Vizitig: A visual tool for colored de Bruijn graphs exploration

Bastien Degardins, Margaux Mouton, Bruno Guillon, Charles Paperman, Camille Marchet

Vizitig: A visual tool for colored de Bruijn graphs exploration

Bastien Degardins[1], Margaux Mouton[1], Bruno Guillon[2],
Charles Paperman[1] and Camille Marchet [1*]
[1] UMR9189 CRIStAL, Univ Lille, INRIA, CNRS, Centrale, F-59000 Lille, France
[2]LIMOS - Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
[*]**Corresponding author** camille.marchet@univ-lille.fr

Abstract

Vizitig is a novel visualization tool designed for the exploration of colored de Bruijn graphs, a structure increasingly used for comparative genomics, pangenomics, and metagenomics. Unlike existing tools such as Bandage, Vizitig supports the simultaneous visualization of multiple genomes or samples by leveraging color-coding schemes to highlight shared and unique sequences or meta-data. Vizitig allows users to efficiently manage, navigate, and render graphs annotated with various metadata, such as genomic features or sample-specific information. Developed as a cross-platform Python application, Vizitig integrates a user-friendly web interface and a command-line-free environment, making it accessible to a broader audience of genomic researchers.

1 Introduction

Bandage [Wick et al. (2015)] has been a popular option in genomics for facilitating the visualization and exploration of assembly graphs generated by *de novo* assemblers. Bandage provides an intuitive graphical user interface (GUI) that allows researchers to interact with assembly graphs, offering functionalities like zooming, panning, and manual node manipulation to assess assembly quality.
De Bruijn graph are graph that represents overlaps between genomic substring of size k (k-mers), making it a key structure for solving assembly and more generally to outline the DNA/RNA structure and variants out of a read sample.However, as genomic research increasingly moves towards comparative genomics, pangenomics, and metagenomics, there is a growing need for tools that can handle colored de Bruijn graphs [Marchet (2024); Andreace et al. (2023)]—an extension of traditional de Bruijn graphs where each k-mer node is annotated with one or more "colors" representing different genomes, strains, or samples. Colored de Bruijn graphs enable the simultaneous visualization and comparison of multiple genomic datasets within a single graph structure, facilitating the identification of shared and unique sequences and exploration of genomic diversity [Marchet (2024)].
Vizitig is designed to address the limitations of existing visualization tools like Bandage in the context of colored de Bruijn graphs. While Bandage supports visualizing single-genome assembly graphs, it lacks native support for allowing navigation in meta-data associated to graphs. Vizitig allows to:

- Simultaneously visualize multiple genomes or samples within a single graph through intuitive color-coding schemes (Figure 1 (c)).
- Explore complex relationships between different datasets, including shared sequences and unique variations (Figure 1 (a,b)).
- Efficiently manage and render colored graphs. A key aspect of Vizitig is that it generalized the "color" concept to any meta-data. Users can color their graphs according to features in provided gff/gtf, samples, or any custom filter.

Vizitig extends the toolkit available to genomic researchers, empowering them with visual exploration tools and the possibility to export and share subgraphs of interest.

2 Implementation

Vizitig is developed as a cross-platform application using Python. The software leverages NetworkDisk[1], that allows Vizitig to handle de Bruijn graphs on disk as if it were a NetworkX object. K-mers of the graph are associated to meta-data through the SQLite relational database. Meta-data encompasses presence in a sample, and fields of gtf/gff files. Vizitig benefits from a Python API in addition to the visualization. The API is connected to a web interface that exposes the graphs and their data. This user interface allows the use of all the CLI commands, making it a command-line free tool. This latter aspect is crucial for allowing other scientists to use the tool.

Availability. Vizitig is available at https://gitlab.inria.fr/pydisk/examples/vizitig and can be installed using installed using `pip install vizitig`. The documentation provides the command lines to install the software. It is fully compatible on all Linux distributions.

Performance metrics. Ingesting the human genome currently takes 20 minutes plus 2 hours to index on a 8GHz/6 cores CPU. The index files takes 40 GB on disk, and requires no RAM allocation when used. The query of one million random k-mers (including k-mer that are not in the graph) on this index takes 3 seconds. The querying of metadata is instantaneous in terms of user time (in the order of the second or less). The RAM consumption depends on the size of the manipulated subgraph, but takes ¡30 GB for 5,000 nodes.

3 Case studies

3.1 Find splicing variations in samples in comparison to references

For this case study we selected several samples from the Cancer Cell Line Encyclopedia (CCLE), associated with aberrant transcripts (SRR_8615240, SRR_8615242, SRR_8616107) in DepMap[2]. We built a de Bruijn graph with $k = 61$ for this sample, By manually exploring the CIC gene, we found patterns showing an alternative splicing (marked in Figure 1 (a), blue: reference transcript for CIC (NM_001304815 from RefSeq HG38), orange: signal existing the SRR_8615240 dataset but not in annotated sequences of CIC). Vizitig can display the meta-data associated to the signal showed in the graph, corresponding to an un-annotated alternatively spliced exon in sample SRR_8615240 and SRR_8616107. Using Vizitig's table view, we can extract the sequence corresponding to that event, and we used blat on the human genome (online USCS's blat version[3]) to generate an alternative view of the event (Supplementary Figure S2). Using Reindeer on the `Transipedia.fr`[4] platform [Bessière et al. (2024)], we confirmed that the genomic signature of the alternative spliced exon exists and is expressed in SRR_8615240 and SRR_8616107 along with other cancer-related samples. (see Supplementary Figure S2).

[1] https://networkdisk.inria.fr/
[2] https://depmap.org/portal/home/
[3] https://genome.ucsc.edu/cgi-bin/hgBlat
[4] https://transipedia.fr/

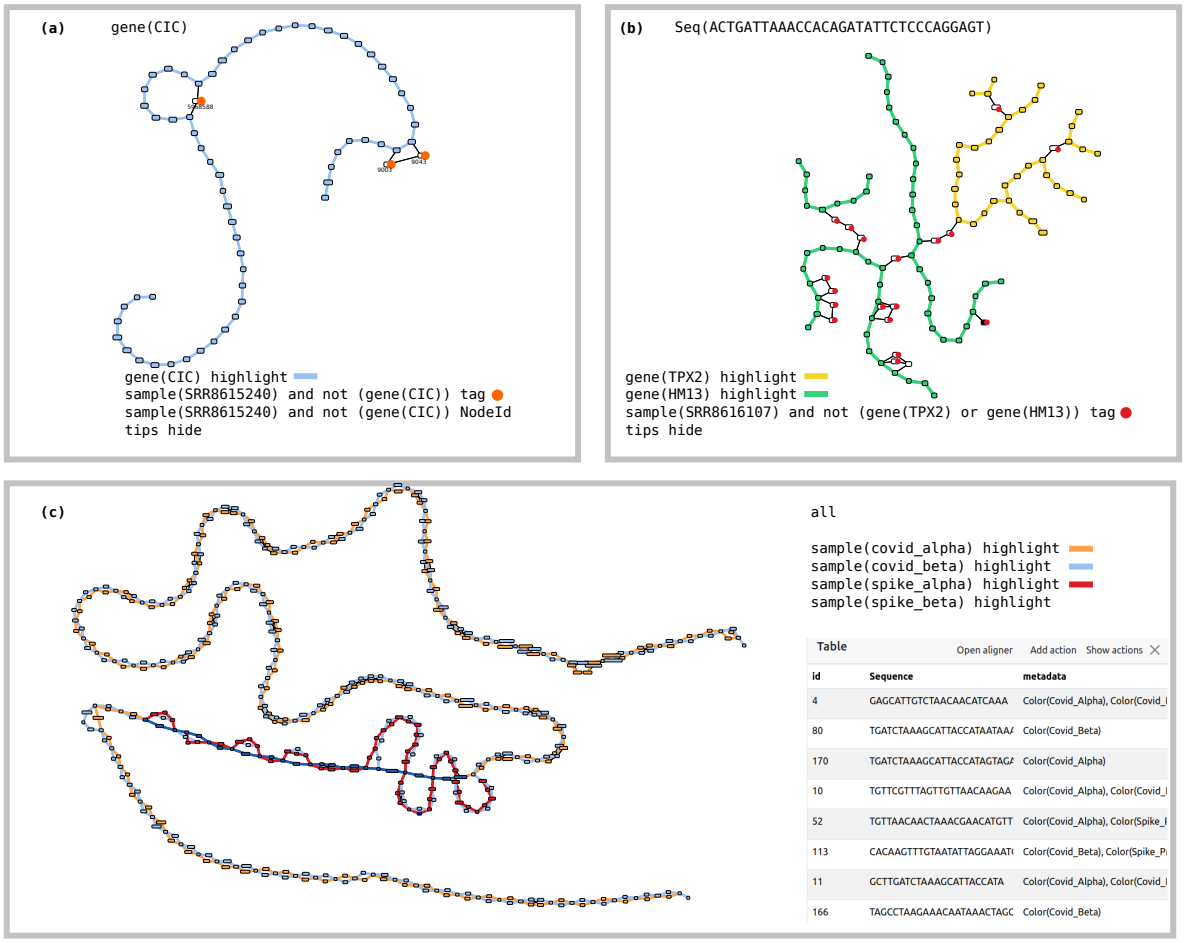

Figure 1: For transcriptomics (a,b): all presented graphs were generated by ingested de Bruijn graphs built from selected samples and annotated with Refseq transcripts. All subgraphs were generated by queries to Vizitig (command line at the top of each panel) and colored using filters (command lines at the bottom of each panel), and exported directly by Vizitig in svg format. (a) Un-annotated splicing seen in the CIC gene for a cancer-related sample from the CCLE. (b) Signal of a fusion transcripts, two nodes marked in red make the junction between genes TPX2 and HM13, seen in a fusion sample of the CCLE. (c) A pangenome built from two SARS-COV-2 genomes and their associated Spike protein sequence, accounting for the alpha and beta variants. The nodes for the spike protein of each have been colored differently, and shower a greater amount of variations.

3.2 Extract the subgraph associated to a fusion transcript

As in the previous case, we used the CCLE to find datasets with fusion candidates (SRR_8616107 referenced by DepMap) and validated k-mers supporting the fusion junction using the Transipedia.fr platform. This time, we started with a subsequence probe, marker of the fusion junction, that we queried to the graph. We then extended the subgraph around this junction (Figure 1 (b)), and colored regions corresponding to TPX2 in yellow, and HM13 in green. Note that a subsequence smaller than the k-mer size used to build the graph (here k=61, see the queried subsequence at the top of panel (b)) can be queried, mitigating a weakness in ad-hoc colored graph structure that usually request k-mers or longer queries. A lookup on Transipedia confirms that this event is expressed and seems exclusively found in SRR_8616107 in the CCLE collection (see Figure S3 in Appendix).

3.3 Display a SARS-COV-2 pangenome

For this example, we selected two SARS-COV-2 genomes and annotated spike protein regions that we co-assembled (NCBI accessions). The two genomes have been colored in yellow and light blue, and spikes in red and dark blue in Figure Figure 1 (c). Figure 1 (c) also shows the table view displayed by Vizitig, where the sequences and meta-data corresponding to the nodes can be accessed.

4 Conclusion

Vizitig offers a powerful and intuitive solution for exploring colored de Bruijn graphs, enabling the simultaneous visualization and comparison of multiple genomic datasets with integrated metadata. Its flexibility, combined with an accessible user interface, extends our capabilities in analyzing complex relationships and variations across samples or genomes.

Fundings

This study has been supported by ANR JCJC Find-RNA [ANR-23-CE45-0003].

References

Francesco Andreace, Pierre Lechat, Yoann Dufresne, and Rayan Chikhi. Comparing methods for constructing and representing human pangenome graphs. *Genome Biology*, 24(1):274, 2023.

Chloé Bessière, Haoliang Xue, Benoit Guibert, Anthony Boureux, Florence Rufflé, Julien Viot, Rayan Chikhi, Mikaël Salson, Camille Marchet, Thérèse Commes, et al. Transipedia. org: k-mer-based exploration of large rna sequencing datasets and application to cancer data. *Genome Biology*, 25(1):266, 2024.

Camille Marchet. Advances in colored k-mer sets: essentials for the curious. *arXiv preprint arXiv:2409.05214*, 2024.

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.

Appendix

Validation of the case studies

Aberrant splicing transcript

The sequence passed to UCSC's blat was `GGGCGGCAGTGGGTAAAGGAGGAACGGATCACAGGTGAAAACACCTTCCGAC CAAAGCCCAATGACATCATCATCCCCTTCCTCCTCACAGATGACCATGCGCTCCTCATCACTCGTCA`, found in the meta-data of the orange node in the panel (a) of Figure 1). Figure S3 (a) shows that the spliced out exon signature is present in multiple datasets of the CCLE. We also mapped the sequence to the human genome using UCSC's blat, that reveals the un-annotated event as well (Figure S3 (b)).

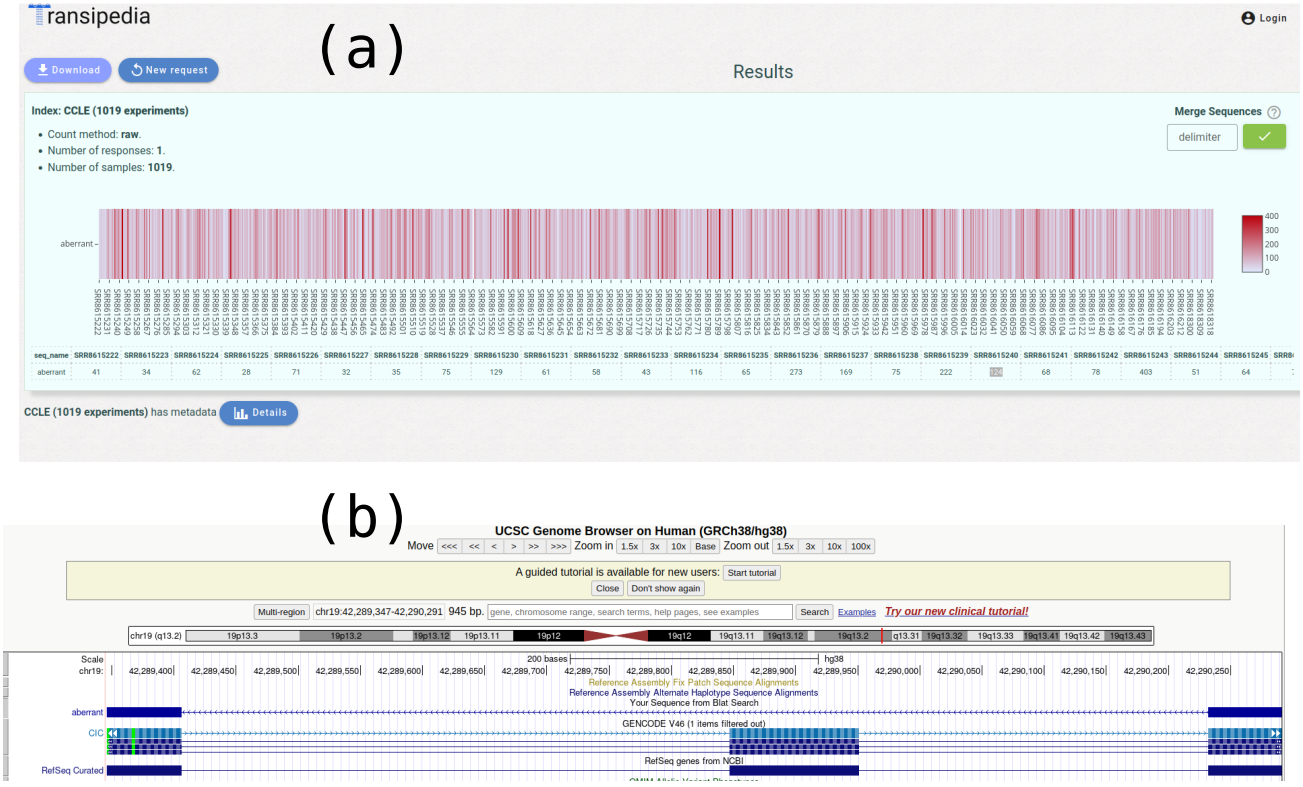

Figure S2: (a) Lookup of the aberrant splicing signature in the CCLE dataset using Transipedia.fr The fusion appears with multiple datasets. (b) Mapping of the aberrant event on the human genome using UCSC's blat (top track), showing an un-annotated spliced exon (references in lower tracks), accordingly to our graph finding.

Fusion transcript

We looked-up in Transipedia the same sequence that the one used to query Vizitig in panel (b) of Figure 1. The result is reported in Figure S3 panel (a). The Figure also reports in panel (b) our attempt to map the sequence on the human genome Hg38 with UCSC's blat. Two different parts map as expected on distant loci.

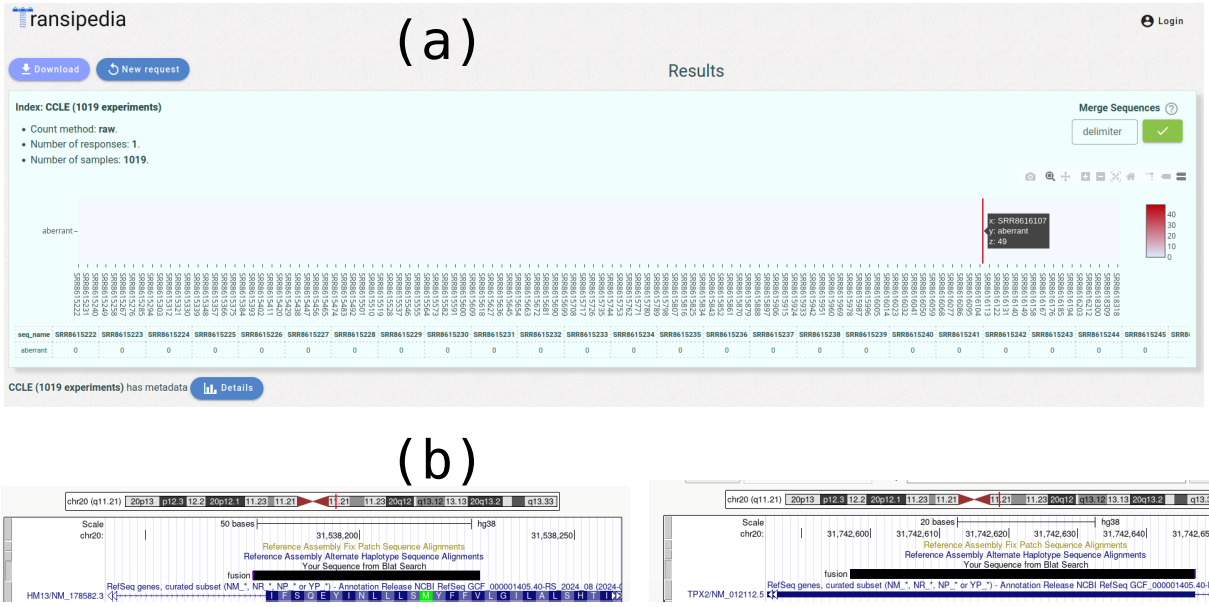

Figure S3: (a) Lookup of the fusion signature in the CCLE dataset using Transipedia.fr The fusion appears with an abundance of 49 (raw) and 6.5 (normalized) for a single dataset, SRR_8616107. (b) Mapping of the fusion event on the human genome using UCSC's blat.