BioData Mining

BRIEF REPORT Open Access

Seven quick tips for gene-focused computational pangenomic analysis

Vincenzo Bonnici[1*] and Davide Chicco[2,3*]

*Correspondence:
vincenzo.bonnici@unipr.it;
davidechicco@davidechicco.it

[1] Present Address: Dipartimento di Scienze Matematiche Fisiche e Informatiche, Università di Parma, Parma, Italy
[2] Dipartimento di Informatica Sistemistica e Comunicazione, Università di Milano-Bicocca, Milan, Italy
[3] Institute of Health Policy Management and Evaluation, University of Toronto, Toronto, Ontario, Canada

Abstract

Pangenomics is a relatively new scientific field which investigates the union of all the genomes of a clade. The word *pan* means *everything* in ancient Greek; the term pangenomics originally regarded genomes of bacteria and was later intended to refer to human genomes as well. Modern bioinformatics offers several tools to analyze pangenomics data, paving the way to an emerging field that we can call *computational pangenomics*. Current computational power available for the bioinformatics community has made computational pangenomic analyses easy to perform, but this higher accessibility to pangenomics analysis also increases the chances to make mistakes and to produce misleading or inflated results, especially by beginners. To handle this problem, we present here a few quick tips for efficient and correct computational pangenomic analyses with a focus on bacterial pangenomics, by describing common mistakes to avoid and experienced best practices to follow in this field. We believe our recommendations can help the readers perform more robust and sound pangenomic analyses and to generate more reliable results.

Keywords: Pangenomics, Genomics, Bioinformatics, Computational biology, Pangenome, Recommendations, Guidelines, Quick tips

Introduction

Pangenomic analysis aims at recognizing the sharing of biological information between living organisms [1]. The term was introduced by Tettelin and colleagues [2] in studying the genomic composition of multiple pathogenic isolates of *Streptococcus agalactiae* motivated by the fact that a single genome does not reflect how genetic variability drives pathogenesis within a bacterial species, and also limits genome-wide screens for vaccine candidates or for antimicrobial targets.

A building block in a pangenomic study is the identification of homologies among the genes that compose the input genomes. A gene family is a set of several similar genes, formed by duplication of a single original gene [3]. In this context, it is a good idea to cluster genes into gene families in order to identify the presence of a family within a genome and to study its genetic composition, but more importantly to understand the global genetic composition of the whole group of genomes. For this purpose, the concept of sequence homology is taken into account [4]. Transmission of genetic material

might occur vertically or horizontally [5], which, added up to gene duplication, makes us distinguish homology into three categories: paralogy (namely copies of the same gene produced by a duplication event); orthology (the same gene transmitted vertically produced by a speciation event); and xenology (given by a gene transmitted horizontally between two genomes). Thus, a gene family is a set of genes that are in relation through such type of homologies. Pangenomic studies are then essentially based on such information. In accordance with homology relations, the presence of a family in the whole group of genomes identifies it as a core family. On the contrary, the absence of the gene family within one or more genomes defines the family as "accessory", thus not constituting an essential biological function for the whole group.

Pangenomic analysis can be seen as a way of exploring the distribution of genetic information across a group of genomes, to determine if the group complexity is saturated, hence the pangenome is considered *closed*, or if any new genome that is included in the analysis increases the pangenome size [6], in which case the pangenome is considered *open*. This approach relies on the use of mathematical models to predict how fast we would expect a pangenome to reach a plateau in an open or closed pangenome. In general, a closed pangenome is an indication that we have already discovered the majority of the genomic content of a given group of organisms. In contrast, an open pangenome is clear evidence that more has to be discovered from that clade. Recently, Rubio et al. [7] showed, by means of a pangenomic analysis of *Streptococcus pneumonia* strains, that accessory genes help to increase functional redundancy in bacteria.

Alternative approaches switch from such a gene-level pangenomic information to the level in which whole DNA sequences are taken into account. In these approach, the pangenomic content of genomes is represented through formal languages or graph structures [8, 9], highly used to represent population-level information in human studies [10]. However, such a type of representation aims at recognizing similarity in the individual's variations of the DNA sequence making no distinction between particular genomic regions, such as genes. In this study, we only consider gene-level analyses, rather than genomic-level approaches.

Pangenomics, however, can also be exploited to investigate specific biological questions, through a plethora of downstream applications. During disease outbreaks, for instance, pangenomic studies can be employed to characterize bacterial isolates and to track the spread of infections by comparing the genomes of different isolates, helping public health authorities implement targeted control measures and prevent further spread [11]. More in general, the collective analysis of all the genes is developed for many specific interests, for example, for the study of a bacterial strain of a given species [12, 13]. Pangenome analyses found many applications in clinical studies [14, 15], for example, they help in identifying drug-target genes in clinical studies [16, 17], or in exploring phylogenetic lineages of bacteria [18] that can be linked to strain-specific disease phenotypes [19], or for recognizing possible antiviral response in bacteria [20] .

Figure 1 summarizes the main steps of a workflow for pangenomic analyses in which the genetic composition of genomes is retrieved and analysed for detecting gene family composition by means of genetic homology. The result of such a clustering step is presented to the researcher for downstream analyses possibly by aggregating it with supplementary information (that is, gene coordinates, biological function, etc.). It is

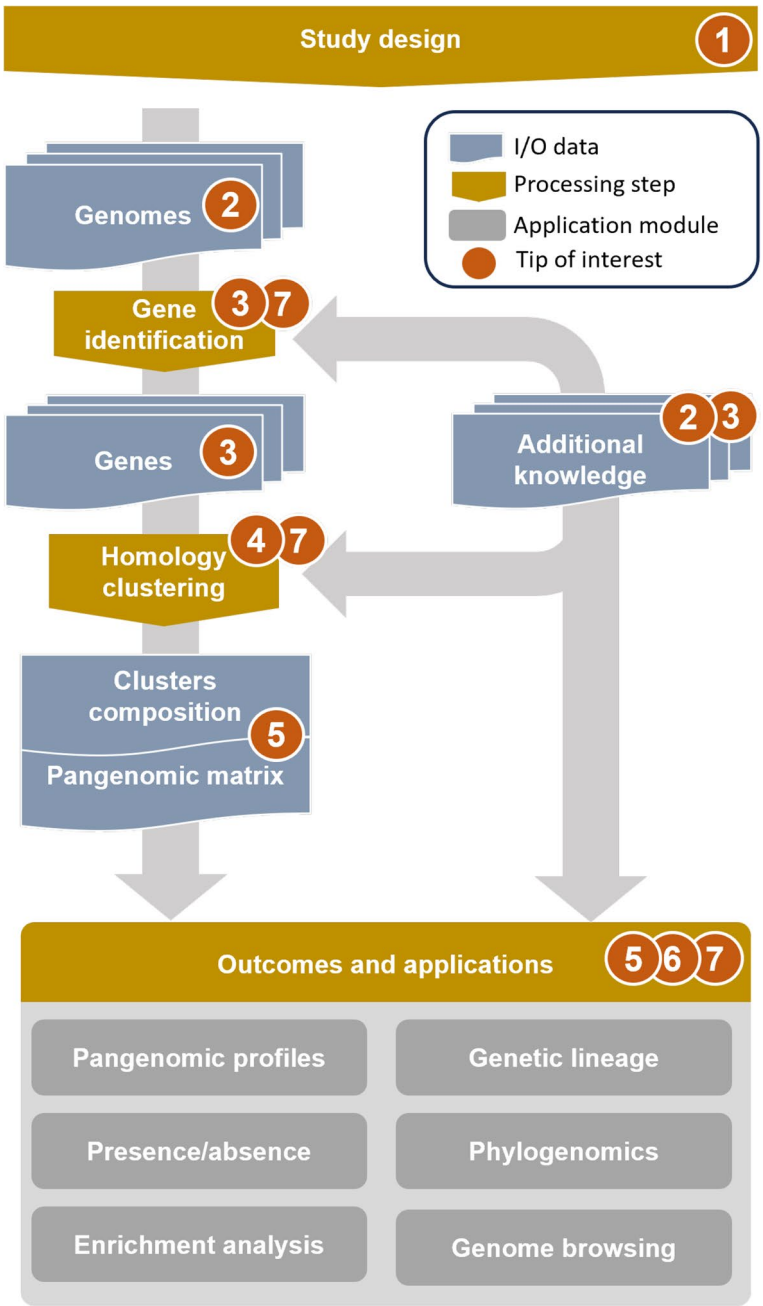

Fig. 1 Architecture of a pangenomic analysis workflow. Red circles refer to the tips described in this study. Examples of application modules for downstream analyses are reported in [21–26]

important to notice that computational pangenomic analyses need extensive computational resources. Additional knowledge is anything that is not the mere sequences of genes. The calculation of homology is mainly based on genetic sequences, which is the essential information needed by every tool for computing gene families. Any other additional data is an accessory. It can help with the homology computation, or it can be used for downstream analyses.

Concerning the tips proposed in this document and reported in Fig. 1, Tip 1 puts attention to the research question of the reader and how it relates to the other tips. Issues related to input data and its quality are reported in Tip 2, while output formats and their visualization are the focus of Tip 5. Because this study mainly regards gene-level pangenomics, Tip 3 gives clues on genome annotation strategies and they can affect pangenomic analyses. Tip 4 pays attention to the core of gene-level pangenomics, which is the computation of sequence homology and the clustering of genetic sequences into gene families. Tip 6 gives guidelines on how to critically evaluate the results of a pangenomic detection methodology. Lastly, but not less important, Tip 7 claims openness and reproducibility of data and experiments.

So far, the Quick Tips article series has published manuscript on several topics (for example, [27–33]), but not on pangenomic analysis. We fill this gap by presenting our recommendations on this theme: a list of simple tips that should be followed by anyone working on this field, to avoid common mistakes and errors.

Our goal is to provide some advice on how to properly conduct a computational pangenomic analysis and help you obtain a pangenome which is reliable, reproducible, and that will serve as a starting point for answering your research question.

Tip 1: Ensure the data in your hands may answer your research question

Having a clear research question may seem trivial for an inexperienced researcher. However, it is fundamental to know what is the ultimate goal of your research project because several aspects of the analysis will be affected by this. For sure, it is possible that pieces of evidence can arise from agnostic data analysis, but, in a general case, it is essential to plan a research goal and then find the instruments for driving toward such a goal. First and foremost, the ultimate goal of your analysis depend on the conditions of the data you use as input. This means that when you start a project on bacterial pangenomics you do not only need to determine the set of organisms that best reflects the problem you want to solve, but you also have to make sure that the data you have has the right characteristics. It is accepted that bacteria from the same species have big differences in their genetic content, due to horizontal gene transfer and mutations, leading to substantial differences in phenotype [34]. If the genomic sequence quality and depth are insufficient to capture strain-level information, the pangenomic analysis will not be able to detect intra-species variation in the gene family identification process [35, 36]. This means that, even if you ensure the quality of your sequencing [37, 38], in this step, you may find that the data you require to solve a question will not always be available, and this means that you might want to reshape your research question. Being aware of the data you need affects the feasibility of your analysis and ultimately your project.

Another point of relevance is that the questions you may want to solve normally require some downstream analysis that is specific to the question at hand. The key is understanding what is required to perform these tasks, as their input is the output of your pangenomic analysis. Depending on downstream needs, you may need to use different pangenomic analysis tools. For example, your goal might be to study the taxonomic lineage of a species. A common approach is to identify genes that are present in all genomes being analyzed, called *core genes*, and use these to build a phylogenetic tree [39]. However, you might also consider to use for this purpose a gene

presence-absence matrix, which contains information about each gene's presence across genomes, information that would otherwise be missed when using only core genes. In that case, you need to make sure the tool you use provides that information [40]. This type of data representation would be useful also if you wanted to use a wider core gene threshold by taking all genes present at least in a percentage of genomes, instead of all of them [41].

What is important is knowing where you are headed with your analysis, because this will help you make the right decisions along the way.

Tip 2: Know your pangenomic input data and double-check its quality

Complete and fragmented genomes usually come in the form of one-genome-one-file, such that multiple fragments of the same genomes are included within the same file. Regarding metagenomic material, it is usually distributed in the form of a Metagenome-Assembled Genome (MAG) [42], which is a single-taxon assembly based on binned sequences that have been asserted to be a close representation of an actual individual genome. When selecting analysis tools, ensure that they are designed to handle the type of data you are working with.

Additionally, always make sure your data has been quality-checked and filtered to remove any potential issues [37, 38]. This can mean different things depending on the data at hand. If you start your analysis from raw reads, you will want to make sure you discard low-quality reads and contaminants. When reads have been assembled, make sure they respect the highest standards in terms of completeness and contamination [43]. Be aware that analyzing highly fragmented genomes using a tool that does not account for that can introduce biases in your analysis, leading to an overestimation of the pangenome size and an increase in the number of singletons [41]. In fact, many of the genomes that are available in public databases are at draft level. Some existing methodologies are able to deal with them [40, 44–46], however, it is always convenient to understand when a genome is too fragmented to be a good source of pangenomic information. Thus, you should evaluate the number of fragments, their length and the various statistics based on them, such as N50, L50, and U50 [47]. You can also check if the quality of those fragmented genomes is fine by running a phylogenetic analysis including reference non-fragmented genomes and by checking the accordance between the expected position of the fragmented genome within the obtained phylogenetic tree and its resultant location. In some cases, when you are dealing with assembled genomes, you might come across unmapped reads. Unfortunately, there is only one tool that can effectively utilize these reads, and prematurely discarding them may result in missed analysis opportunities [45]. By recognising and addressing these considerations, you can conduct a more accurate and meaningful analysis of your pangenomic data.

Tip 3: Be mindful of how your sequences are annotated

Biological sequences are usually distributed in the form of FASTA files [48], which are textual files in which the nucleotidic or amino acidic string of one or more sequences are reported. No additional information is included in such files, that, instead, is embedded in file formats such as GBK (GenBank file format) [49] and GFF (General Feature Format) [50], which version 2 is identical to GTF (Gene Transfer Format). The aim of such

formats is to provide information regarding "annotation" of a given genomic sequence. In the case of genomic sequences, such an annotation can be, for example, the location of the genes contained in it. The GBK file usually contains information on genes and, more in general, on CDS (coding sequences) annotation, but recently non-coding elements have become increasingly common. For this reason, GBK files often report the translation of a gene, its known biological functions and so forth. On the contrary, GFF files are very general. They allow embedding any type of data, but they are more specialized in describing annotation elements within the sequence.

It is important to examine how annotations have been inferred. Which computational tool was used to recognize the elements, with which parameters, and did the annotation pass through manual curation? Tool parameters may be the cause for the discarding of specific elements, such as transfer RNAs (tRNAs), and manual curation may force the deletion of some of them. For this reason, we suggest always running a gene detection tool, such as Prodigal [51] or Prokka [52], but also to evaluate the difference between the output of multiple detection tools. It has to be noticed that in the case of fragmented genomes or metagenomic data, because of the possible lack of the information on some parts of a genome, some of the genes may be missed. This lack of information might have a severe impact on pangenomic studies, for example, by switching a gene family from core to accessory because one or more copies of a given gene have not been sequenced or recognized. Additionally, it is important that the different genomes in the pangenome have been annotated with the same pipeline to avoid biases due to annotation strategy.

Some tools may necessitate an annotated version of the genome, in which case it is important to ensure that the annotations have been generated using the same version of the reference file or the same CDS predictor tool. Be mindful that using manually curated or *in silico* annotation for defining units affects all further steps. Even more importantly, pangenomics can be carried out on any unit of information, from genes, CDS, and sequence chunks [53], making consistency in annotation units and techniques necessary for performing a meaningful comparison.

Tip 4: Use a homology detection approach that is coherent with your input: pay attention to parameters

Homology detection is the key step for clustering single genetic sequences spread across the input genomes into gene families. If you are focusing on functional clustering, the homology of the amino acidic sequences is preferred to nucleotidic sequences because such a piece of information more closely represents the secondary structure of the proteins. However, you may focus the analysis on mere evolutionary considerations and compute nucleoditic similarity that not always relates to secondary structure because of possible frame shifts.

When performing a new pangenomic analysis, it is advisable to employ multiple approaches for homology detection rather than relying solely on one. Finding a consensus between different results will make your results more reliable.

In most cases, tools use alignment-based sequence similarity approaches where similarity is computed by local aligners such as BLAST+ [54]. Although not all tools provide a direct measure of the significance of sequence similarity, when possible, we suggest

you to employ a significance threshold for adjusted *e*-value of 0.005 [55] rather than using the traditional, too permissive 0.05 threshold.

You should also consider that alignment-based measures are not always the best solution and that such alternative sequence similarity measures could be involved, especially if the tool implementing them is aimed to reduce the number of user-defined parameters [56, 57]. Such approaches aim to infer the optimal similarity threshold, which can be specific for a pair of genomes or within a given gene family. In the other cases, when you use a pangenomic analysis tool, it is essential to be mindful of any parameters implemented and their impact on the results, since using different parameters drastically affects the size of the final pangenome [58]. Even when you are employing a parameter-free tool, such as PanDelos [57] or Edgar [56], make sure you completely understand the approach used. If the tool is determining the required thresholds based on the data (that is the case of Edgar) it's a reasonable approach but if the parameters are fixed under the hood by the developer, the tool is not truly parameter-free, you are just not allowed to tweak underlying values, which might result in very unreliable results. In this case, parameters usually are required to account for different characteristics of the genomes under study. Understanding which is the level of similarity and the average nucleotide identity (ANI) [59] that is expected to be found between related genes is essential to set parameters about sequence identity and coverage [60] that are often set to default BLAST values, or in other cases to be generally no lower than 70% of sequence similarity, and for some extreme cases about 50% of coverage and identity. These may largely vary depending on the level of similarity between the genomes under study, whether they belong to the same species, group, or higher taxonomic levels [53], although most commonly pangenomics refers to species-level analyses. Additionally, it is worth noting that different species might have varying evolutionary rates, meaning that within-species similarity can have different implications across different species [61]. Unfortunately, the majority of existing approaches have no specific procedures of parameters for what concerns a specific level of similarity between paralogous sequences. The only current approach that defines a specific treatment for them is PanDelos [57], which sets the sequence similarity between paralogous to be equal or greater than the similarity between the two most similar orthologous genes of two genomes.

To conclude, the construction of a synthetic benchmark by simulating bacterial evolution [62–64] can support the choice of a specific tool, especially when synthetic evolutionary parameters reflect the phylogenomic composition of the studied population. In fact, even if there is evidence that some tools always perform better than others, the most reliable of them are sometimes discordant because they are suitable for specific experiment conditions, such as the level of evolutionary distance between the analysed genomes. Such conditions can be simulated in silico and tools can be run over such artificial benchmarks for understanding which solution works better under such conditions. Evaluating the output of a clustering procedure is not a trivial task. Several measures can be used as an indication of the divergence between the output and the suspected clusters (see [65] for some examples of such evaluation criteria). However, until a golden truth is not available, the match between found and real clusters remains unverified. Synthetic benchmarks provide such a golden truth and thus allow for supervised validation of the obtained clusters. Thus, synthetic benchmarks enable such evaluation because the

expected outcome is known by construction (of the benchmark), which is in contrast with experiments of living organisms.

Tip 5: Be aware of the available output formats and visualization options

Given a set of genomes, composed of genetic sequences, the essential information of a pangenomic analysis is clustering the genetic sequences into gene families. Starting from such a cluster's compositions, usually, a gene of each cluster is selected to be representative of its family. An information surrogated from such clustering data is the presence-absence matrix, also called pangenomic matrix, which reports the presence-absence of each identified gene family for each of the input genomes. No standards are currently employed for the storage and representation of cluster composition and presence-absence matrix, which usually come in the form of raw text files.

Some available tools might help you visualize the presence-absence matrix, as well as other aspects regarding gene family composition, the inferred evolutionary history and their distribution along the genomes [66]. These tools are not intended for providing a pangenomic content discovery methodology but only for visualizing the results and for running downstream analysis. Thus, you need to convert the output of content discovery tools to meet the visualization platform's input formats. In contrast, other solutions, such as Roary [60] already provides embedded visualization instruments. Unfortunately, the methodologies for homology detection that better perform in general conditions [61] and for fragmented genomes [44] are still strictly focused on the clustering step and lack downstream/visualization instruments.

And eventually, as it has been said elsewhere for pathway enrichment analysis [33], keep in mind that different visualization techniques can highlight or hide different results, even in pangenomics.

Tip 6: Critically evaluate the resulting gene families

Be always ready to critically evaluate the output produced by a specific methodology. Because of their intrinsic behavioural difference, different tools may produce different gene family compositions that need to be evaluated case by case. No tool is able to always provide the correct output.

Given an output gene family, always compare the number of genetic sequences that compose the family with the number of input genomes. When doing this, be aware that paralogs increase the family size but not the number of genomes a family is present in, also called *diffusivity* [57]. Hence some tools may merge two or more families, with a resultant increased family size, but still preserving the diffusivity of the most diffused family. To catch this aspect, it is always a good practice to evaluate the functional coherence of the genes of the family and to analyze and visualize phylogenetic relationships among the genes. Functional coherence can be evaluated by comparing functions assigned to genes that can be already known or predicted via tools, such as Prokka [52]. Moreover, in case a particular genetic biomarker is already known for the species involved in your study, you should check the presence of this gene across genomes. However, it has to be noticed that gene families are computed *strictly*, which means that a gene might belong to only one gene family. This aspect excludes the possibility of

representing gene fusions [67–70], or any other evolutionary event that is not embedded in the problem modelling of a given tool.

When performing these critical evaluations, keep in mind that public resources may carry misidentification of genomes, as well as incomplete or incorrect gene functional annotation [71–73]. Moreover, because of the intense horizontal exchange of genetic material among microbial organisms, the actual scheme of evolutionary events is closer to being a "web of life" [5], rather than a tree. However, the evolution of genetic families can still be evaluated in terms of bifurcating trees. Thus, a single phylogenetic tree might be insufficient to evaluate the phylogenomic relationships of your organisms. Alternatively, phylogenetic analyses of specific strains by means of core genes may be more informative and reliable than whole genome comparisons. This analysis can be performed by using alignment-free tools, such as `CVTree` [74], on the sets of sequences of core genes, or by multiple sequence alignment of them [75].

Thus, be open to collaboration: pangenomics is a field which requires computational expertise (which can be computationally demanding) but also necessitates a good understanding of the species being analyzed to start from a biological question which is relevant and understanding the result.

Tip 7: Make your pangenomic analyses open and reproducible

Reproducibility has become a pivotal topic in bioinformatics in recent years: with the availability of massive computational resources (in terms of microprocessor capacity and memory available for the computation) and fast computers, the possibility to reproduce a computational analysis has become easier and broader, even for bioinformatics beginners [76, 77]. Making a study reproducible is also pivotal to allow external researchers to find possible mistakes in the computational pipeline, helping generate more robust results. Even if computational resources for scientific reproducibility are available at low cost worldwide, bioinformatics analyses can be replicated and reproduced only if open science best practices are taken into account:

(a) The usage of open-source programming languages and software platforms;
(b) The sharing of data publicly online;
(c) The sharing of your open software code publicly online;
(d) The publication in open-access journals.

Open source programming languages and software platforms, such as R or Python, in fact, are necessary to make a pangenomic analysis reproducible by anyone, since they are free and have an open license. The R statistical computing language, in particular, provides two bioinformatics platforms which supply a larger number of R software libraries for computational biology analyses: Bioconductor [78] and Bioconda [79]. Bioconductor provides the `PanViz` [66] software library for pangenomes' visualization, and Bioconda furnishes the `PPanGGOLiN` [80, 81] software package for pangenome partition and the `PanTools` [82], Pangenome Graph Builder (`PGGB`) [83], `PanX` [26], `Pagoo` [84], and `pgr-tk` [85] software libraries for pangenomic data analysis.

A few packages are available for Python as well [86]. The recently-released programming language Julia also provides a software library for pangenome graph creation [87].

Regarding application programming interfaces (API) and visualization tools, we mention `ODGI` [88].

On the contrary, the usage of proprietary programming languages makes the replication of the analysis doable only by people who have that license.

Releasing software code on online platforms such as GitHub [89] and GitLab [90], moreover, can enhance the possibility to reproduce a study [91]. Sharing data online is another key component of reproducibility: a pangenomic analysis can be re-performed openly only if its datasets are available online to anyone without restrictions. Therefore, we suggest publishing your raw and processed datasets in open online repositories such as Gene Expression Omnibus (GEO) [92], ArrayExpress [93], Sequence Read Archive (SRA) [94], Kaggle [95], Figshare [96], Zenodo [97], or the University of California Irvine Machine Learning Repository [98], following the principles of FAIR (Findability, Accessibility, Interoperability, and Reuse) data sharing [99]. In case you are implementing a new tool for pangenomic analysis, the use of synthetic benchmarks is crucial for allowing a quantitative evaluation of the results and a fair comparison with other tools.

Similarly, regarding the paper writing and publishing, we recommend submitting your article to an open-access journal. Once published, your article will be available to be read for free by anyone in the world, even in the least developed countries. A list of open-access journals in bioinformatics can be found on the ScimagoJR website [100].

Conclusions

In the context of bacterial and more in general microbiome research, pangenomic studies exploit advanced bioinformatics tools to explore the genetic content of various organisms, providing valuable insights into genetic diversity and evolution. A core procedure is the clustering of genetic sequences spread along input genomes into gene families by means of homology computation. How this step of computational pangenomic pipelines significantly affects results and downstream analyses. By following the seven tips outlined here, researchers can enhance the reliability and reproducibility of their pangenomic analyses. Ensuring clear research questions, high-quality input data, appropriate annotation strategies, and critical evaluation of results are fundamental steps. Additionally, utilizing open-source tools and sharing data openly is crucial for advancing the field and fostering collaboration. Ultimately, these practices contribute to a more thorough and accurate understanding of genetic landscapes, paving the way for future discoveries and innovations in microbial research.

Abbreviations

API	Application programming interface
ANI	Average nucleotide identity
CDS	Coding sequences
FAIR	Findability, Accessibility, Interoperability, and Reuse
GBK	GenBank file format
GEO	Gene Expression Omnibus
GFF	General Feature Format
MAG	Metagenome-Assembled Genome
PGGB	Pangenome Graph Builder
RNA	Ribonucleic acid
SRA	Sequence Read Archive
tRNAs	Transfer RNAs

Acknowledgements
The authors thank Claudia Mengoni (Università di Trento) for her help.

Authors' contributions
V.C. designed the study and contributed to the writing of the manuscript. D.C. supervised the study and contributed to the writing of the manuscript. All the authors reviewed and approved this version of the manuscript.

Funding
The work of D.C. was funded by the European Union - Next Generation EU programme, in the context of The National Recovery and Resilience Plan, Investment Partenariato Esteso PE8 "Conseguenze e sfide dell'invecchiamento", Project Age-It (Ageing Well in an Ageing Society), and also partially supported by Ministero dell'Università e della Ricerca of Italy under the "Dipartimenti di Eccellenza 2023-2027" ReGAInS grant assigned to Dipartimento di Informatica Sistemistica e Comunicazione at Università di Milano-Bicocca. V.B. is partially supported by the Università di Parma, project number MUR_DM737_B_MAFI_BONNICI, and by the CINI (Consorzio Interuniversitario Nazionale per l'Informatica) InfoLife laboratory.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Availability of data and materials
Not applicable.

Data availability
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 27 March 2024 Accepted: 12 August 2024
Published online: 03 September 2024

References

1. Tettelin H, Medini D. The pangenome: Diversity, dynamics and evolution of genomes. Berlin: Springer Nature; 2020.
2. Tettelin H, Masignani V, Cieslewicz MJ, Donati C, Medini D, Ward NL, et al. Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome". Proc Natl Acad Sci. 2005;102(39):13950–5.
3. Nussbaum RL, McInnes RR, Willard HF. Thompson & Thompson genetics in medicine. Amsterdam: Elsevier Health Sciences; 2015.
4. Koonin EV. Orthologs, paralogs, and evolutionary genomics. Annu Rev Genet. 2005;39:309–38.
5. Soucy SM, Huang J, Gogarten JP. Horizontal gene transfer: building the web of life. Nat Rev Genet. 2015;16(8):472–82.
6. Hiller NL, Janto B, Hogg JS, Boissy R, Yu S, Powell E, et al. Comparative genomic analyses of seventeen *Streptococcus pneumoniae* strains: insights into the pneumococcal supragenome. J Bacteriol. 2007;189(22).
7. Rosconi F, Rudmann E, Li J, Surujon D, Anthony J, Frank M, et al. A bacterial pan-genome makes gene essentiality strain-dependent and evolvable. Nat Microbiol. 2022;7(10):1580–92.
8. Bonizzoni P, De Felice C, Pirola Y, Rizzi R, Zaccagnino R, Zizza R. Can formal languages help pangenomics to represent and analyze multiple genomes? In: Proceedings of DLT 2022 – the 22nd International Conference on Developments in Language Theory. Berlin: Springer; 2022. pp. 3–12.
9. Baaijens JA, Bonizzoni P, Boucher C, Della Vedova G, Pirola Y, Rizzi R, et al. Computational graph pangenomics: a tutorial on data structures and their applications. Nat Comput. 2022;21(1):81–108.
10. Liao WW, Asri M, Ebler J, Doerr D, Haukness M, Hickey G, et al. A draft human pangenome reference. Nature. 2023;617(7960):312–24.
11. Ceres KM, Stanhope MJ, Gröhn YT. A critical evaluation of *Mycobacterium bovis* pangenomics, with reference to its utility in outbreak investigation. Microb Genomics. 2022;8(6):1–8.
12. Medini D, Donati C, Tettelin H, Masignani V, Rappuoli R. The microbial pan-genome. Curr Opin Genet Dev. 2005;15(6):589–94.
13. Tettelin H, Riley D, Cattuto C, Medini D. Comparative genomics: the bacterial pan-genome. Curr Opin Microbiol. 2008;11(5):472–7.
14. Holt KE, Parkhill J, Mazzoni CJ, Roumagnac P, Weill FX, Goodhead I, et al. High-throughput sequencing provides insights into genome variation and evolution in Salmonella typhi. Nat Genet. 2008;40(8):987–93.
15. Earle SG, Wu CH, Charlesworth J, Stoesser N, Gordon NC, Walker TM, et al. Identifying lineage effects when controlling for population structure improves power in bacterial association studies. Nat Microbiol. 2016;1(5):1–8.
16. Serruto D, Serino L, Masignani V, Pizza M. Genome-based approaches to develop vaccines against bacterial pathogens. Vaccine. 2009;27(25–26):3245–50.

17. Muzzi A, Masignani V, Rappuoli R. The pan-genome: towards a knowledge-based discovery of novel targets for vaccines and antibacterials. Drug Discov Today. 2007;12(11–12):429–39.
18. Zhang Y, Sievert SM. Pan-genome analyses identify lineage-and niche-specific markers of evolution and adaptation in epsilonproteobacteria. Front Microbiol. 2014;5:71536.
19. D'Auria G, Jiménez-Hernández N, Peris-Bondia F, Moya A, Latorre A. Legionella pneumophila pangenome reveals strain-specific virulence factors. BMC Genomics. 2010;11:1–13.
20. Rubio A, Sprang M, Garzón A, Moreno-Rodriguez A, Pachón-Ibáñez ME, Pachón J, et al. Analysis of bacterial pangenomes reduces CRISPR dark matter and reveals strong association between membranome and CRISPR-Cas systems. Sci Adv. 2023;9(12):eadd8911.
21. Chaudhari NM, Gupta VK, Dutta C. BPGA-an ultra-fast pan-genome analysis pipeline. Sci Rep. 2016;6(1):24373.
22. Eren AM, Esen ÖC, Quince C, Vineis JH, Morrison HG, Sogin ML, et al. Anvi'o: an advanced analysis and visualization platform for 'omics data. PeerJ. 2015;3:e1319.
23. Lukjancenko O, Thomsen MC, Voldby Larsen M, Ussery DW. PanFunPro: pan-genome analysis based on FUNctional PROfiles. F1000Research. 2013;2:265.
24. Snipen L, Ussery DW. Standard operating procedure for computing pangenome trees. Stand Genomic Sci. 2010;2(1):135–41.
25. Snipen L, Liland KH. micropan: an R-package for microbial pan-genomics. BMC Bioinformatics. 2015;16:1–8.
26. Ding W, Baumdicker F, Neher RA. panX: pan-genome analysis and exploration. Nucleic Acids Res. 2018;46(1):e5.
27. Lubiana T, Lopes R, Medeiros P, Silva JC, Goncalves ANA, Maracaja-Coutinho V, et al. Ten quick tips for harnessing the power of ChatGPT in computational biology. PLoS Comput Biol. 2023;19(8):e1011319.
28. Hou Q, Waury K, Gogishvili D, Feenstra KA. Ten quick tips for sequence-based prediction of protein properties using machine learning. PLoS Comput Biol. 2022;18(12):e1010669.
29. Lee BD, Gitter A, Greene CS, Raschka S, Maguire F, Titus AJ, et al. Ten quick tips for deep learning in biology. PLoS Comput Biol. 2022;18(3):e1009803.
30. Tang YA, Pichler K, Füllgrabe A, Lomax J, Malone J, Munoz-Torres MC, et al. Ten quick tips for biocuration. PLoS Comput Biol. 2019;15(5):e1006906.
31. Diaz-Uriarte R, Gómez de Lope E, Giugno R, Fröhlich H, Nazarov PV. Nepomuceno-Chamorro IA, et al. Ten quick tips for biomarker discovery and validation analyses using machine learning. PLoS Comput Biol. 2022;18(8):e1010357.
32. Nguyen LH, Holmes S. Ten quick tips for effective dimensionality reduction. PLoS Comput Biol. 2019;15(6):e1006907.
33. Chicco D, Agapito G. Nine quick tips for pathway enrichment analysis. PLoS Comput Biol. 2022;18(8):e1010348.
34. Leimbach A, Hacker J, Dobrindt U. *E. coli* as an all-rounder: the thin line between commensalism and pathogenicity. Between Pathogenicity Commensalism. 2013;358:3–32.
35. Overholt WA, Hölzer M, Geesink P, Diezel C, Marz M, Küsel K. Inclusion of Oxford Nanopore long reads improves all microbial and viral metagenome-assembled genomes from a complex aquifer system. Environ Microbiol. 2020;22(9):4000–13.
36. Wick RR, Judd LM, Gorrie CL, Holt KE. Completing bacterial genome assemblies with multiplex MinION sequencing. Microb Genomics. 2017;3(10)3–32.
37. Gargis AS, Kalman L, Lubin IM. Assuring the quality of next-generation sequencing in clinical microbiology and public health laboratories. J Clin Microbiol. 2016;54(12):2857–65.
38. Smits TH. The importance of genome sequence quality to microbial comparative genomics. BMC Genomics. 2019;20(1):662.
39. Eisen JA, Fraser CM. Phylogenomics: intersection of evolution and genomics. Science. 2003;300(5626):1706–7.
40. Gabrielaite M, Marvig RL. GenAPI: a tool for gene absence-presence identification in fragmented bacterial genome sequences. BMC Bioinformatics. 2020;21(1):1–8.
41. Li T, Yin Y. Critical assessment of pan-genomic analysis of metagenome-assembled genomes. Brief Bioinforma. 2022;23(6):bbac413.
42. Setubal JC. Metagenome-assembled genomes: concepts, analogies, and challenges. Biophys Rev. 2021;13(6):905–9.
43. Bowers RM, Kyrpides NC, Stepanauskas R, Harmon-Smith M, Doud D, Reddy TBK, et al. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. Nat Biotechnol. 2017;35(8):725–31.
44. Bonnici V, Mengoni C, Mangoni M, Franco G, Giugno R. PanDelos-frags: A methodology for discovering pangenomic content of incomplete microbial assemblies. J Biomed Inform. 2023;148:104552.
45. Veras A, Araujo F, Pinheiro K, Guimarães L, Azevedo V, Soares S, et al. Pan4Draft: a computational tool to improve the accuracy of pan-genomic analysis using draft genomes. Sci Rep. 2018;8(1):9670.
46. Tonkin-Hill G, MacAlasdair N, Ruis C, Weimann A, Horesh G, Lees JA, et al. Producing polished prokaryotic pangenomes with the Panaroo pipeline. Genome Biol. 2020;21:1–21.
47. Castro CJ, Ng TFF. U50: a new metric for measuring assembly output based on non-overlapping, target-specific contigs. J Comput Biol. 2017;24(11):1071–80.
48. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proc Natl Acad Sci. 1988;85(8):2444–8.
49. National Library of Medicine. GenBank Overview. 2023. https://www.ncbi.nlm.nih.gov/genbank/. Accessed 4 Nov 2023.
50. Ensembl. GFF/GTF File Format - Definition and supported options. 2023. https://www.ensembl.org/info/website/upload/gff.html. Accessed 4 Nov 2023.
51. Hyatt D, Chen GL, LoCascio PF, Land ML, Larimer FW, Hauser LJ. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics. 2010;11:1–11.
52. Seemann T. Prokka: rapid prokaryotic genome annotation. Bioinformatics. 2014;30(14):2068–9.

53. Vernikos G. A review of pangenome tools and recent studies. In: The Pangenome: Diversity, Dynamics and Evolution of Genomes, chap 4. Berlin: Springer International Publishing; 2020. pp. 89–112.
54. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC Bioinformatics. 2009;10:1–9.
55. Benjamin DJ, Berger JO, Johannesson M, Nosek BA, Wagenmakers EJ, Berk R, et al. Redefine statistical significance. Nat Hum Behav. 2018;2(1):6–10.
56. Blom J, Albaum SP, Doppmeier D, Pühler A, Vorhölter FJ, Zakrzewski M, et al. EDGAR: a software framework for the comparative analysis of prokaryotic genomes. BMC Bioinformatics. 2009;10:1–14.
57. Bonnici V, Giugno R, Manca V. PanDelos: a dictionary-based method for pan-genome content discovery. BMC Bioinformatics. 2018;19(15):47–59.
58. Costa SS, Guimarães LC, Silva A, Soares SC, Baraúna RA. First steps in the analysis of prokaryotic pan-genomes. Bioinforma Biol Insights. 2020;14:1177932220938064.
59. Kim M, Oh HS, Park SC, Chun J. Towards a taxonomic coherence between average nucleotide identity and 16S rRNA gene sequence similarity for species demarcation of prokaryotes. Int J Syst Evol Microbiol. 2014;64(Pt_2):346–351.
60. Page AJ, Cummins CA, Hunt M, Wong VK, Reuter S, Holden MT, et al. Roary: rapid large-scale prokaryote pan genome analysis. Bioinformatics. 2015;31(22):3691–3.
61. Bonnici V, Maresi E, Giugno R. Challenges in gene-oriented approaches for pangenome content discovery. Brief Bioinforma. 2021;22(3):bbaa198.
62. Dalquen DA, Anisimova M, Gonnet GH, Dessimoz C. ALF-a simulation framework for genome evolution. Mol Biol Evol. 2012;29(4):1115–23.
63. Meyer F, Lesker TR, Koslicki D, Fritz A, Gurevich A, Darling AE, et al. Tutorial: assessing metagenomics software with the CAMI benchmarking toolkit. Nat Protoc. 2021;16(4):1785–801.
64. Bonnici V, Giugno R. PANPROVA: pangenomic prokaryotic evolution of full assemblies. Bioinformatics. 2022;38(9):2631–2.
65. Saxena A, Prasad M, Gupta A, Bharill N, Patel OP, Tiwari A, et al. A review of clustering techniques and developments. Neurocomputing. 2017;267:664–81.
66. Pedersen TL, Nookaew I, Wayne Ussery D, Månsson M. PanViz: interactive visualization of the structure of functionally annotated pangenomes. Bioinformatics. 2017;33(7):1081–2.
67. Lovino M, Ciaburri MS, Urgese G, Di Cataldo S, Ficarra E. DEEPrior: a deep learning tool for the prioritization of gene fusions. Bioinformatics. 2020;36(10):3248–50.
68. Lovino M, Montemurro M, Barrese VS, Ficarra E. Identifying the oncogenic potential of gene fusions exploiting miRNAs. J Biomed Inform. 2022;129:104057.
69. Lovino M, Urgese G, Macii E, Di Cataldo S, Ficarra E. Predicting the oncogenic potential of gene fusions using convolutional neural networks. In: Proceedings of CIBB 2018 — the 15th International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics. Berlin: Springer; 2018. pp. 277–84.
70. Citarrella F, Bontempo G, Lovino M, Ficarra E. FusionFlow: an integrated system workflow for gene fusion detection in genomic samples. In: Proceedings of ADBIS 2022 – the 26th European Conference on Advances in Databases and Information Systems. Berlin: Springer; 2022. pp. 79–88.
71. Stavrou AA, Mixão V, Boekhout T, Gabaldón T. Misidentification of genome assemblies in public databases: the case of Naumovozyma dairenensis and proposal of a protocol to correct misidentifications. Yeast. 2018;35(6):425–9.
72. Vilgalys R. Taxonomic misidentification in public DNA databases. New Phytol. 2003;160(1):4–5.
73. Lobb B, Tremblay BJM, Moreno-Hagelsieb G, Doxey AC. An assessment of genome annotation coverage across the bacterial tree of life. Microb Genomics. 2020;6(3)1–11.
74. Qi J, Luo H, Hao B. CVTree: a phylogenetic tree reconstruction tool based on whole genomes. Nucleic Acids Res. 2004;32(suppl_2):W45–W47.
75. Tarracchini C, Argentini C, Alessandri G, Lugli GA, Mancabelli L, Fontana F, et al. The core genome evolution of Lactobacillus crispatus as a driving force for niche competition in the human vaginal tract. Microb Biotechnol. 2023;16(9):1774–89.
76. Wratten L, Wilm A, Göke J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nat Methods. 2021;18(10):1161–8.
77. Markowetz F. Five selfish reasons to work reproducibly. Genome Biol. 2015;16(1):1–4.
78. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, et al. Orchestrating high-throughput genomic analysis with Bioconductor. Nat Methods. 2015;12(2):115–21.
79. Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat Methods. 2018;15(7):475–6.
80. Gautreau G, Bazin A, Gachet M, Planel R, Burlot L, Dubois M, et al. PPanGGOLiN: depicting microbial diversity via a partitioned pangenome graph. PLoS Comput Biol. 2020;16(3):e1007732.
81. Bazin A, Gautreau G, Médigue C, Vallenet D, Calteau A. panRGP: a pangenome-based method to predict genomic islands and explore their diversity. Bioinformatics. 2020;36(Supplement_2):i651–i658.
82. Jonkheer EM, van Workum DJM, Sheikhizadeh Anari S, Brankovics B, de Haan JR, Berke L, et al. PanTools v3: functional annotation, classification and phylogenomics. Bioinformatics. 2022;38(18):4403–5.
83. Garrison E, Guarracino A, Heumos S, Villani F, Bao Z, Tattini L, et al. Building pangenome graphs. bioRxiv. 2023;05.535718:1–14.
84. Ferrés I, Iraola G. An object-oriented framework for evolutionary pangenome analysis. Cell Rep Methods. 2021;1(5):100085.
85. Jayanti R, Kim A, Pham S, Raghavan A, Sharma A, Samanta MP. Comparative Analysis of Plastid Genomes Using Pangenome Research ToolKit (PGR-TK). 2023. https://doi.org/10.48550/arXiv.2310.19110.
86. GitHub. Pangenome Python repositories. 2024. https://github.com/topics/pangenome?l=python. Accessed 24 Jun.

87. PanGraph jl. A fast, self-contained Julia library and command line tool suite to align multiple genomes into a pangenome graph. 2023. https://neherlab.github.io/pangraph/. Accessed 13 Nov 2023.
88. Guarracino A, Heumos S, Nahnsen S, Prins P, Garrison E. ODGI: understanding pangenome graphs. Bioinformatics. 2022;38(13):3319–26.
89. GitHub. Let's build from here. 2023. https://www.github.com. Accessed 4 Nov 2023.
90. GitLab. Software. Faster. 2023. https://www.gitlab.com. Accessed 4 Nov 2023.
91. Barnes N. Publish your computer code: it is good enough. Nature. 2010;467(7317):753.
92. Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res. 2002;30(1):207–10.
93. Brazma A, Parkinson H, Sarkans U, Shojatalab M, Vilo J, Abeygunawardena N, et al. ArrayExpress-a public repository for microarray gene expression data at the EBI. Nucleic Acids Res. 2003;31(1):68–71.
94. Kodama Y, Shumway M, Leinonen R. The Sequence Read Archive: explosive growth of sequencing data. Nucleic Acids Res. 2012;40(D1):D54–6.
95. Kaggle. Kaggle datasets – Explore, analyze, and share quality data. 2022. https://www.kaggle.com/datasets. Accessed 13 Jul 2023.
96. Figshare. Store, share, discover research. 2011. https://www.figshare.com. Accessed 13 Jul 2023.
97. Zenodo. Research, shared. 2013. https://www.zenodo.org. Accessed 13 Jul 2023.
98. University of California Irvine. Machine Learning Repository. 1987. https://archive.ics.uci.edu/. Accessed 13 Jul 2023.
99. Wilkinson MD, Dumontier M, Aalbersberg IJ, 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.
100. Scimago. Journal Rank - Biochemistry, genetics, and molecular biology. 2023. https://scimagojr.com/journalrank.php?openaccess=true&area=1300. Accessed 13 Nov 2023.

Publisher's Note