BMC Genomics

SOFTWARE

Open Access

OpenGenomeBrowser: a versatile, dataset-independent and scalable web platform for genome data management and comparative genomics

Thomas Roder[1], Simone Oberhänsli[1], Noam Shani[2] and Rémy Bruggmann[1*]

Abstract

Background: As the amount of genomic data continues to grow, there is an increasing need for systematic ways to organize, explore, compare, analyze and share this data. Despite this, there is a lack of suitable platforms to meet this need.

Results: OpenGenomeBrowser is a self-hostable, open-source platform to manage access to genomic data and drastically simplifying comparative genomics analyses. It enables users to interactively generate phylogenetic trees, compare gene loci, browse biochemical pathways, perform gene trait matching, create dot plots, execute BLAST searches, and access the data. It features a flexible user management system, and its modular folder structure enables the organization of genomic data and metadata, and to automate analyses. We tested OpenGenomeBrowser with bacterial, archaeal and yeast genomes. We provide a docker container to make installation and hosting simple. The source code, documentation, tutorials for OpenGenomeBrowser are available at opengenomebrowser.github.io and a demo server is freely accessible at opengenomebrowser.bioinformatics.unibe.ch.

Conclusions: To our knowledge, OpenGenomeBrowser is the first self-hostable, database-independent comparative genome browser. It drastically simplifies commonly used bioinformatics workflows and enables convenient as well as fast data exploration.

Keywords: Genome database, Genome browser, Comparative genomics, Open-source, Self-hosted

Background

Driven by advances in sequencing technologies, many organizations and research groups have accumulated large amounts of genomic data. As sequencing projects progress, the organization of such genomic datasets becomes increasingly difficult. Systematic ways of storing data and metadata, tracking and denoting changes in assemblies or annotations, and enabling easy access are key challenges. While standardized data formats and free software are widely used in the field to process genomic data, data exploration is often still cumbersome. This is especially true for non-bioinformaticians, although numerous platforms have been developed to simplify data access.

Most of these platforms have different user interfaces and sometimes limited functionality. The reason for this heterogeneity is that most of them have been developed independently, i.e., each one for a specific genomic dataset. Such platforms exist for many well-studied organisms, such as *Pseudomonas* spp. [1], but also for

*Correspondence: remy.bruggmann@bioinformatics.unibe.ch

[1] Interfaculty Bioinformatics Unit and Swiss Institute of Bioinformatics, University of Bern, 3012 Bern, Switzerland
Full list of author information is available at the end of the article

non-model species such as ginseng [2] and cork oak [3]. These platforms share a set of core features: access to data, sequence similarity searches (like BLAST [4]), and limited annotation searches. The most advanced of these platforms, such as CoGe [5], MicrobesOnline [6], WormBase [7], Genomicus [8], MicroScope [9] and ChlamDB [10], include additional functions to answer a wide range of questions.

However, these platforms tend to be tied to the characteristics of a specific dataset and adapting their software to other projects would be extremely difficult. This is surprising given that the underlying data are essentially the same: genome assemblies, genes, proteins, and their annotations. Fortunately, this information is stored in standardized data formats across many fields, which in principle would allow code reuse and collaborative development. Even while some degree of purpose-built software tools may still be necessary for certain projects, independent development comes at a significant initial cost as well as a long-term maintenance cost and a higher risk of becoming outdated.

We addressed these issues by developing OpenGenomeBrowser, a self-hostable, open-source software based on the Python web framework Django [11]. OpenGenomeBrowser runs on all modern browser engines (Firefox, Chrome, Safari). It contains more features than most similar platforms, is highly user-friendly and *dataset-independent* – i.e., not bound to any specific genomic dataset. A comparison of OpenGenomeBrowser and similar platforms is available in Table S1.

Implementation

To enable automated processing of genomic data, as in OpenGenomeBrowser, it is essential that the data is stored in a systematic fashion. We present our solution to this problem in detail in the section "*folder structure*". The subsequent section "*OpenGenomeBrowser tools*" describes a set of scripts that simplify the handling of the aforementioned folder structure.

Folder structure

Every sequencing project faces an important challenge: systematic storage of data and metadata according to the FAIR principles [12]. These principles enable reproducibility, automation, data interoperability and sharing. Especially in long-term projects, it is crucial to know when and how the data was generated, and to have a transparent way of handling different genome and annotation versions. Different versions are the result of organism re-sequencing, raw data re-assembly or assembly re-annotation. Importantly, each version of a gene must have a unique identifier, and legacy data should be kept instead of being overwritten.

To address these problems, we developed a modular folder structure (Fig. 1A). The *organisms* folder contains a directory for each biological entity, e.g., a bacterial strain. Each of these folders must contain a metadata file, *organism.json* (Fig. 1A, center), describing the biological entity, and a folder named *genomes*. The *genomes* folder contains one folder for each genome version. One of these genomes must be designated as the *representative* genome of the biological entity in *organism.json*. This allows project maintainers to update an assembly transparently, by designating the new version as *representative* without removing the old one.

Each genome folder must contain a metadata file, *genome.json* (Fig. 1A), and the actual data: an assembly FASTA file, a GenBank file, and a gff3 (general feature format version 3) file. While not strictly required but strongly recommended, annotation files in tab-separated format which map gene identifiers to annotations, may be provided. OpenGenomeBrowser supports several annotation types by default, such as Enzyme Commission numbers, KEGG [13] genes and KEGG reactions, Gene Ontology terms [14, 15], and annotations from EggNOG [16]. Additional annotation types can be easily configured. Files that map annotations to descriptions (e.g., EC:1.1.1.1 ⟶ alcohol dehydrogenase) can be added to a designated folder.

OpenGenomeBrowser tools

A set of scripts called *OpenGenomeBrowser Tools* simplifies the creation of the previously described folder structure and the incorporation of new genomes. As shown below, a functional folder structure that contains one genome can be set up with only four commands.

```
#!/bin/bash
# Install OpenGenomeBrowser Tools (requires Python 3.10+)
pip install opengenomebrowser-tools
# Set desired location of the folder structure
export FOLDER_STRUCTURE=/path/to/folder_structure
# Create a bare-bone folder structure
# Download annotation descriptions for default annotation types
init_folder_structure
# Add a genome to the folder structure. The import-dir must at least contain:
# - an assembly FASTA (.fna)
# - a GenBank file (.gbk)
# - a general feature format file (.gff)
# The output directories of Prokka [17] and PGAP [18] are directly compatible.
import_genome –import-dir=/path/to/genomic/files
```

Software architecture

OpenGenomeBrowser itself is distributed as a Docker container [19]. Using Docker Compose, the container is combined with a database and a webserver to create a production-ready software stack (Fig. 1B).

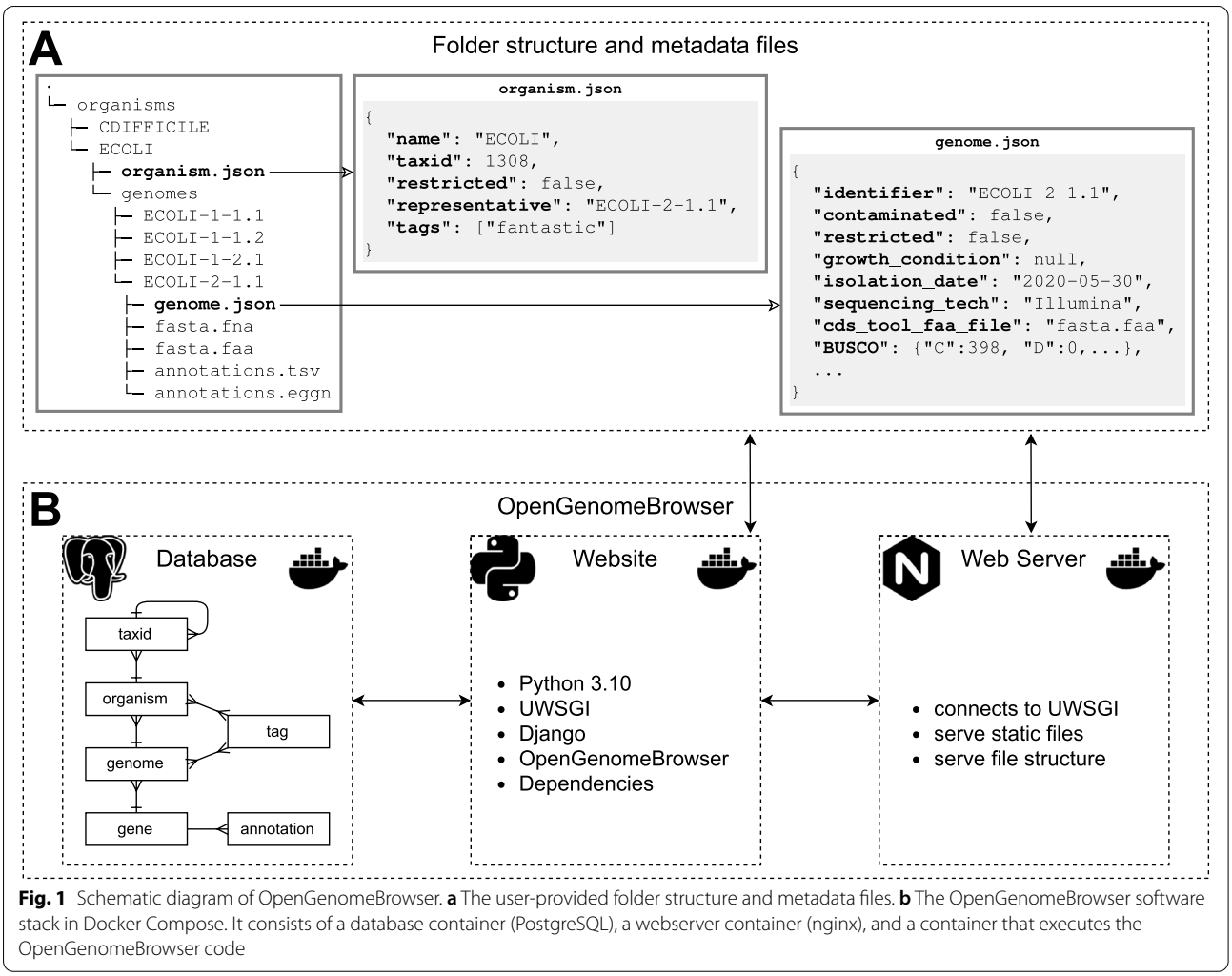

Fig. 1 Schematic diagram of OpenGenomeBrowser. **a** The user-provided folder structure and metadata files. **b** The OpenGenomeBrowser software stack in Docker Compose. It consists of a database container (PostgreSQL), a webserver container (nginx), and a container that executes the OpenGenomeBrowser code

Results and discussion

The following section describes the main features of OpenGenomeBrowser. The reader may try them out at opengenomebrowser.bioinformatics.unibe.ch, where a freely accessible demo server with 70 bacterial genomes is hosted. Notably, on most pages, users may click on *Tools*, then *Get help with this page* to be redirected to a site that explains how the tool works and how to use it. Moreover, advanced configuration options are available on some pages. They can be accessed via a sidebar that opens when one clicks on the settings wheel (⚙) at the top right corner of the page.

Genomes table

Especially in large sequencing projects, it is vital that the data can be filtered and sorted according to metadata. This is the purpose of the *genomes table view* (Fig. 2) which serves as the entry point of OpenGenomeBrowser. By default, only the *representative* genomes are listed and only the name of the organism, the genome identifier, the taxonomic name, and the sequencing technology are shown as columns. Furthermore, there are over forty additional metadata columns available that can be dynamically added to the table. All columns can be used to filter and sort the data, which makes this view the ideal entry point for an analysis.

Detail views

The *genome detail view* (Fig. S1A) shows all available metadata of the respective genome and allows the user to download the associated files.

The *gene detail view* (Fig. S1B) is designed to facilitate easy interpretation of the putative functions of genes. It shows all annotations, their descriptions, the nucleotide- and protein sequences, metadata from the GenBank file and an interactive gene locus visualization facilitated by DNA features viewer [20]. If the gene is annotated with a gene ontology term that represents a subcellular location, this location will be highlighted on a SwissBioPics image [21].

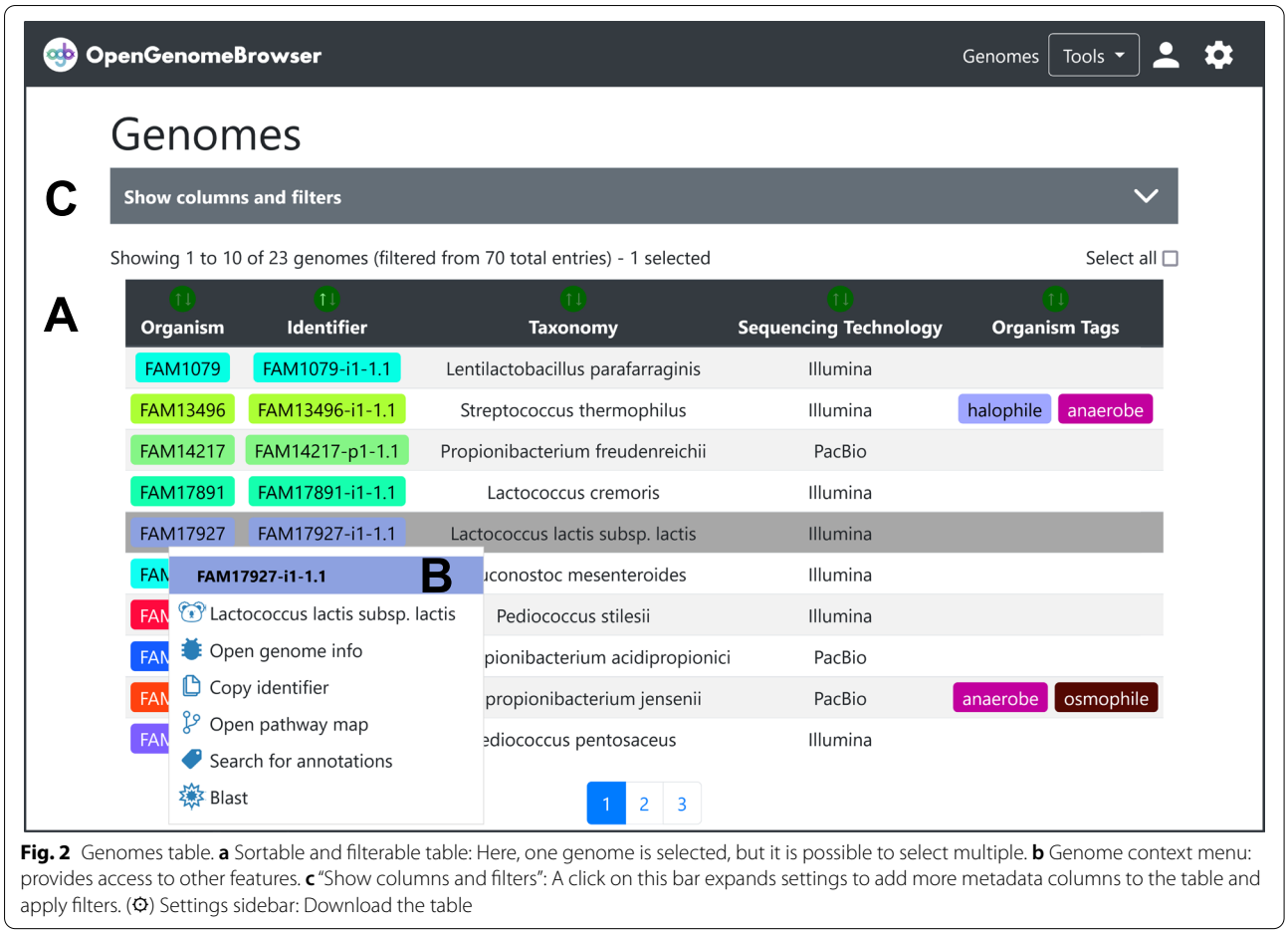

Fig. 2 Genomes table. **a** Sortable and filterable table: Here, one genome is selected, but it is possible to select multiple. **b** Genome context menu: provides access to other features. **c** "Show columns and filters": A click on this bar expands settings to add more metadata columns to the table and apply filters. (⚙) Settings sidebar: Download the table

Genomes in OpenGenomeBrowser can be labelled with tags, i.e., a short name (e.g., "*halophile*") and a description (e.g., "*extremophiles that thrive in high salt concentrations*"). The *tag detail view* (Fig. S1C) shows the description of the tag and the genomes that are associated with it. Tags are particularly useful to quickly select groups of genomes in many tools of OpenGenomeBrowser. For example, to select all genomes with the tag "*halophile*", the syntax "*@tag:halophile*" can be used.

Similarly, the *TaxId detail view* (Fig. S1D) shows all genomes that belong to the respective NCBI Taxonomy identifier (TaxId) [22], as well as the parent TaxId. Similar to tags, TaxIds can be used to select all genomes that belong to a certain TaxId, like this: "*@taxphylum:Firmicutes*", or simply "*@tax:Firmicutes*".

Gene comparison

The *gene comparison view* (Fig. 3) enables users to easily compute multiple sequence alignments and to compare gene loci side-by-side. Currently, Clustal Omega [23], MAFFT [24] and MUSCLE [25] are supported alignment algorithms. Alignments are visualized using MSAViewer [26] (Fig. 3B). Furthermore, the genomic regions around the genes of interest can be analyzed using a customized implementation of DNA features viewer [20] (Fig. 3C). Figure 3 shows an alignment of all genes on the demo server that contain the annotation *K01610* (phosphoenolpyruvate carboxykinase; from the pyruvate metabolism pathway). The gene loci comparison reveals that in all queried *Lacticaseibacilli*, the genes are located in syntenic regions, i.e., next to the same orthologous genes.

Annotation search

Despite conceptually and technically straightforward, searching for annotations in a set of genomes can be tedious or even impossible for non-programmers. In OpenGenomeBrowser, annotation search is quick and easy, thanks to the PostgreSQL backend that allows fast processing of annotation information. In the *annotation search view* (Fig. 4), users can search for annotations in genomes, resulting in a *coverage matrix* (Fig. 4C) with one column per genome and one row per annotation. The numbers in the cells show how many genes in the genome have the same annotation. Clicking on these cells shows the relevant genes (Fig. 4D), while

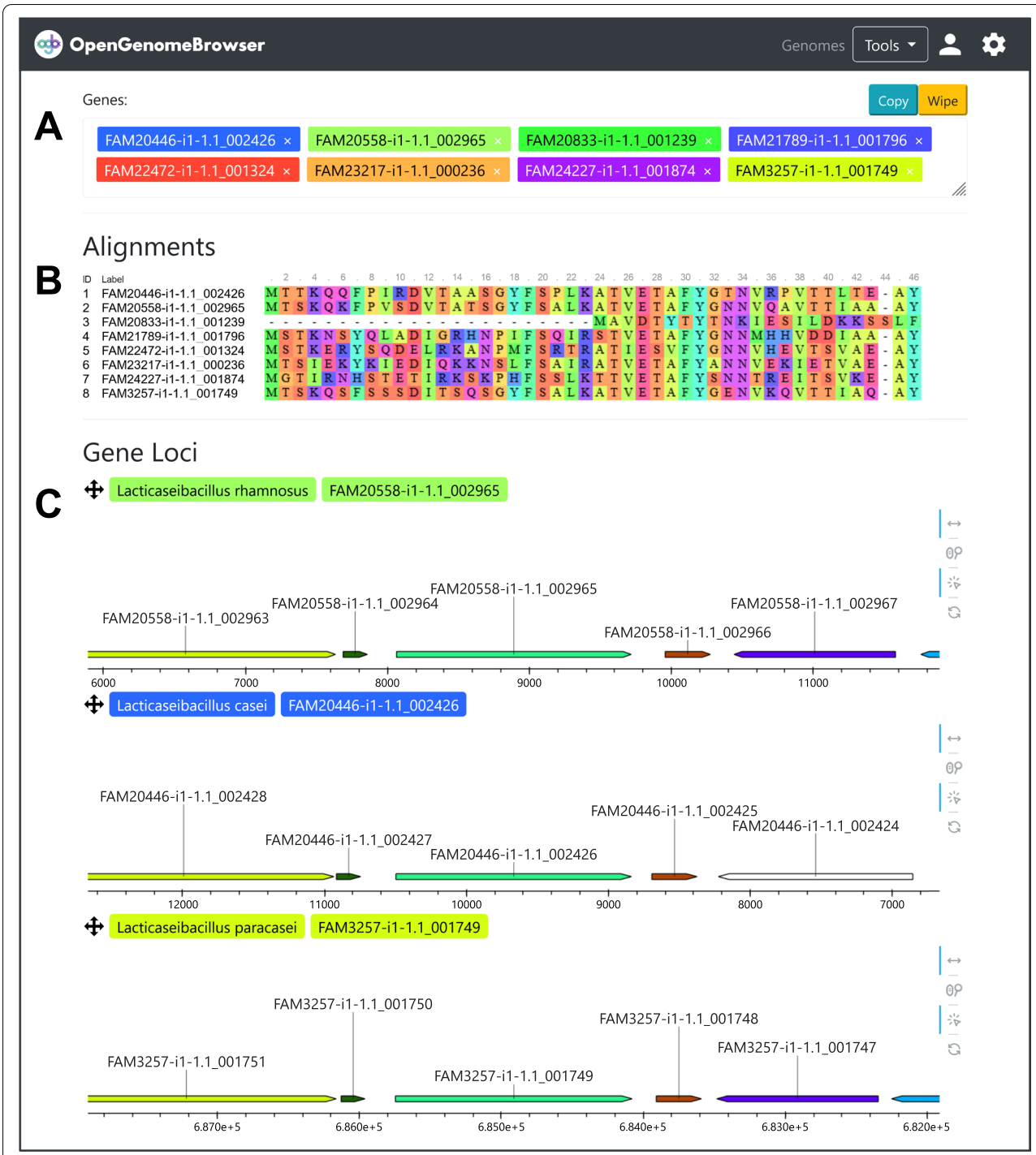

Fig. 3 Gene comparison. **a** Input mask for genes to be searched. **b** Output: alignments. Can be exported in aligned FASTA format. **c** Output: gene loci. Each subplot shows the genes around one of the queried genes, which are represented as colorful arrows. Orthologous genes have the same colors while genes without orthologs are white. The plots are interactive: pan, zoom, and click on genes. (⚙) Settings sidebar. For alignments: Choose between DNA and the protein sequence alignments, change the alignment method. For gene loci: Set the range around the gene's locus, change the annotation category by which to color the genes

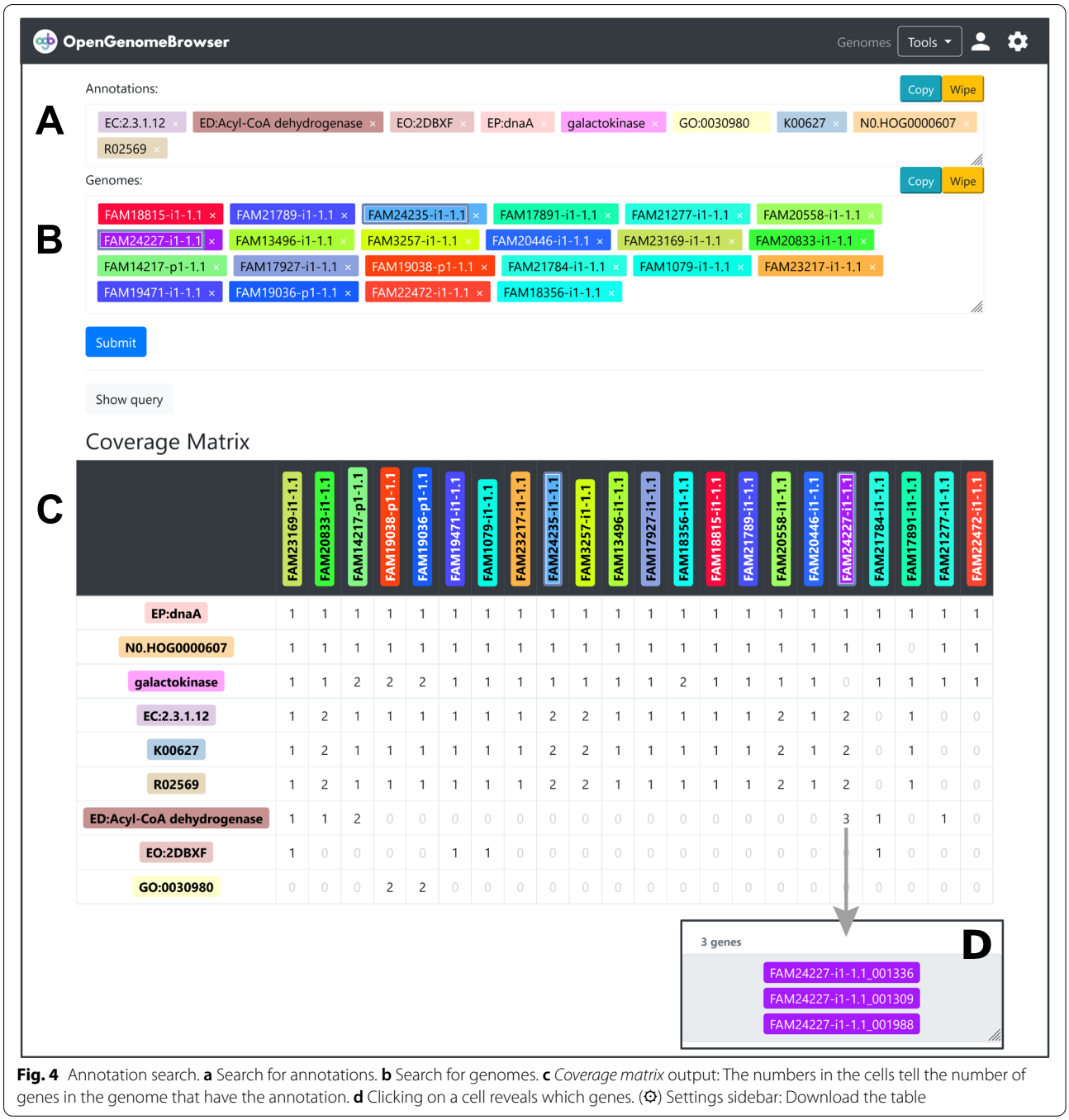

Fig. 4 Annotation search. **a** Search for annotations. **b** Search for genomes. **c** *Coverage matrix* output: The numbers in the cells tell the number of genes in the genome that have the annotation. **d** Clicking on a cell reveals which genes. (⚙) Settings sidebar: Download the table

clicking on an annotation enables users to compare the corresponding genes (*gene comparison view).*

Pathways

Pathway maps, particularly the ones from the KEGG [27], are valuable tools to understand the metabolism of an organism. However, using them may be cumbersome. Commonly, biologists upload sequences to a service like BlastKOALA [28]. This service is designed to process one organism at a time, and calculation times can last multiple hours. Because each genome must be submitted individually, it becomes cumbersome when multiple organisms must be processed. Furthermore, it is not trivial to visualize multiple genomes on a pathway map. In OpenGenomeBrowser, this process is straightforward (Fig. 5A-C), user-friendly, and fast, as the annotations are pre-calculated and loaded into the database beforehand. Pathway maps are interactive, which allows

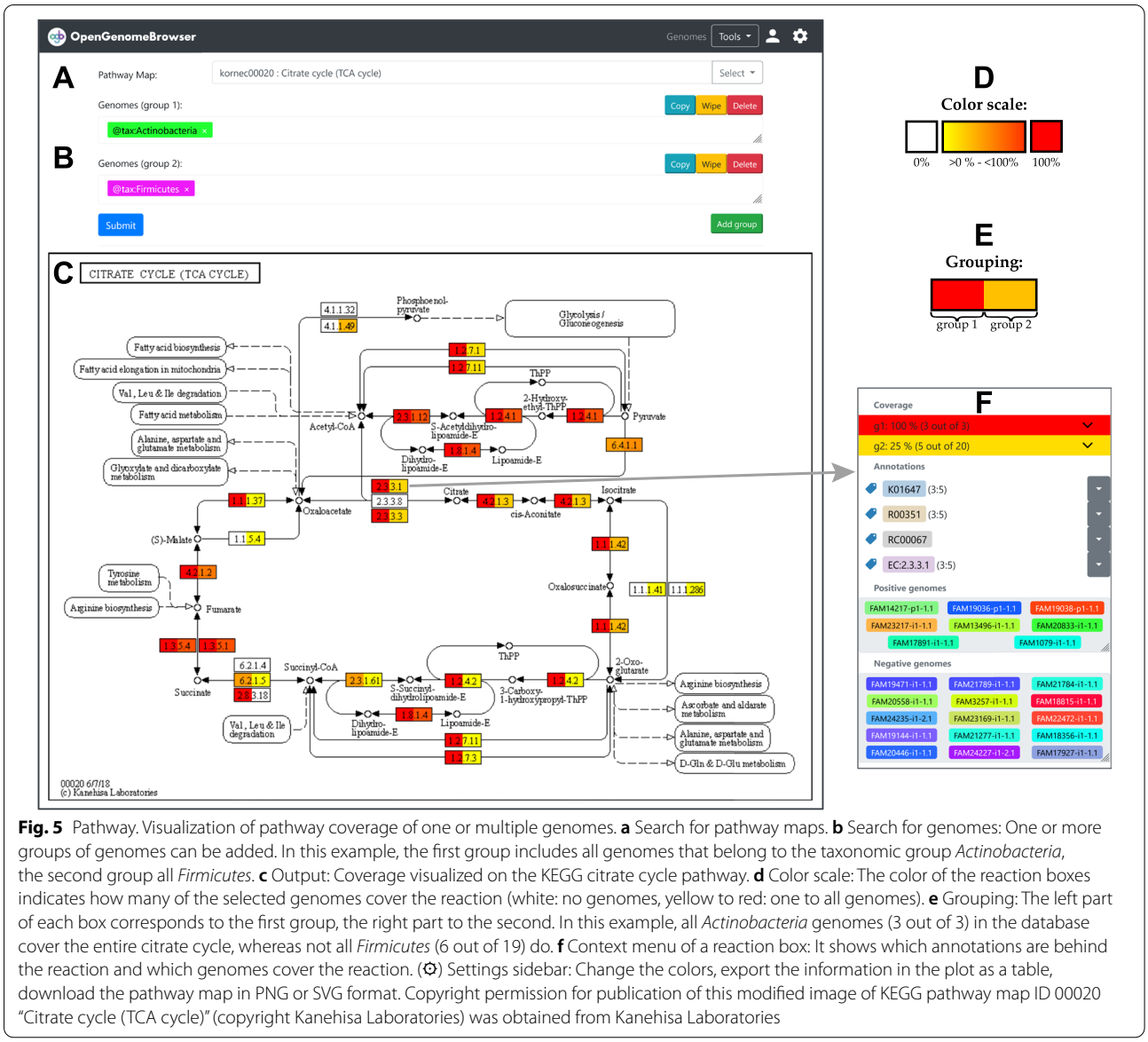

Fig. 5 Pathway. Visualization of pathway coverage of one or multiple genomes. **a** Search for pathway maps. **b** Search for genomes: One or more groups of genomes can be added. In this example, the first group includes all genomes that belong to the taxonomic group *Actinobacteria*, the second group all *Firmicutes*. **c** Output: Coverage visualized on the KEGG citrate cycle pathway. **d** Color scale: The color of the reaction boxes indicates how many of the selected genomes cover the reaction (white: no genomes, yellow to red: one to all genomes). **e** Grouping: The left part of each box corresponds to the first group, the right part to the second. In this example, all *Actinobacteria* genomes (3 out of 3) in the database cover the entire citrate cycle, whereas not all *Firmicutes* (6 out of 19) do. **f** Context menu of a reaction box: It shows which annotations are behind the reaction and which genomes cover the reaction. (⚙) Settings sidebar: Change the colors, export the information in the plot as a table, download the pathway map in PNG or SVG format. Copyright permission for publication of this modified image of KEGG pathway map ID 00020 "Citrate cycle (TCA cycle)" (copyright Kanehisa Laboratories) was obtained from Kanehisa Laboratories

the user to explore this information in great detail (Fig. 5D-F). For example, to investigate the genes that are involved in a certain enzymatic step, one needs only to click on the enzyme box, then on an annotation of interest, and finally on "compare the genes" to be redirected to *gene comparison view*.

While OpenGenomeBrowser does not include KEGG maps for licensing reasons, users with appropriate rights can generate them using a separate program [29]. The pathway maps do not necessarily have to be from KEGG. Pathway maps in a custom Scalable Vector Graphics (SVG) may be added to a designated folder in the folder structure (not shown in Fig. 1).

Blast

OpenGenomeBrowser allows users to perform a local alignment of protein and nucleotide sequences using BLAST [4]. The results are visualized using the BlasterJS [30] library.

Trees

OpenGenomeBrowser computes three kinds of phylogenetic trees. The fastest type of tree is based on the NCBI taxonomy ID which is registered in the metadata. It is helpful to get a quick taxonomic overview, but it entirely depends on the accuracy of the metadata.

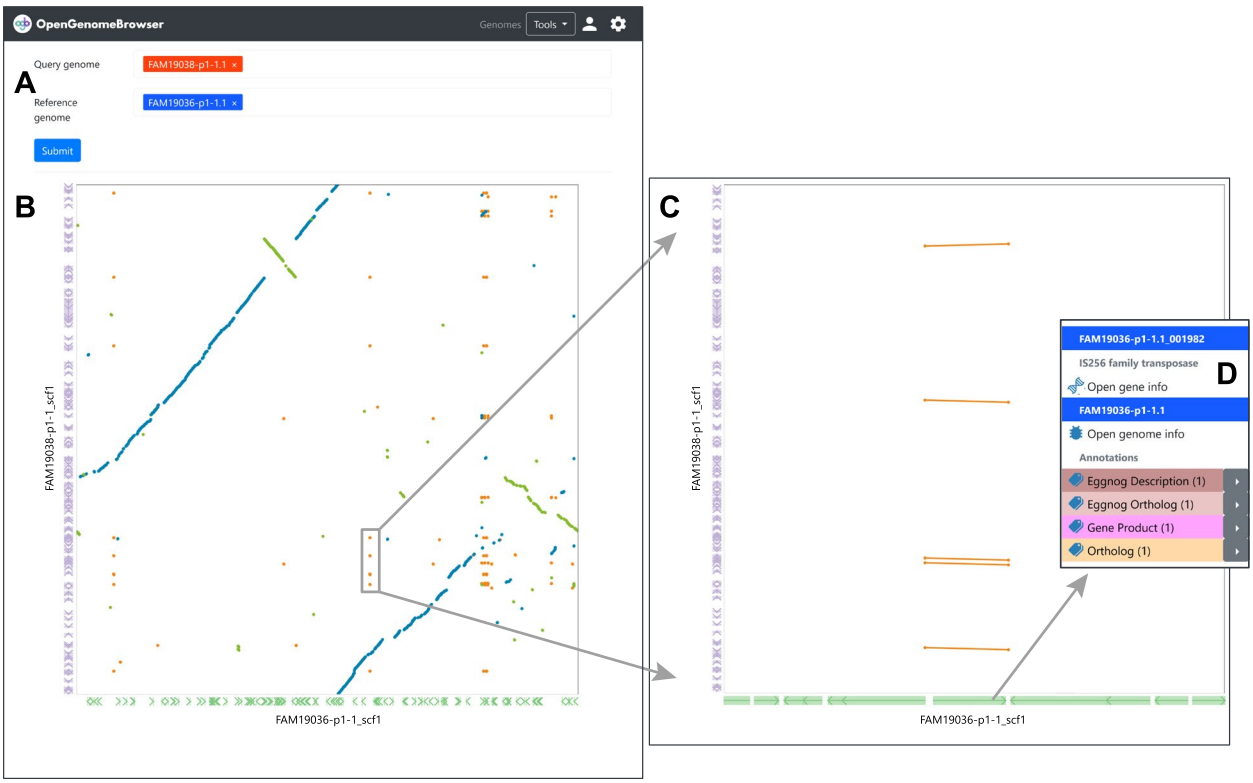

Fig. 6 Dot plot. **a** Search for query and reference genomes. **b** Output: dot plot. The reference genome is on the X-axis, the query genome on the Y-axis. The genes are shown at the edge of the plot in green and violet, respectively. By default, unique forward alignments are colored in blue, unique reverse alignments in green and repetitive alignments in orange. **c** Zooming in is achieved by drawing a rectangle with the mouse over a region with repeated elements. **d** A click on the suspicious gene reveals it to be a transposase. (⚙) Settings sidebar: Configure how the plot is rendered

The second type of tree is based on genome similarity. The assemblies of the selected genomes are compared to each other using GenDisCal-PaSiT6, a fast, hexanucleotide-frequency-based algorithm with similar accuracy as average nucleotide identity (ANI) based methods [31]. This algorithm yields a similarity matrix from which a dendrogram is calculated with the unweighted pair group method with arithmetic mean (UPGMA) algorithm [32]. We recommend this type of tree as a good compromise between speed and accuracy, specifically if many genomes are to be compared.

The third type of tree is based on the alignment of single-copy orthologous genes. This type of tree is calculated using the OrthoFinder [33] algorithm. Of all proposed tree type algorithms it is the most time- and computation-intensive and requires pre-computed all-vs-all DIAMOND [34] searches.

Dot plot

Dot plot is a simple and established [35] method of comparing two genome assemblies. It allows the discovery of insertions, deletions, and duplications, especially in closely related genomes sequenced with long-read technologies. In OpenGenomeBrowser's implementation of dot plot, the assemblies are aligned against each other using MUMmer [36] and visualized using the *Dot* library [37]. The resulting plot (Fig. 6) is interactive, i.e., the user can zoom in on regions of interest by drawing a rectangle with the mouse and clicking on a gene which then opens the context menu with detailed information.

Gene trait matching

The *gene trait matching view* enables users to find annotations that correlate with a (binary) phenotypic trait. The input must consist of two non-intersecting sets of organisms that differ in a trait. OpenGenomeBrowser applies a Fisher's exact test for each orthologous gene and corrects for multiple testing (alpha = 10%) using the Benjamini-Hochberg method [38, 39]. The multiple testing parameters can be adjusted in the settings sidebar. The test can be used on orthogenes as well as any other type of annotation, such as KEGG-gene annotation. The gene candidates that may be causing the trait can easily be

further analyzed, for example by using the *compare genes view*.

Flower plot

The *flower plot view* provides the users with a simple overview of the shared genomic content of multiple genomes. The genomes are displayed as petals of a flower. Each petal indicates the number of annotations that are unique to this genome and the number of genes that are shared by some but not all others. The number of genes shared by all genomes is indicated in the center of the flower. (The code is also available as a standalone Python package [40]).

Downloader

The *downloader view* facilitates the convenient download of multiple raw data files, for example all protein FASTA files for a set of organisms.

Admin panel

OpenGenomeBrowser has a powerful user authentication system and admin interface, inherited from the Django framework. Instances of OpenGenomeBrowser can be configured to require a login or to allow basic access to anonymous users. Users can be given specific permissions, for example to create other user accounts, to edit metadata of organisms, genomes, and tags, and even to upload new genomes through the browser.

Resource requirements

OpenGenomeBrowser is not resource intensive. An instance containing over 1400 bacterial genomes runs on a computer with 8 CPU-cores (2.4 GHz) and 20 GB of RAM. The Docker container is about 3 GB in size and the Postgres database takes 21 GB of storage (SSD recommended).

Conclusions

OpenGenomeBrowser is, to our knowledge, the first comparative genome browser that is not tied to a specific dataset. It automates commonly used bioinformatics workflows, enabling convenient and fast data exploration, particularly for non-bioinformaticians, in an intuitive and user-friendly way.

The software has minimal hardware requirements and is easy to install, host, and update. OpenGenomeBrowser's folder structure enforces systematic yet flexible storage of genomic data, including associated metadata. This folder structure (i) enables automation of analyses, (ii) guides users to maintain their data in a coherent and structured way, and (iii) provides version tracking, a precondition for reproducible research.

OpenGenomeBrowser is flexible and scalable. It can run on a local machine or on a public server, access may be open for anyone or restricted to authenticated users. Annotation types can be customized, and ortholog-based features are optional. While the demo server only holds 70 genomes, the performance scales and is still outstanding even when hosting over 1400 microbial genomes [41].

We believe that our software will be useful to a large community since sequencing microbial and other genomes has become a commodity. Therefore, researchers performing new sequencing projects can directly benefit from OpenGenomeBrowser by saving development costs, making their data potentially FAIR, and adapting the browser for their purposes. It could also replace older, custom-made platforms which may be outdated and more difficult to maintain. Because our software is open-source, adaptations of OpenGenomeBrowser and new features will be available for the whole community under the same conditions. The open-source model also allows problems to be identified and quickly fixed by the community, making OpenGenomeBrowser a sustainable platform.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12864-022-09086-3.

Additional file 1: Table S1. Comparison of OpenGenomeBrowser's features with alternative software platforms. Legend: ✓: feature present; ¢: feature present, but with limitations; Ñ: feature absent. Features were inferred to the best of our knowledge.

Additional file 2: Fig. S1. Detail views. (A) Genome detail view: Shows genome-associated metadata. (B) Gene detail view: Displays a gene's annotations, nucleotide- and protein sequence, metadata extracted from the GenBank file, as well as an interactive plot that shows the adjacent genes. (C) Tag detail view: Shows the tag's name, its description and the organisms and genomes that have it. (D) TaxId detail view: Shows the NCBI TaxId, its taxonomic rank, its parent TaxId and the organisms and their genomes that belong to it.

Acknowledgements

We are grateful to Darja Studer for designing the logo, Lars Vögtlin for his advice on containerization, Linda Studer for her advice on the manuscript, to Kimberly Gilbert for proofreading the article, and Pierre Berthier for his support in hosting OpenGenomeBrowser. We thank Emmanuelle Arias-Roth, Remo Schmidt, Cornelia Bär, Ueli von Ah und Guy Vergères (Agroscope) for their support and feedback on this project.

Availability and requirements

Project name: OpenGenomeBrowser.
Project home page: https://opengenomebrowser.github.io/
Operating system(s): Linux (hosting); platform independent (usage).
Programming language: Python, JavaScript.
Other requirements: Docker.
License: GPL-3.
Any restrictions to use by non-academics: GPL-3.

Authors' contributions

TR, SO and RB conceived the project. TR programmed the software. SO, RB and NS contributed conceptually and with feedback to the software. TR and

RB wrote the manuscript. All authors edited, read, and approved the final manuscript.

Funding
This research was funded by Gebert Rüf Stiftung within the program "Microbials", grant number GRS-070/17 and the Canton of Bern to RB. The funding bodies played no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials
The data used to generate the figures in this study are included in the published article Roder et al., 2020 [41] where the GenBank accession numbers are listed in Supplementary Table S1. https://doi.org/10.3390/microorganisms8070966.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Interfaculty Bioinformatics Unit and Swiss Institute of Bioinformatics, University of Bern, 3012 Bern, Switzerland. [2]Methods Development and Analytics, Agroscope, Schwarzenburgstrasse 161, CH-3003 Bern, Switzerland.

Received: 15 August 2022 Accepted: 14 December 2022
Published online: 27 December 2022

References

1. Winsor GL, Lam DKW, Fleming L, Lo R, Whiteside MD, Yu NY, et al. Pseudomonas Genome Database: Improved comparative analysis and population genomics capability for Pseudomonas genomes. Nucleic Acids Res. 2011 Jan;39(SUPPL. 1).
2. Jayakodi M, Choi BS, Lee SC, Kim NH, Park JY, Jang W, et al. Ginseng genome database: an open-access platform for genomics of Panax ginseng. BMC Plant Biol. 2018 Apr;12:18(1).
3. Arias-Baldrich C, Silva MC, Bergeretti F, Chaves I, Miguel C, Saibo NJM, et al. CorkOakDB-the cork oak genome database portal. Database. 2020;2020.
4. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC Bioinformatics. 2009 Dec;15:10.
5. Nelson ADL, Haug-Baltzell AK, Davey S, Gregory BD, Lyons E. EPIC-CoGe: managing and analyzing genomic data. Bioinformatics. 2018;34(15):2651–3.
6. Dehal PS, Joachimiak MP, Price MN, Bates JT, Baumohl JK, Chivian D, et al. MicrobesOnline: An integrated portal for comparative and functional genomics. Nucleic Acids Res. 2009 Nov;38(SUPPL.1).
7. Harris TW, Arnaboldi V, Cain S, Chan J, Chen WJ, Cho J, et al. WormBase: a modern model organism information resource. Nucleic Acids Res. 2020 Jan 1;48(D1):D762–7.
8. Nguyen NTT, Vincens P, Crollius HR, Louis A. Genomicus 2018: karyotype evolutionary trees and on-the-fly synteny computing. Nucleic Acids Res. 2018 Jan 1;46(D1):D816–22.
9. Vallenet D, Calteau A, Dubois M, … PAN acids, 2020 undefined. MicroScope: an integrated platform for the annotation and exploration of microbial gene functions through genomic, pangenomic and metabolic comparative analysis. academic.oup.com [Internet]. [cited 2022 Nov 23]; Available from: https://academic.oup.com/nar/article-abstract/48/D1/D579/5606622
10. Pillonel T, Tagini F, Bertelli C, Greub G. ChlamDB: a comparative genomics database of the phylum Chlamydiae and other members of the Planctomycetes-Verrucomicrobiae-Chlamydiae superphylum. Nucleic Acids Res. 2020;48(D1):D526–34.
11. Django Software Foundation. Django [Internet]. Lawrence, Kansas: Django Software Foundation; 2013 [cited 2021 Jan 1]. Available from: https://djangoproject.com/
12. Wilkinson MD, Dumontier M. Aalbersberg IjJ, Appleton G, Axton M, Baak a, et al. the FAIR guiding principles for scientific data management and stewardship. Sci Data. 2016;3(1):1–9.
13. Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes [Internet]. Vol. 28, Nucleic Acids Research. 2000. Available from: http://www.genome.ad.jp/kegg/
14. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene ontology: tool for the unification of biology the gene ontology consortium* [internet]. 2000. Available from: http://www.flybase.bio.indiana.edu
15. Carbon S, Douglass E, Good BM, Unni DR, Harris NL, Mungall CJ, et al. The gene ontology resource: enriching a GOld mine. Nucleic Acids Res. 2021 Jan 8;49(D1):D325–34.
16. Huerta-Cepas J, Szklarczyk D, Heller D, Hernández-Plaza A, Forslund SK, Cook H, et al. EggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. Nucleic Acids Res. 2019;47(D1):D309–14.
17. Seemann T. Prokka: rapid prokaryotic genome annotation. Bioinformatics. 2014 Jul 15;30(14):2068–9.
18. Li W, O'Neill KR, Haft DH, Dicuccio M, Chetvernin V, Badretdin A, et al. RefSeq: expanding the prokaryotic genome annotation pipeline reach with protein family model curation. Nucleic Acids Res. 2021 Jan 8;49(D1):D1020–8.
19. Merkel D. Docker: lightweight linux containers for consistent development and deployment. Linux journal. 2014;2014(239):2.
20. Zulkower V, Rosser S. DNA features viewer: a sequence annotation formatting and plotting library for Python. Bioinformatics. 2020 Aug 1;36(15):4350–2.
21. Bolleman J, Bansal P, Redaschi N. SwissBioPics [Internet]. https://www.swissbiopics.org/. 2021 [cited 2021 Sep 1]. Available from: https://www.swissbiopics.org/
22. Schoch CL, Ciufo S, Domrachev M, Hotton CL, Kannan S, Khovanskaya R, et al. NCBI Taxonomy: a comprehensive update on curation, resources and tools. Database [Internet]. 2020 Jan 1;2020:baaa062. Available from: https://doi.org/10.1093/database/baaa062.
23. Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal omega. Mol Syst Biol. 2011;7.
24. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol. 2013 Apr;30(4):772–80.
25. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004;32(5):1792–7.
26. Yachdav G, Wilzbach S, Rauscher B, Sheridan R, Sillitoe I, Procter J, et al. MSAViewer: interactive JavaScript visualization of multiple sequence alignments. Bioinformatics. 2016 Nov 15;32(22):3501–3.
27. Kanehisa M, Sato Y, Kawashima M, Furumichi M, Tanabe M. KEGG as a reference resource for gene and protein annotation. Nucleic Acids Res. 2016;44(D1):D457–62.
28. Kanehisa M, Sato Y, Morishima K. BlastKOALA and GhostKOALA: KEGG tools for functional characterization of genome and metagenome sequences. J Mol Biol [Internet]. 2016;428(4):726–31 https://www.sciencedirect.com/science/article/pii/S002228361500649X.
29. Roder T. KeggMapWizard [Internet]. Bern: GitHub; 2021. https://github.com/MrTomRod/kegg-map-wizard
30. Blanco-Míguez A, Fdez-Riverola F, Sánchez B, Lourenço A. BlasterJS: a novel interactive JavaScript visualisation component for BLAST alignment results. PLoS One. 2018 Oct;13(10).
31. Goussarov G, Goussarov G, Cleenwerck I, Mysara M, Leys N, Monsieurs P, et al. PaSiT: a novel approach based on short-oligonucleotide frequencies for efficient bacterial identification and typing. Bioinformatics. 2020 Apr 15;36(8):2337–44.
32. Kunzmann P, Hamacher K. Biotite: a unifying open source computational biology framework in Python. BMC Bioinformatics. 2018 Oct;1:19(1).
33. Emms DM, Kelly S. OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biol. 2019 Nov;14:20(1).

34. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. In: Vol. 12, Nature Methods: Nature Publishing Group; 2014. p. 59–60.
35. Gibbs AJ, Mcintyre GA. The diagram, a method for comparing sequences its use with amino acid and nucleotide sequences. Eur J Biochem. 1970;16.
36. Marçais G, Delcher AL, Phillippy AM, Coston R, Salzberg SL, Zimin A. MUMmer4: A fast and versatile genome alignment system. PLoS Comput Biol. 2018 Jan;14(1).
37. Maria Nattestad. Dot - an interactive dot plot viewer for genome-genome alignments. https://github.com/MariaNattestad/dot. 2021.
38. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020 Mar 1;17(3):261–72.
39. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B Methodol. 1995;57(1):289–300.
40. Thomas Roder. flower-plot [Internet]. GitHub. 2021 [cited 2022 Jan 1]. Available from: https://github.com/MrTomRod/flower-plot
41. Roder T, Wüthrich D, Bär C, Sattari Z, von Ah U, Ronchi F, et al. In Silico comparison shows that the Pan-genome of a dairy-related bacterial culture collection covers Most reactions annotated to human microbiomes. Microorganisms. 2020;8(7):966.

Publisher's Note