexing (e.g., r-index [2]), in terms of memory impact, throughput, and time consumption for creation and query.

References

[1] Camille Marchet, Lolita Lecompte, Antoine Limasset, Lucie Bittner, and Pierre Peterlongo. A resource-frugal probabilistic dictionary and applications in bioinformatics. *Discrete Applied Mathematics*, 274:92–102, 2020. Stringology Algorithms.

[2] Travis Gagie, Gonzalo Navarro, and Nicola Prezza. Fully functional suffix trees and optimal text searching in BWT-runs bounded space. *Journal of the ACM*, 67(1):1–54, jan 2020.

Abstract

SeqBIM

Improved sub-genomic RNA prediction with the ARTIC protocol

Thomas Baudeau[1]*, Kristoffer Sahlin[2]

[1] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000, Lille, France*
[2] *Department of Mathematics, Science for Life Laboratory, Stockholm University, 106 91, Stockholm, Sweden*
***Corresponding author**: thomas.baudeau@univ-lille.fr

Abstract

Viral subgenomic RNA (sgRNA) is a unique biological mechanism observed in certain RNA-positive viruses, including those from the Coronaviridae family, such as SARS-COV-2. These sgRNAs play a pivotal role in the virus's replication cycle, pathogenicity, and evolution. The production of sgRNAs in viruses like SARS-COV2 involves specific sequences, and they are crucial for the virus's life cycle. They not only influence the virus's pathogenicity but could also have a significant impact on its evolution. The ARTIC[1] protocol, especially when combined with Oxford Nanopore Technologies (ONT) sequencing, offers a promising avenue for viral monitoring during epidemics. However, the inherent errors in the sequencing data present computational challenges.

To address the computational challenges of sgRNA analysis we developed, sgGENERATE: This evaluation pipeline was introduced to assess the accuracy and efficacy of sgRNA detection tools that utilize the ARTIC sequencing protocol in mimicking the specificity on such data. Through sgGENERATE, we evaluated "periscope[2]," a tool designed to detect sgRNA from ARTIC sequencing data. The findings revealed that periscope has biased predictions and requires high computational resources. Periscope Multi: Building on the insights from sgGENERATE, we redesigned the periscope algorithm. The new version, named "periscope multi," uses multiple references from canonical sgRNAs to address alignment challenges and enhance both sgRNA and non-canonical sgRNA detection. Evaluations on simulated and biological sequencing datasets showed that periscope multi offers a significant improvement in sgRNA detection accuracy.

In conclusion, this research advances the available tools for studying viral sgRNA, paving the way for more accurate and efficient analyses in the realm of viral RNA discovery.

References

[1] Quick & al. Multiplex pcr method for minion and illumina sequencing of zika and other virus genomes directly from clinical samples. *Nature protocols*, 12(6):1261–1276, 2017.

[2] Matthew D. Parker & al. Subgenomic RNA identification in SARS-CoV-2 genomic sequencing data. *Genome Research*, 31(4):645–658, March 2021.

SeqBIM

Assessing alternatively spliced transcript diversity with long reads

Lilian Marchand[1]*, Hélène Touzet[1], Jean-Stéphane Varré[1]

[1]*Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL, 59000 Lille, France*
***Corresponding author**: lilian.marchand@univ-lille.fr

Abstract

We introduce RNA-tailor, a novel tool designed to precisely inventory the repertoire of alternatively spliced transcripts of a target gene from third-generation sequencing data. Alternative splicing (AS) is a regulation mechanism that enables the production of various RNA isoforms, forming proteins involved in regulation of gene expression, tissue specialization, or immune response. In this context, accurate identification of the alternatively spliced transcript diversity can be crucial information. For the study of AS, long read (LR) sequencing technology are preferred to short read as they should span the full length of the transcripts, allowing to capture the combinatorics of exons. Analyzing LRs requires specific computational approaches to take full advantage of their length despite their higher error sequencing rate [1]. The aim of RNA-tailor is to provide a nucleotide-level precise picture of all possible alternative transcripts of a given target gene without any annotation data. It takes as input an RNA-seq dataset and a reference sequence (the gene of interest or the genome and the locus of the gene). Reads of interest are selected by mapping them on the reference either using Megablast [2] or Minimap2 [3]. Selected reads are corrected using isONcorrect [4]. Corrected reads are next realigned against the gene reference sequence using the splice-aware alignment tool Exonerate [5]. We then identify potential misalignments in the remaining reads. Read segments poorly supported that could be realigned elsewhere are realigned locally. Transcripts are clustered according to their predicted intronic junction structure, to account for read completeness issue. Finally, RNA-tailor outputs a GTF file of isoforms and a XLSX spreadsheet allowing to visualize the exonic structure of each read.

References

[1] N. Kono and K. Arakawa. Nanopore sequencing: Review of potential applications in functional genomics. *Development, Growth & Differentiation*, 61(5):316–326, June 2019.

[2] S. F. Altschul et al. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.

[3] H. Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, September 2018.

[4] K. Sahlin and P. Medvedev. Error correction enables use of Oxford Nanopore technology for reference-free transcriptome analysis. *Nature Communications*, 12(1):2, January 2021.

[5] G. Slater and E. Birney. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, 6(1):31, February 2005.

Abstract

SeqBIM

A space-efficient, locality-preserving and dynamic data structure for indexing *k*-mers

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

[1] *Univ. Lille, CNRS, CRIStAL, Lille, France*
***Corresponding author**: igor.martayan@ens-rennes.fr

Abstract

Because of the unprecedented amount of genomic data available, designing space-efficient data structures for indexing DNA sequences has never been that essential. In particular, sets of k-mers are a fundamental object prior to downstream analysis, whether for assembling genomes, mapping or searching sequences. In this work, we are interested in designing a space-efficient data structure for k-mer sets satisfying the following properties:

- *dynamic*: it should support insertion and deletion of k-mers after construction.
- *efficient queries*: membership queries should require $\mathcal{O}(1)$ memory accesses.
- *locality-preserving*: queries on consecutive k-mers should access a contiguous region of the memory in order to reduce cache misses.

State-of-the-art data structures usually do not fit all these criterias. For instance, full-text indexes are compact but have expensive membership queries. In 2011, Conway-Bromage [1] introduced the idea of encoding k-mers in a bitset of size 4^k where each bit correspond to a k-mer. Since this bitset is usually very sparse, its space usage can be reduced using Elias-Fano coding [2]. While this data structure supports efficient queries and is both dynamic [3] and space-efficient in a sparse setting, it maps consecutive k-mers to very distant regions of the memory, leading to a lot cache misses in practice.

We propose a variation of this data structure that uses a bijective transformation of the k-mer space in order to improve the locality of consecutive k-mers. This transformation is based on computing the *necklace* of a k-mer, that is, its lexicographically smallest cyclic rotation, and the rank of this necklace. The benefit of this representation is that the necklaces of consecutive k-mers likely share a long prefix (corresponding to a common minimizer), thus mapping them close to each other. What's more, this transformation can be combined with a well-chosen k-mer encoding so that each k-mer and its reverse-complement are mapped to the same entry in $[\![0, 4^k/2 - 1]\!]$. While computing the rank of a necklace requires $\mathcal{O}(k^2)$ time using Sawada's algorithm [4], we observe a better compression of the bitset after ranking.

References

[1] Thomas C Conway and Andrew J Bromage. Succinct data structures for assembling large genomes. *Bioinformatics*, 27(4):479–486, 2011.

[2] Peter Elias. Efficient storage and retrieval by content and address of static files. *Journal of the ACM (JACM)*, 21(2):246–260, 1974.

[3] Giulio Ermanno Pibiri and Rossano Venturini. Dynamic elias-fano representation. In *28th Annual symposium on combinatorial pattern matching (CPM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[4] Joe Sawada and Aaron Williams. Practical algorithms to rank necklaces, lyndon words, and de bruijn sequences. *Journal of Discrete Algorithms*, 43:95–110, 2017.

Abstract

SeqBIM

Recent advancements in approximate colored compacted de Bruijn graph representations

Yoshihiro Shibuya[1]*, Pierre Peterlongo[2], Giulio Ermanno Pìbiri[3,4]

[1] *Institut Pasteur, Paris, France*
[2] *INRIA, CNRS, IRISA, Univ. Rennes, Rennes, France*
[3] *DAIS, Ca'Foscari University of Venice, Venice, Italy*
[4] *ISTI-CNR, Pisa, Italy*
***Corresponding author**: yoshihiro.shibuya@pasteur.fr

Abstract

The problem of finding the subset of references that are likely to contain a given query sequence is of central to many areas in modern bioinformatics, due to the rapidly growing collections of sequencing data. Size and number of modern datasets do not allow naive searches, leading to the need of pre-processing the references into an index supporting fast queries. Current, competitive methods split references and queries into sets of their constituent k-mers (substring of length k) in order to avoid costly alignments [1]. Assigning k-mers to the set of references they belong to is a problem best formalized by *colored compacted* de Bruijn graphs (ccdBGs).

Themisto [2] and Fulgor [3] are two recent ccdBG indexes. Both are *exact* data structures, able to recognize the presence/absence of k-mers from the ccdBG in addition to retrieving their colors. However, in applications where false positives can be tolerated, approximate indexes can offer great advantages in terms of space and complexity of the queries. k-mers in Fulgor are internally stored as unitigs (paths of the ccdBG sharing the same color) by SSHash [4], which can represent a non-negligible part of the whole index (23% and 69% for Fulgor and mscdBG, respectively, when indexing 10000 *S.Enterica* genomes [5]).

This talk presents *kaminari*, a work-in-progress implementation of approximate ccdBGs based on Fulgor, where SSHash is replaced by LPHash [6], a locality-preserving MPHFs, and the design challenges arising from doing so.

References

[1] Camille Marchet, Christina Boucher, Simon J. Puglisi, Paul Medvedev, Mikaël Salson, and Rayan Chikhi. Data structures based on k-mers for querying large collections of sequencing data sets. *Genome Research*, 31(1):1–12, January 2021.

[2] Jarno N Alanko, Jaakko Vuohtoniemi, Tommi Mäklin, and Simon J Puglisi. Themisto: a scalable colored k-mer index for sensitive pseudoalignment against hundreds of thousands of bacterial genomes. *Bioinformatics*, 39(Supplement_1):i260–i269, 06 2023.

[3] Jason Fan, Noor Pratap Singh, Jamshed Khan, Giulio Ermanno Pibiri, and Rob Patro. Fulgor: A Fast and Compact {k-mer} Index for Large-Scale Matching and Color Queries. In Djamal Belazzougui and Aïda Ouangraoua, editors, *23rd International Workshop on Algorithms in Bioinformatics (WABI 2023)*, volume 273 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:21, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[4] Giulio Ermanno Pibiri. Sparse and skew hashing of K-mers. *Bioinformatics*, 38(Supplement_1):i185–i194, 06 2022.

[5] Giulio Ermanno Pibiri, Jason Fan, and Rob Patro. Meta-colored compacted de Bruijn graphs: overview and challenges, July 2023.

[6] Giulio Ermanno Pibiri, Yoshihiro Shibuya, and Antoine Limasset. Locality-preserving minimal perfect hashing of k-mers. *Bioinformatics*, 39(Supplement_1):i534–i543, 06 2023.