RedOak: a reference-free and alignment-free structure for indexing a collection of similar genomes

Clément Agret[*1, 2, 5], **Annie Chateau**[1, 4], **Gaetan Droc**[2], **Gautier Sarah**[3], **Manuel Ruiz**[2, 4], **and Alban Mancheron**[1, 4]

1 LIRMM, Univ Montpellier, CNRS, Montpellier, France **2** Cirad, UMR AGAP, Avenue Agropolis, Montpellier, France **3** INRA, UMR AGAP, 2 Place Pierre Viala, Montpellier, France **4** Institut de Biologie Computationnelle, Montpellier, France **5** CRIStAL, Centre de Recherche en Informatique Signal et Automatique de Lille, Lille, France

DOI: 10.21105/joss.04167

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Submitted: 15 February 2022
Published: 15 February 2022

Summary

As the cost of DNA sequencing decreases, high-throughput sequencing technologies become increasingly accessible to many laboratories. Consequently, new issues emerge that require new algorithms, including tools for indexing and compressing hundred to thousands of complete genomes.

This paper presents RedOak, a reference-free and alignment-free software package that allows for the indexing of a large collection of similar genomes. RedOak can also be applied to reads from unassembled genomes, and it provides a nucleotide sequence query function. Our method is about the analysis of complete genomes from the 3000 rice genomes sequencing project, but our indexing structure is generic enough to be used in similar projects. This software is based on a *k*-mer approach and has been developed to be heavily parallelized and distributed on several nodes of a cluster. The source code of our RedOak algorithm is available at RedOak.

Statement of need

RedOak may be really useful for biologists and bioinformaticians expecting to extract information from large sequence datasets.

The indexation of complete genomes is an important stage in the exploration and understanding of data from living organisms. Complete genomes, or at least a set of sequences representing whole genomes, *i.e.*, draft genomes, are becoming increasingly easy to obtain through the intensive use of high-throughput sequencing. A new genomic era is coming, therein not only being focused on the analyses of specific genes and sequences regulating them but moving toward studies using from ten to several thousands of complete genomes per species. Such a collection is usually called a pan-genome (Computational Pan-Genomics, 2016)(Golicz et al., 2016). Within pan-genomes, large portions of genomes are shared between individuals. This feature could be exploited to reduce the storage cost of the genomes.

Based on this idea, this paper introduces an efficient data structure to index a collection of similar genomes in a reference- and alignment-free manner. A reference-free and alignment-free approach avoids the loss of information about genetic variation not found in the direct mapping of short sequence reads onto a reference genome (Computational Pan-Genomics, 2016). Furthermore, the method presented in this paper can be applied to next-generation

[*] corresponding author

sequencing (NGS) reads of unassembled genomes. The method enables the easy and fast exploration of the presence-absence variation (PAV) of genes among individuals without needing the time-consuming step of *de novo* genome assembly nor the step of mapping to a reference sequence.

Complexity

In this part, we present the time and space complexity of the algorithm, using the notations below:

Total number of distinct kmers: $N = K$

Total number of core kmers: $N^* = K^*$

Total number of shell kmers: $N^+ = K^+$

Total number of cloud kmers: $N^- = K^-$

Number of instances running in parallel: np

Size in bits of a memory word: u

Theorem1. The space needed for indexing n genomes is equal to $2k_2N+N^+(n+u)+O(4^{k_1}n)$ bits.

If k_1 is defined as $k_1 = \frac{\log(N)-\log(\log(N))+O(1)}{2}$,then the memory space required by RedOak to index the k-mers of n genomes is increased by

$N(2, k_2+n)+o(n, N)$ bits.

Theorem 2. The time needed to index the N distinct k-mers of n genomes is $O(nNk)$.

Theorem 3 Assuming that the number of genomes per indexed k-mer follows a Poisson distribution of parameter λ (where λ is the average number of genome sharing a k-mer), the size of N is $O(\frac{nm}{\lambda})$.

Proof Since the run time clearly depends on the number of indexed k-mers, let us use a simple model to approximate the time complexity. Suppose that each genome has m distinct k-mers and that each k-mer has a fixed probability p_i to be shared exactly by i genomes out of n. The total number of indexed k-mers is then

$$N = n\sum_{i=1}^{n}\frac{p_i m}{i} = nm\sum_{i=1}^{n}\frac{p_i}{i}$$

.

Implementation

RedOak is implemented in C/C++ and its construction relies on parallelized data processing. A preliminary step, before indexing genomes, is performing an analysis of the composition in k-mers of the different genomes. During this step, k-mer counting tools could be involved and their performance is crucial in the whole process(Manekar & Sathe, 2018). We looked for a library allowing us to handle a large collection of genomes or reads, zipped or not, working in RAM memory, and providing a sorted output. Indeed, RedOak uses libGkArrays-MPI from private communication, Mancheron et al. which is based on the Gk Arrays library (Nicolas Philippe et al., 2011). The Gk array library and libGkArrays-MPI are available under CeCILL licence (GPL compliant). The libGkArrays-MPI library is highly parallelized with both Open~MPI and OpenMP.

To manipulate *k*-mers, the closest method is Jellyfish (Marcais & Kingsford, 2011). This approach is not based on disk but uses memory and allows the addition of genomes to an existing index. However, we did not use it because in the output, *k*-mers are in "fairly pseudo-random" order and "no guarantee is made about the actual randomness of this order"Documentation of JellyFish.

Value of k and k_1, in most of the *k*-mer based studies, the *k*-mer size varies between 25 (with reference genome) and 40 (without reference genome). The value of this parameter can be statistically estimated as stated in (N. Philippe et al., 2009).

The k_1 prefix length in our experiments has been defined on the basis of analytic considerations presented in (Park et al., 2009) but can be arbitrarily fixed to some value between 10 and 16, which respectively leads to an initial memory allocation from 8MiB to 32GiB, equally split across the running instances of RedOak. Setting a higher value is not necessary; otherwise, it may allocate unused memory.

References

Computational Pan-Genomics, C. (2016). Computational pan-genomics: Status, promises and challenges [Journal Article]. *Brief Bioinform*. https://doi.org/10.1093/bib/bbw089

Golicz, A. A., Batley, J., & Edwards, D. (2016). Towards plant pangenomics [Journal Article]. *Plant Biotechnol J*, *14*(4), 1099–1105. https://doi.org/10.1111/pbi.12499

Manekar, S. C., & Sathe, S. R. (2018). A benchmark study of k-mer counting methods for high-throughput sequencing [Journal Article]. *Gigascience*, *7*(12). https://doi.org/10.1093/gigascience/giy125

Marcais, G., & Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, *27*(6), 764–770. https://doi.org/10.1093/bioinformatics/btr011

Park, G., Hwang, H.-K., Nicodème, P., & Szpankowski, W. (2009). Profiles of Tries. *Siam Journal on Computing*, *38*(5), 1821–1880. https://doi.org/10.1137/070685531

Philippe, N., Boureux, A., Brehelin, L., Tarhio, J., Commes, T., & Rivals, E. (2009). Using reads to annotate the genome: influence of length, background distribution, and sequence errors on prediction capacity. *Nucleic Acids Research*, *37*(15), e104. https://hal.archives-ouvertes.fr/hal-00452377

Philippe, Nicolas, Salson, M., Lecroq, T., Léonard, M., Commes, T., & Rivals, E. (2011). Querying large read collections in main memory: a versatile data structure. *BMC Bioinformatics*, *12*(1), 242. https://doi.org/10.1186/1471-2105-12-242