Résumés des exposés de

SeqBIM
2021 – 25-26 nov.
Lyon (France)

25 et 26 novembre 2021

Lyon

Exposé invité

Back and forth in pangenomics : data structures for querying large collections of sequence data

Paola Bonizzoni (AlgoLab, Université de Milan, Italie)

Résumé

The speed in producing large amounts of genome data, driven by advances in sequencing technologies, is far from the slow progress in developing new methods for analyzing multiple related genomes. Most recent advances in the field are still based on notions rooted in established and quite old literature on combinatorics on words and space-efficient data structures.

In this talk we will go back and forth through the state-of-art with the goal of analyzing query operations and data structures that may help in managing and analyzing multiple genomes : the theoretical foundations of computational pangenomics.

Exposé invité

Cartesian Pattern Matching

Thierry Lecroq (équipe TIBS, LITIS, Rouen)

Résumé

Cartesian trees are associated to strings of numbers. They are structured as heap and original strings can be recovered by symmetrical traversal of the trees. Let x be a string of numbers of length m. The Cartesian tree of x is the binary tree where :

— the root corresponds to the index i of the minimal element of x (if there are several occurrences of the minimal element, the leftmost one is chosen) ;
— the left subtree of the root corresponds to the Cartesian tree of x[1..i-1] ;
— the right subtree of the root corresponds to the Cartesian tree of x[i+1..m]. Cartesian pattern matching can be applied to find patterns in time series data.

In this talk, we will review the existing Cartesian pattern matching algorithms and describe more in details solutions for the following problems :

— given a text and a pattern that consist of sequences of numbers, find all the substrings of the text that have the same Cartesian tree than the pattern ;
— given a text and a finite set of patterns that consist of sequences of numbers, find all the substrings of the text that have the same Cartesian tree than one of the patterns.
— given two strings that consist of sequences of numbers, find the length of the longest substring of both strings that have the same Cartesian tree.

Abstract

SeqBIM

Space-efficient representation of genomic k-mer count tables

Yoshihiro Shibuya[1]*, Djamal Belazzougui[2], Gregory Kucherov[1,3]

[1] *Laboratoire d'Informatique Gaspard Monge, CNRS & Université Gustave Eiffel, Marne-la-Vallée*
[2] *CAPA, DTISI, Centre de Recherche sur l'Information Scientifique et Technique, Algiers, Algeria*
[3] *Skolkovo Institute of Science and Technology, Moscow, Russia*
***Corresponding author**: yoshihiro.shibuya@univ-eiffel.fr

Abstract

Counting k-mers is one of the fundamental tasks in bioinformatics with many available tools to choose from [1, 2], producing count tables containing both k-mers and counts. In many applications, the set of k-mers is known making the k-mers inside the table redundant. Therefore, it makes sense to store counts independently from their k-mers in order to be able to use a more memory efficient implementation supporting fast random-access queries.

Here we present *locom* [3], an efficient representation of k-mer count tables supporting fast random-access queries. Our algorithm makes use of a recently proposed implementation of *Compressed Static Functions* [4] together with Bloom Filters to achieve space close to the empirical zero-order entropy of the counts. We call this extention *Bloom-enhanced CSF* or *BCSF* for short. Our second contribution is the combination of BCSF with minimizer-based bucketing of counts producing even more efficient representations, with the ability to break the entropy lower-bound, for large enough k. Finally, we extend our idea to the approximate case, gaining additional space at the cost of a user-defined absolute error over counts.

References

[1] Guillaume Rizk, Dominique Lavenier, and Rayan Chikhi. DSK: k-mer counting with very low memory usage. *Bioinformatics*, 29(5):652–653, March 2013.

[2] Marek Kokot, Maciej Długosz, and Sebastian Deorowicz. KMC 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761, 05 2017.

[3] Yoshihiro Shibuya, Djamal Belazzougui, and Gregory Kucherov. Space-Efficient Representation of Genomic k-Mer Count Tables. In Alessandra Carbone and Mohammed El-Kebir, editors, *21st International Workshop on Algorithms in Bioinformatics (WABI 2021)*, volume 201 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[4] Marco Genuzio, Giuseppe Ottaviano, and Sebastiano Vigna. Fast scalable construction of ([compressed] static | minimal perfect hash) functions. *Information and Computation*, 273:104517, August 2020.

Abstract

SeqBIM

Answer Set Programming based haplotype phasing of long read for di-polyploid species

Clara Delahaye[1]*, Jacques Nicolas[2]

[1,2] *Univ Rennes, Inria, CNRS, IRISA F-35000, Rennes, France*
***Corresponding author**: clara.delahaye@irisa.fr

Abstract

Diploid and polyploid species have their chromosomes present in at least two copies called *haplotypes*. Although haplotypes are highly similar, they differ through the presence of variants that can be of high interest as they may be related to biological processes or genetic diseases. However, most of reference genomes available today are consensus *monoploid* genomes, *i.e.* only one sequence per chromosome is given, merging variants found, and thus leading to missing or erroneous information.

This has led to the rise of new assembly methods attempting to provide genomes that integrate haplotype information: *haplotype phasing* methods. However, most of these methods are designed for short reads and/or diploid species only. While short reads provide accurate data they struggle on repeated regions of genome, whereas long reads ease haplotype phasing by spanning wider regions of the genome. Moreover, haplotype phasing of diploid genomes is now quite well handled as it is reduced to a binary choice, e.g. decide if the read belong to the first haplotype. Reasoning is more complex in the case of polyploid phasing. One strategy is to cluster reads based on their similarity (e.g. using cliques of overlapping reads [1] or a scoring function [2]): reads belonging to a given cluster are expected to originate from a same haplotype.

Here we propose a combinatorial method for diploid and polyploid haplotype phasing of long read data. We address the haplotype phasing as an optimization problem and use *Answer Set Programming* [3] (ASP), with clingo system to solve it . Rather than providing a unique and likely erroneous answer to this hard problem, the ASP framework allows to reason on the set of possible solutions. Moreover, ASP is a high-level declarative language that offers both efficiency (inspired on SAT-solver techniques) and expressiveness (more than ILP for example): the user can easily express preferences and get a global view of confident and ambiguous phased regions.

Starting from reads and related variant information, we construct a graph of reads and split it into connected components that will be phased independently. For the phasing steps, we try to build haplotypes based on a minimization of differences between reads of a same haplotype, taking into account potential unknown errors in reads. The phased fragments are then re-assembled to produce the final haplotypes.

The overall (still ongoing) method will be designed in two complementary parts: a rather traditional one computing similarities between reads; and a combinatorial one that will interact with the user to explore the set of possible phasing solutions.

References

[1] Jasmijn A. Baaijens, Amal Zine El Aabidine, Eric Rivals, and Alexander Schönhuth. De novo assembly of viral quasispecies using overlap graphs. *Genome Research*, 27(5):835–848, April 2017.

[2] Sven D. Schrinner, Rebecca Serra Mari, Jana Ebler, Mikko Rautiainen, Lancelot Seillier, Julia J. Reimer, Björn Usadel, Tobias Marschall, and Gunnar W. Klau. Haplotype threading: accurate polyploid phasing from long reads. *Genome Biology*, 21(1), September 2020.

[3] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Answer set solving in practice. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(3):1–238, December 2012.

Abstract

VirHunter: a deep learning-based method for detection of novel viruses in plant sequencing data

Grigorii Sukhorukov[1,2]*, Macha Nikolski[1,2]

[1]*University of Bordeaux, CNRS, IBGC, UMR 5095, Bordeaux, France*
[2]*University of Bordeaux, Bordeaux Bioinformatics Center, Bordeaux, France*
***Corresponding author**: grigorii.sukhorukov@u-bordeaux.fr*

Abstract

Plant viruses are a major plant pathogen; plant infections by viruses result in more than $30 billion losses a year [1]. One of the best ways to fight against plant infections caused by viruses is the surveillance and early detection of viral presence in plant populations. For this task high-throughput sequencing (HTS) is often used as an excellent tool to investigate the viral presence in large plant samples. However HTS data comprises a mixture of sequences containing not only viral sequences but also contaminating sequences coming from the host and bacteria. The downstream bioinformatics analysis of these data can be difficult and time consuming. Moreover, computational tools often miss certain viruses present in the sample, especially the previously uncharacterised viral species.

To help solve this task we have developed an artificial neural network approach that classifies sequences as belonging to viral, bacterial or plant host origin and is implemented in the Keras framework. The developed approach uses the one-hot encoding for both sequences and their reverse complement and relies on convolutional layers to learn k-mers distinguishing viral sequences. The network was trained on a dataset composed of three balanced classes corresponding to plant viruses, plants and bacteria. To be performant our VirHunter method has to be trained for the analysis of a specific plant's virome, as the training dataset is generated using the host information. We test our tool in different settings assessing its ability to detect novel viruses. Then we compare the performance of our tool to that of DeepVirFinder, a previously developed deep learning-based method for virus detection [2]. Finally, we validate the VirHunter on several real datasets.

References

[1] Sastry, K. Subramanya. Plant Virus and Viroid Diseases in the Tropics: Volume 1: Introduction of Plant Viruses and Sub-Viral Agents, Classification, Assessment of Loss, Transmission and Diagnosis. *Springer Science & Business Media*, 2013.

[2] Ren J, Song K, Deng C, Ahlgren NA, Fuhrman JA, Li Y, Xie X, Poplin R, Sun F. Identifying viruses from metagenomic data using deep learning. *Quantitative Biology*, 22:1-4, 2020.

Abstract

SeqBIM

Parametrized algorithms for consensus problems with swaps

Estéban Gabory[1], Supervised by Laurent Bulteau[2]

[1] CWI, Amsterdam, The Netherlands
[2] Laboratoire d'Informatique Gaspard-Monge, Marne-La-Vallée, France

Abstract

Finding a string s (a "consensus") being as close as possible from a given set of strings S is a classic type of problem in string optimization: it has application every time one wants to find an original string s if S is a set of copies of s with possible errors, or mutations. It has been studied under the Hamming distance d_H, and we extend here some known results by allowing to swap two adjacent symbols, which is a common type of copying error. Let d be a distance on strings, we study 3 versions of the problem: minimize the maximal distance, or the sum of distances, or both. We call those problems respectively (r)-CLOSEST STRING$_d$, (s)-CLOSEST STRING$_d$, and (r,s)-CLOSEST STRING$_d$. It is known that unless $P = NP$, only (s)-CLOSEST STRING$_{d_H}$ can be computed in polynomial time, but it also has been shown that (r)-CLOSEST STRING$_{d_H}$ and (r,s)-CLOSEST STRING$_{d_H}$ are fixed-parameter tractable for d, the smallest maximal distance for which a solution exists. To extend those results by allowing swaps, which are support disjoint exchanges of two adjacent distinct symbols in the strings, we define the swap distance (allowing only swaps between two strings and counting them) and the SH distance (we allow swap and mismatches and count them, both having the same cost). We show that every mentioned consensus problem under the swap distance can be reduced to an equivalent problem under the Hamming distance in less than $O(kn)$ steps, with k and n being the number of strings and their length. Then, we give a dynamic algorithm to solve (s)-CLOSEST STRING$_{d_{SH}}$ in polynomial time, and we extend the FPT method for (r)-CLOSEST STRING$_{d_H}$ to (r)-CLOSEST STRING$_{d_{SH}}$ and (r,s)-CLOSEST STRING$_{d_{SH}}$.

References

[1] Jens Gramm, Rolf Niedermeier, and Peter Rossmanith. Fixed-parameter algorithms for closest string and related problems. *Algorithmica*, 37:25–42, 09 2003.

[2] Amihood Amir, Haim Paryenty, and Liam Roditty. On the hardness of the consensus string problem. *Information Processing Letters*, 113:371–374, 05 2013.

[3] Laurent Bulteau and M. Schmid. Consensus strings with small maximum distance and small distance sum. In *MFCS*, 2018.

[4] M. Frances and A. Litman. On covering problems of codes. *Theory of Computing Systems*, 30:113–119, 2007.

Abstract

SeqBIM

Algorithms for searching dinucleotidic Position Weight Matrices (di-PWM)

Marie Mille[1], Julie Ripoll[1], Bastien Cazaux[1], Eric Rivals[1]

[1]*LIRMM, Montpellier University, CNRS, UMR 5506, Montpellier, France*
***Corresponding author**: rivals@lirmm.fr

Abstract

Transcription regulation is an important cellular process. Specialized proteins, called Transcription Factors (TF), bind on short, specific, DNA sequences to regulate the expression of nearby genes. The sequences recognized by a TF in the vicinity of different genes are not identical, but similar. One captures the similarity of those binding site in different representations, which are generally called *motifs*. The most widely used sort of motifs are Position Weight Matrix (PWM) (also known as a position-specific weight matrix (PSWM) or position-specific scoring matrix (PSSM)). A PWM is built from a multiple alignment of "true" binding sequences and capture the observed variation of nucleotides at the different positions. Several databases (JASPAR, TRANSFAC, etc.) collect PWMs for known TFs. Those PWMs are used to scan new DNA sequences to find putative binding sites and possibly to annotate them. In the case of complete genomes, the scanning procedure for many PWM may last a long time [1].

PWM assume that the distinct positions of the bound sequence are independent of each other. However, several works have observed that a mutation at given position influence the probability of mutation at neighboring positions. To overcome this limitation of PWMs, Kulakovskiy et al. have proposed a more complex sort of motif, called di-PWMs, which model the frequency of occurrence of dinucleotides in the binding sites (instead of mononucleotides for PWMs) [2]. Their studies show that di-PWMs improve in sensitivity compared to PWMs, and thus produce less false positives when scanning a sequence. Many search algorithms are available for mononucleotidic PWM, but only one exist for di-PWMs [1].

We propose two search algorithms for di-PWMs: the first one is a scanning window algorithm with some adapted speed up trick, the second one is enumeration based. The online scanning algorithm computes a partial score for some positions in the current window, and estimates the maximum achievable score for the whole window. If this score does not match requested threshold, the window can be discarded. A new precomputed table is provided and compare to a classic LookAheadTable [3].

The enumeration strategy relies on the observation that searching for exact matches is faster than computing window scores. The underlying idea is to first enumerate all *valid words* (i.e., words whose score lies above the user defined score threshold) and their score, then in a second phase to search for the set of valid words using any algorithm that solves the Set Pattern Matching problem [4]. Here

for this sake, we used a Python module that implements the classical Aho-Corasick automaton [5].

We also conducted running time experiments for searching di-PWMs from the HOCOMOCO database [6] with both algorithms, and compared our Python implementations to a tool written in Java, called SPRY-SARUS [1].

Numerous perspectives of this work can be considered, including off-line search of the set of valid words in a precomputed genome index (as done for PWM within the MOTIF software [7]).

A presentation in French of these algorithms can be found in [8]. The di-PWM search algorithms will soon be available as a Python package entitled `dipwmsearch` (which can be installed with PyPI).

Acknowledgments: The internship of Marie Mille was kindly supported by the GEM Flagship project funded from Labex NUMEV (ANR-10-LABX-0020). Julie Ripoll is supported by the INCA project "FluoRib". ER thanks the support from the Marie-Curie ITN "ALgorithms for PAngenome Computational Analysis" (ALPACA).

References

[1] Cinzia Pizzi, Pasi Rastas, and Esko Ukkonen. Finding significant matches of position weight matrices in linear time. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(1):69–79, January 2011.

[2] Ivan Kulakovskiy, Victor Levitsky, Dmitry Oshchepkov, Leonid Bryzgalov, Ilya Vorontsov, and Vsevolod Makeev. From binding motifs in chip-seq data to improved models of transcription factor binding sites. *Journal of Bioinformatics and Computational Biology*, 11(01):1340004, Feb 2013.

[3] Michael Beckstette, Robert Homann, Robert Giegerich, and Stefan Kurtz. Fast index based algorithms and software for matching position specific scoring matrices. *BMC Bioinformatics*, 7(1), Aug 2006.

[4] Dan Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, 1997.

[5] Alfred Aho and Margaret Corasick. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18:333–340, 1975.

[6] Ivan V Kulakovskiy, Ilya E Vorontsov, Ivan S Yevshin, Ruslan N Sharipov, Alla D Fedorova, Eugene I Rumynskiy, Yulia A Medvedeva, Arturo Magana-Mora, Vladimir B Bajic, Dmitry A Papatsenko, and et al. HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Research*, 46(D1):D252–D259, Nov 2018.

[7] David Martin, Vincent Maillol, and Eric Rivals. Fast and accurate genome-scale identification of dna-binding sites. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 201–205, 2018.

[8] Marie Mille. Recherche de motifs probabilistes : le cas des Matrices Poids Position dinucléotidiques (di-PWM). Research report, Université de Montpellier, July 2021.

Abstract

SeqBIM

SpecGlob: a new Dynamic Programming Algorithm to interpret Mass Spectra

Albane Lysiak[1,3], Guillaume Fertin[1], Géraldine Jean[1], Dominique Tessier[2,3]

[1] *Université de Nantes, CNRS, LS2N, F-44000, Nantes, France*
[2] *INRAE, BIBS facility, F-44316, Nantes, France*
[3] *INRAE, UR BIA, F-44316*

This communication has been accepted as a poster to ASMS 2021

Abstract

In proteomics, one of the main reasons for the poor identification rate of mass spectra when they are compared to candidate peptides (CP) is that most of them correspond to the fragmentation of peptides carrying modifications. Open Modification Search (OMS) methods accept a wide range of mass difference within a Peptide-Spectrum Match (PSM) to improve the identification of spectra carrying modifications. Nevertheless, even if some of these methods are able to identify and localize a *single* modification, no efficient algorithm exists to interpret a PSM containing *several modifications*, especially without *a priori*.

We developed `SpecGlob`, an algorithm that interprets PSMs by realigning a CP to its spectrum, even when several unknown modifications have occured. For each PSM, `SpecGlob` uses dynamic programming to determine the best alignment between a spectrum and its CP, while allowing the insertion of possibly multiple mass offsets.

Given a PSM, `SpecGlob` outputs a sequence of amino acids interleaved by one or several mass offset(s), thus providing information on the modifications to apply to the CP so as to retrieve the spectrum sequence. Depending on the mass offsets values, these modifications are more or less difficult to infer, something we quantified in order to evaluate the quality of `SpecGlob`. For example, if DYSIR plays the role of the experimental spectrum and DWYIR is the CP, the ouptut of `SpecGlob` allows us to infer two modifications, namely deletion of W and insertion of S. Hence we consider this PSM interpretation to be complete, because suggested mass offsets correspond to known (combinations of) amino acids masses.

Theoretical peptides from the human proteome (Ensembl 99) were compared to each other (self-identification excluded) using the `SpecOMS` software [1]. Resulting PSMs were then processed by `SpecGlob`, which takes as input masses of both spectra. Altogether, `SpecGlob` completely interprets a large proportion of PSMs, even if they carry several scattered modifications. On the human dataset, `SpecGlob` returns a complete interpetation for roughly 30% of the 455,404 PSMs provided by `SpecOMS`. Our results also suggest that, even when a spectrum cannot be completely retrieved, a substantial portion of the initial amino acids sequence can still be determined.

References

[1] 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.

Abstract

GraphUnzip: unzipping assembly graphs with long reads and Hi-C

Roland Faure[1], Nadège Guiglielmoni[2]*, Jean-François Flot[3]

[1]*Université Rennes 1, Inria RBA, CNRS UMR 6074, Rennes, France*
[2]*University of Cologne (UOC), Germany*
[3]*Service Evolution Biologique et Ecologie, Université Libre de Bruxelles (ULB), Belgium*
***Corresponding author**: nadege.guiglielmoni@ulb.be

Abstract

The growing length and precision of sequencing reads, associated with the sharp decrease of their cost, has enabled great advances in the field of genome assembly. However, repeats in the genome remain a challenge to the assemblers. More specifically, homozygous regions can make it very difficult to assemble the different haplotypes of the sample separately. Collapsing all haplotypes into a single sequence is the most frequent strategy nowadays to obtain a contiguous assembly despite heterozygosity. The cost of this operation is the obtention of a sequence that is a medley of all actual haplotypes. We present an alternative to collapsing assemblies: GraphUnzip, a tool capable of *unzipping* (untangling) the assembly graph to recover the original haplotypes separately and solve genomic repeats, with Hi-C data and/or long reads. GraphUnzip naive approach makes no assumption on the ploidy or the heterozygosity rate of the data and can thus unzip very heterozygous genomes as well as genomes with high ploidy, or can be plainly used with long reads to solve repeats in any assembly graph.

Existing software either start by collapsing the assembly or *phase* the graph (provide a partition of contigs in several haplotypes, without linking the contigs) [1]. GraphUnzip is fast and memory-efficient tool that builds upon the assembly graph provided by most modern assemblers. As GraphUnzip only connects sequences in the assembly graph that already had a potential link based on overlaps, it yields high-quality gap-less supercontigs. It can exploits Hi-C data to correctly link very distant heterozygous regions.

To demonstrate the efficiency of GraphUnzip, we tested it on a simulated diploid *Escherichia coli* genome, and on two real datasets for the genomes of the rotifer *Adineta vaga* and the potato *Solanum tuberosum*. In all cases, GraphUnzip yielded highly continuous phased assemblies.

GraphUnzip implements a new post-assembly strategy, using the assembly graphs to obtain continuous sequences. It paves the way for the recovery of fully phased assemblies, where each contig represents a chromosome of a genome.

References

[1] Zhang X, Wu R, Wang Y, Yu J, Tang H. Unzipping haplotypes in diploid and polyploid genomes. *Comput Struct Biotechnol J.* 2019 Dec 9;18:66-72.

Abstract

SeqBIM

On the fly reduction of Bloom filter false positives

Lucas Robidou[1]*, Pierre Peterlongo[1]

[1] *Univ. Rennes, Inria, CNRS, IRISA, Rennes, France*
***Corresponding author**: lucas.robidou@inria.fr

Motivation:

The indexation of the vast amount of raw sequence data available today has recently received a lot of methodological attention [4]. However, the problem remains open, and no method, by far, may presently index the hundreds of petabytes of data stored at EBI, presently doubling every 26 months [2].

Tools that enable the indexing of largest volumes of genomic data use k-mers. A building block of these indexes is to attribute any queried k-mer to sample(s) it belongs to. This is made possible mainly thanks to Bloom filters as this is for instance the case for `BIGSI` [1] or `HowdeSBT` [3] to cite a few. Even if there exist variations and various optimizations, the core idea of these approaches is, for each sample, to index all its k-mers not considered as erroneous in a Bloom filter. When querying a k-mer, conceptually, all Bloom filters are queried, providing the information of the samples in which this k-mer occurs. Bloom filters are error-prone: false positive calls are possible (but false negatives are not). The precision depends on the amount of memory allocated.

Method overview:

We recall that TP stands for the number of true positive calls, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives. The false positive rate is defined as the proportion of FP among the ground truth negatives: $FPR = \frac{FP}{FP+TN}$ (usually called ϵ wrt Bloom filters).

We propose a method for reducing the FPR when querying K-mers from a sequence against a bank represented by a Bloom filter. Indeed, knowing the number of shared K-mers between the sequence and the bank allows to estimate the similarity between the sequence itself and the bank.

The key idea of our method is, given a query of length $\geq K$, to split each of its K-mer into $z+1$ k-mers ($K \geq k$) and to consider that a K-mer is found if and only if all its k-mers are found in the the Bloom filter.

Results:

The method we propose for reducing false positives has no drawback when used within the recommended set of parameters and allows to perform queries faster than a traditional Bloom filter with a single hash function.

Applied on real data (a sample from HMP), `findere` reduces the false positive rate from 5% to 0.056% (Fig. 1). Alternatively, fixing FPR=0.1%, `findere` reduces the size of the filter from 17 Go down to 0.16 Go, with no impact on the FPR.

As shown in Table 1 `findere` performs its queries about three times faster than

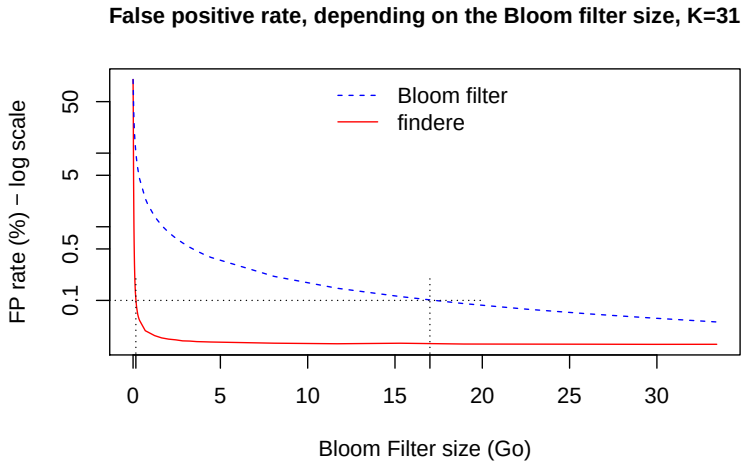

Figure 1. `findere` and BF FPR depending on the space used, on the hmp dataset. Dotted line segment corresponds to 0.1% false-positive rate.

traditional Bloom filters when using recommended $z = 3$ default value.

z	0	1	2	3	4	5	10
BF	42.4						
`findere`	42.9	43.7	24.3	17.5	14.1	12.0	8.6

Table 1. BF and `findere` query time in seconds on the hmp dataset, depending on the z value. BF result does not depend on z and is reported only for $z = 0$.

References

[1] Phelim Bradley, Henk C. den Bakker, Eduardo P.C. Rocha, Gil McVean, and Zamin Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. *Nature Biotechnology*, 37(2):152–159, feb 2019.

[2] ENA consortium. https://www.ebi.ac.uk/ena/about/statistics, 2021.

[3] Robert S Harris and Paul Medvedev. Improved representation of sequence Bloom trees. *Bioinformatics*, 2019.

[4] 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, jan 2021.

Abstract

SeqBIM

Advances in k-mer matrix construction for analysis of large sequencing collections

Téo Lemane[1]*, Rayan Chikhi[2], Pierre Peterlongo[1]

[1] *Univ. Rennes, Inria, CNRS, IRISA, Rennes, France*
[2] *Institut Pasteur, CNRS, Paris, France*
***Corresponding author**: teo.lemane@inria.fr*

Abstract

In the context of multiple sequencing samples analyses, one way to represent the sequence content across samples is to build an abundance k-mer matrix. This holistic representation can help for several reference-free biological analyses like read samples similarity computation [1] or RNA-Seq analysis [2]. Basically, it corresponds to a matrix with k-mers in rows and samples in columns where each cell is the abundance of a k-mer in a sample. Its construction for large collections is difficult in terms of computing resources and therefore requires appropriate methods and tools.

We present an update to `kmtricks` [3], a flexible tool that allows to efficiently build abundance k-mer matrices and Bloom filters. It extends state-of-the-art k-mer counting methods to external joint multi-samples counting and provides various utilities for downstream analysis: 1) Command-line construction tools and pipeline. 2) A C++ API to e.g. stream the matrix in parallel. 3) A C++ plugin support to customize matrix filtering.

At the last SeqBim edition we presented an earlier version of `kmtricks`, which has now been greatly improved and applied to a very large metagenomic dataset from the Tara Ocean Project. We will also present an ongoing work: `kmdiff`, a tool for structural variant calling using k-mer matrices of large case/control cohorts.

References

[1] Gaëtan Benoit, Pierre Peterlongo, Mahendra Mariadassou, Erwan Drezen, Sophie Schbath, Dominique Lavenier, and Claire Lemaitre. Multiple comparative metagenomics using multiset k-mer counting. *PeerJ Computer Science*, 2016(11):e94, nov 2016.

[2] Jérôme Audoux, Nicolas Philippe, Rayan Chikhi, Mikaël Salson, Mélina Gallopin, Marc Gabriel, Jérémy Le Coz, Emilie Drouineau, Thérèse Commes, and Daniel Gautheret. DE-kupl: Exhaustive capture of biological variation in RNA-seq data through k-mer decomposition. *Genome Biology*, 18(1):243, dec 2017.

[3] Téo Lemane, Paul Medvedev, Rayan Chikhi, and Pierre Peterlongo. kmtricks: Efficient construction of Bloom filters for large sequencing data collections. *bioRxiv*, page 2021.02.16.429304, feb 2021.

Abstract

SeqBIM

The advantage of DNA reads overlaps' reverse complement symmetry for their storage in an oriented graph

Victor Epain[1]*, Rumen Andonov[1], Jean-François Gibrat[2], Dominique Lavenier[1]

[1] *Université de Rennes 1, GenScale, Inria, CNRS, IRISA, Rennes F-35000, France*
[2] *Université Paris-Saclay, MaIAGE, INRAe, Jouy-en-Josas F-78350, France*
***Corresponding author**: victor.epain@irisa.fr*

Abstract

We present a concept and formalism for a new data structure permitting to store and to handle in a very efficient way overlaps between DNA sequences from a set of long reads. Overlaps between reads are defined as suffix-to-prefix alignments, and allow to assemble genomes without any reference.

As the two DNA strands are both sequenced in reverse reading, and the obtained reads are randomly sampled from either a strand or its complementary, reads must be considered in both orientations. Let us define the *forward* orientation, corresponding to the original sequence of a read, and the *reverse* one, corresponding to the reverse-complement sequence. Thus, each overlap implies the existence of its reverse. For example, read u in forward orientation (u forward) overlaps v forward if, and only if, v reverse overlaps u reverse.

In order not to duplicate the reads entity by their two possible orientations, E. W. Myers proposed a way to store overlaps in a structure named string graph [1]. However, the graph is not oriented and the way the reads overlap, as well as their relative orientations, are both edges' attributes. Well known long reads assembler CANU [2] still uses the original Myers' idea, and the conventional assembly graph structure too.

Here we present a more compact and explicit oriented graph structure, that takes advantage of the overlaps' reverse symmetry. We show that iterating over either successors or predecessors of an oriented read is more efficient for our graph. It permits us to adapt a basic graph search algorithm using this symmetry to discover inverse repeats.

References

[1] Eugene W. Myers. The fragment assembly string graph. *Bioinformatics*, 21(suppl_2):ii79–ii85, January 2005.

[2] Sergey Koren, Brian P. Walenz, Konstantin Berlin, Jason R. Miller, Nicholas H. Bergman, and Adam M. Phillippy. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Research*, 27(5):722–736, January 2017.

Abstract

SeqBIM

Identical sequences found in distant genomes reveal frequent horizontal transfer across the bacterial domain

Michael Sheinman[1], Ksenia Arkhipova[2], Peter F. Arndt [3], Bas E. Dutilh[2], Rutger Hermsen[2], Florian Massip[4]

[1] *Theoretical Biology and Bioinformatics, Biology Department, Utrecht University, Padualaan 8,3584 CH, Utrecht, The Netherlands*
[2] *Division of Molecular Carcinogenesis, the Netherlands Cancer Institute, Plesmanlaan 121, 1066 CX Amsterdam*
[3] *Max Planck Institute for Molecular Genetics, Ihnestr. 63/73, 14195 Berlin, Germany*
[4] *MINES ParisTech, PSL-Research University, CBIO-Centre for Computational Biology, 75006 Paris, France*

Abstract

Horizontal Gene Transfer (HGT) is an essential force in microbial evolution. Despite detailed studies on a variety of systems, a global picture of HGT in the microbial world is still missing. Here, we exploit that HGT creates long identical DNA sequences in the genomes of distant species, which can be found efficiently using alignment-free methods. Our pairwise analysis of 93 481 bacterial genomes identified 138 273 HGT events. We developed a model to explain their statistical properties as well as estimate the transfer rate between pairs of taxa. This reveals that long-distance HGT is frequent: our results indicate that HGT between species from different phyla has occurred in at least 8% of the species. Finally, our results confirm that the function of sequences strongly impacts their transfer rate, which varies by more than 3 orders of magnitude between different functional categories. Overall, we provide a comprehensive view of HGT, illuminating a fundamental process driving bacterial evolution.

References