Bioinformatics, 2025, **41(12)**, btaf626
https://doi.org/10.1093/bioinformatics/btaf626
Advance Access Publication Date: 13 November 2025
Applications Note

OXFORD

Genome Analysis

Synteny plot quality control with SyntenyQC

Timothy D.J. Kirkwood[1,*], Jack A. Connolly[1], Ee Lui Ang[2,3], Huimin Zhao[4,5,6,7], Eriko Takano[1,3,8], Rainer Breitling[1,9,*]

[1]Manchester Institute of Biotechnology, Department of Chemistry, School of Natural Sciences, Faculty of Science and Engineering, University of Manchester, Manchester M1 7DN, United Kingdom
[2]Synthetic Biology Translational Research Program, Yong Loo Lin School of Medicine, National University of Singapore, Singapore 117597, Singapore
[3]Singapore Institute of Food and Biotechnology Innovation (SIFBI), Agency for Science, Technology and Research (A*STAR), Singapore 138669, Singapore
[4]Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States
[5]Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States
[6]NSF Molecular Maker Lab Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States
[7]NSF iBiofoundry, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States
[8]Singapore Integrative Biosystems and Engineering Research (SIBER) Strategic Research Translational Thrust (SRTT), Agency for Science, Technology and Research (A*STAR), Singapore 138635, Singapore
[9]Bioinformatics Institute (BII), Agency for Science, Technology and Research (A*STAR), Singapore 138671, Singapore

*Corresponding authors. Timothy D.J. Kirkwood, Manchester Institute of Biotechnology, Department of Chemistry, University of Manchester, 131 Princess Street, Manchester, M1 7DN, United Kingdom. E-mail: timothy.kirkwood@manchester.ac.uk; Rainer Breitling, A*STAR Bioinformatics Institute (A*STAR BII), A*STAR. Postal Address, A*STAR Bioinformatics Institute (A*STAR BII), 30 Biopolis Street, #07-01 Matrix, Singapore 138671. E-mail: rainer_breitling@a-star.edu.sg.

Associate Editor: Macha Nikolski

Abstract

Summary: SyntenyQC is a data pre-processing tool for the construction of synteny plots. It supports genomic data collection, annotation and dereplication to facilitate (and in some cases fundamentally enable) the construction of informative synteny plots.

Availability and implementation: SyntenyQC is a command line app developed using Python version 3.10 and tested using pytest. SyntenyQC is available on PyPI (https://pypi.org/project/SyntenyQC) under the MIT License, along with a detailed user tutorial. Package tests can be viewed at https://github.com/Tim-Kirkwood/SyntenyQC.

1 Introduction

The synteny plot is a commonly used comparative genomics tool (Gilchrist and Chooi 2021). Synteny plots consist of multiple tracks, each of which represents a separate genome or genomic subsequence. Synteny plots can be drawn at either genome or neighbourhood scales, with the latter being the focus of this work. Each track of a neighbourhood-scale synteny plot represents genes as arrows, whose length corresponds to gene length and direction to gene strand. Typically, links are drawn across tracks between genes that have a shared relationship (e.g. homology), although richer information such as GC content can be included (Hackl *et al.* 2024). Such neighbourhood analyses can (i) indicate misannotated gene sequences, (ii) indicate co-selective relationships between specific neighbourhood genes, and (iii) suggest whether a given neighbourhood appears to have recently arrived via horizontal gene transfer (Amos *et al.* 2017). These questions are particularly pertinent in the context of biosynthetic gene clusters (BGCs), which are clustered genes that encode pathways responsible for the production of clinically and economically relevant natural products (Jensen 2016). Indeed, syntenic analysis and/or plots are central to several BGC databases and annotation tools (Navarro-Muñoz *et al.* 2020, Blin *et al.* 2024, 2025, Salamzade *et al.* 2025, Zdouc *et al.* 2025, Zhang *et al.* 2025).

Various 'low-code' tools have been developed for synteny plot construction. These typically utilize a command line (Sullivan *et al.* 2011, Gilchrist and Chooi 2021, Gilchrist *et al.* 2021) and/or graphical (Sullivan *et al.* 2011, Medema *et al.* 2013, van den Belt *et al.* 2023) user interface to open up flexible, user-driven analyses without requiring programming proficiency. However, challenges remain for users of these tools. Firstly, from a logistical perspective, it is often useful to annotate synteny plot tracks according to the neighbourhoods that they represent, e.g. via an NCBI accession or organism name. However, this can be challenging to automate as current tools do not retain taxonomically relevant information for identified neighbourhoods (Gilchrist and Chooi 2021, van den Belt *et al.* 2023).

More importantly, identifying neighbourhoods for synteny plot analysis can be complicated by redundancy at the neighbourhood (e.g. similar strains) and database (e.g. multiple

Received: 11 June 2025; **Revised:** 13 October 2025; **Accepted:** 28 October 2025

copies of the same genome) levels. In practice, this means that many neighbourhoods will be highly similar, and thus uninformative in the context of a synteny plot (which typically requires a degree of neighbourhood diversity for the comparison of conserved and non-conserved neighbourhood elements). The inclusion of redundant neighbourhoods also increases computational workloads and may preclude the creation of synteny plots altogether—e.g. CAGECAT (van den Belt *et al.* 2023) imposes a limit of 50 neighbourhoods per synteny plot visualization. There is thus a broken link between stand-alone tools that can source neighbourhoods for synteny plot creation, and the creation/interpretation of synteny plots themselves (Fig. 1).

These challenges are addressed by some BGC annotation tools via genome-level (Blin *et al.* 2024, 2025) or neighbourhood-level (Cediel-Becerra *et al.* 2025, Salamzade *et al.* 2025) redundancy filters. However, limited quality control options exist for low-code synteny plot construction. An exception, CAGEcleaner (De Vrieze *et al.* 2025), is uniquely well placed for BGC-specific, host-centric analyses (e.g. pangenomic analysis of hosts that contain a given BGC). However, CAGEcleaner's genome-level redundancy filter is not optimal for synteny plot creation—genomic variation does not guarantee neighbourhood variation, and vice versa, particularly in scenarios of pervasive horizontal gene transfer, which are often of particular interest for this kind of analysis (see Results Section 1, available as supplementary data at *Bioinformatics* online and Fig. 5, available as supplementary data at *Bioinformatics* online). Additionally, CAGEcleaner is not a cross-platform tool, is strictly coupled to cblaster (Gilchrist *et al.* 2021), and applies a complex workflow with multiple parameters to optimize. CAGEcleaner's similarity metric (Average Nucleotide Identity; ANI) fails at similarities below 82% (De Vrieze *et al.* 2025)—whilst acceptable for genome-scale comparisons, this could hinder comparisons of the smaller contigs commonly observed in most sequence databases. Indeed, CAGEcleaner performs poorly on low-quality assemblies (De Vrieze *et al.* 2025), which is not ideal as neither cblaster nor CAGEcleaner perform sequence quality control.

2 Features

SyntenyQC is a Python app for the quality control and automated curation of neighbourhoods prior to synteny plot creation. It offers two subcommands, Collect and Sieve. Sieve is used to remove redundant neighbourhoods from a user-specified folder of GenBank-format neighbourhood files and is thus not tied to any specific upstream tool. However, in recognition of the increasing popularity of cblaster (Gilchrist *et al.* 2021), we also provide an explicit integration point for the cblaster output via Collect, which downloads (optionally extended) cblaster-identified genomic neighbourhoods from NCBI. These neighbourhoods are written to local GenBank-format files, which are named according to either NCBI Accession or Organism.

3 Usage

As an example use case, we applied the workflow shown in Fig. 1 to the classic actinorhodin BGC, using entry BGC0000196 in the 'Minimum Information about a Biosynthetic Gene cluster' (MIBiG) database of verified clusters as a reference (Zdouc *et al.* 2025). The use of cblaster (Gilchrist and Chooi 2021) highlighted 173 neighbourhoods containing potential cluster homologs—unfortunately, this number of neighbourhoods was too high for synteny plot creation using clinker (Gilchrist and Chooi 2021). Processing with SyntenyQC Collect and Sieve reduced the number of neighbourhoods to 139 and 22, respectively. The 22 neighbourhoods identified with Sieve were used to create a synteny plot with clinker, shown in Fig, 2. See Methods Section 1, available as supplementary data at *Bioinformatics* online, for further information on cblaster and SyntenyQC processing, and Fig. 3 for a comparison of neighbourhood numbers in the context of other *Streptomyces coelicolor* BGCs.

Automated annotation of the synteny plot tracks allows us to see that the actinorhodin BGC can only be observed within the *Streptomyces* genus (Fig. 2). However, several potentially related putative BGCs (pBGCs) can be observed outside of

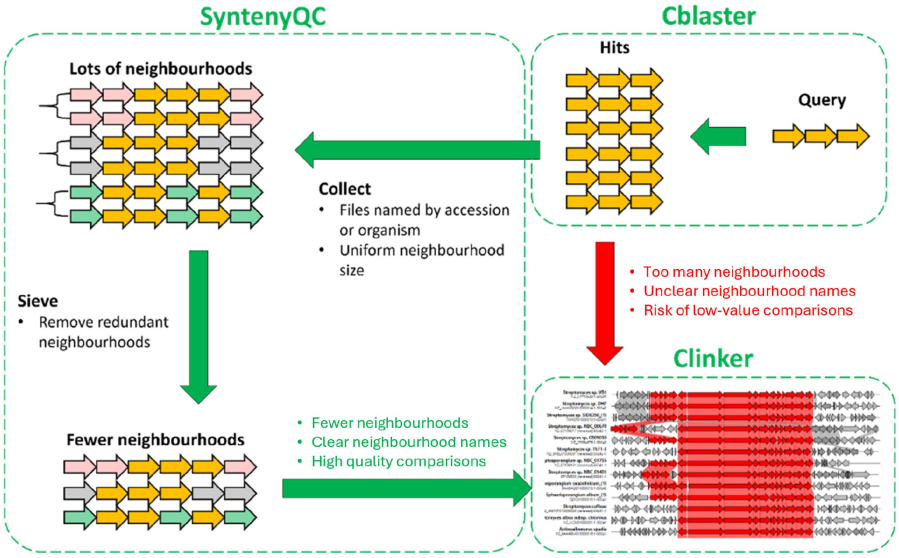


Figure 1. Integrating SyntenyQC into a synteny plot creation workflow. First, cblaster is used to find records (i.e. genomes/contigs) containing a user-specified number of clustered hits to a user-supplied query. The loci of these homologs in each hit record are used to define a neighbourhood for each record, which is extracted using the SyntenyQC Collect subcommand. Files are named according to accession or organism name, facilitating automated annotation of the final synteny plot. Similar neighbourhoods (black braces) are filtered to remove redundant neighbourhoods, with what constitutes 'similar' specified by the user in the form of a similarity threshold. Finally, the collected, sieved neighbourhoods are fed into clinker to create the final synteny plot.

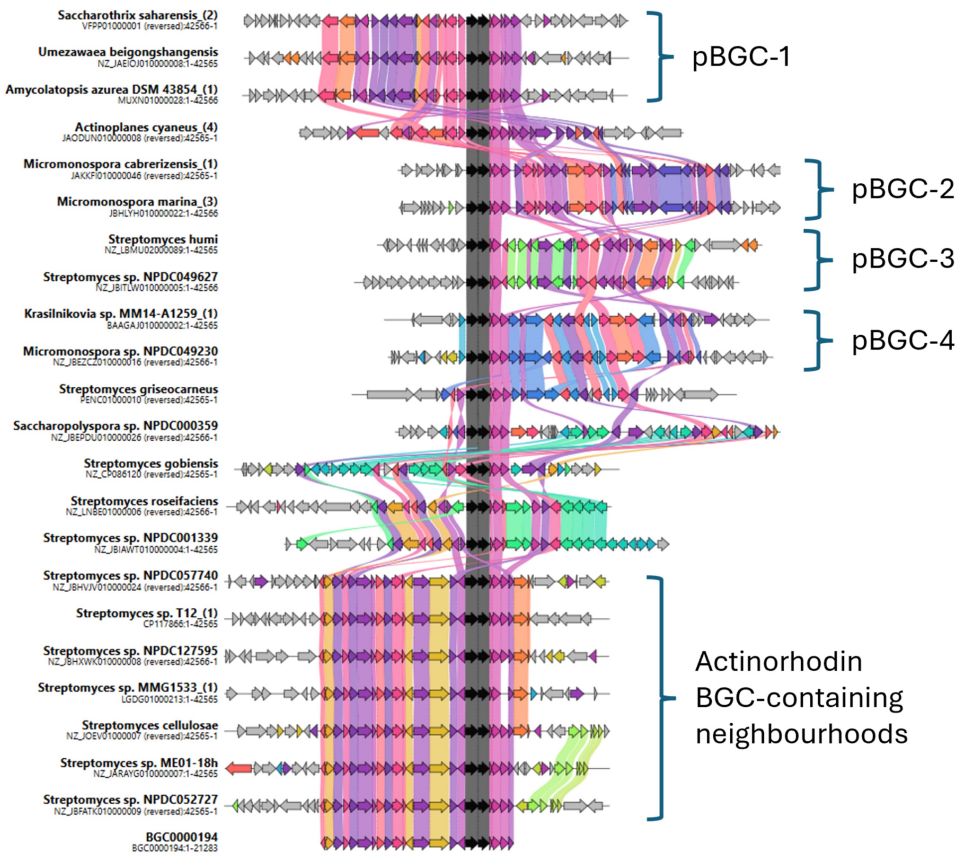
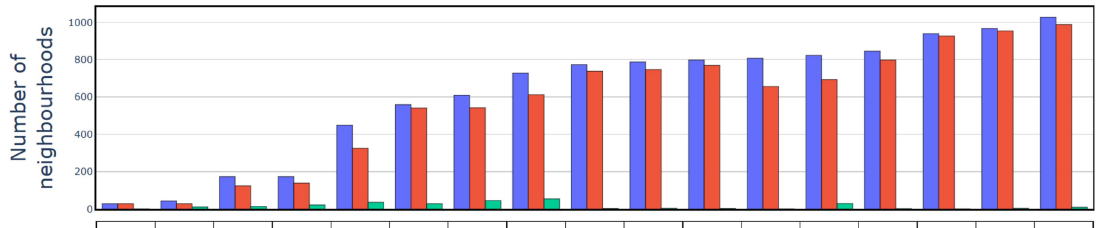


Figure 2. A synteny plot for the actinorhodin BGC. Each track represents a separate genomic neighbourhood. The bottom track shows the genes within actinorhodin MIBiG entry BGC000194, which is used as the BGC reference in this plot. Coloured links indicate genes encoding proteins which fall within the same homology group. Black links indicate genes that encode proteins within the same homology group as SCO5087 and SCO5088, the only genes annotated as core biosynthetic within MIBiG entry BGC0000194. These genes encode the actinorhodin polyketide beta-ketoacyl synthase alpha and beta subunits, respectively. The annotation 'pBGC' (putative BGC) highlights neighbourhoods in different strains and/or genera, which share a clear set of genes with conserved homology and synteny.



MIBiG Entry	#38	#914	#315	#194	#1063	#551	#325	#324	#940	#910	#2127	#663	#595	#2128	#660	#849	#1181
cblaster	28	43	173	173	449	559	609	728	773	788	798	808	823	845	939	967	1027
Collect	28	28	124	139	325	541	542	612	738	747	769	656	693	798	927	954	988
Sieve	1	11	13	22	36	28	45	54	3	4	3	1	29	2	1	4	9

Figure 3. SyntenyQC curation dramatically reduces the number of neighbourhoods associated with a cblaster homology search. 'MIBiG Entry' refers to *Streptomyces coelicolor* entries in the MIBiG database of experimentally characterized biosynthetic gene clusters (BGCs). The 'cblaster' values refer to the number of hit records identified by a cblaster search for homologs to a given BGC. 'Collect' refers to the number of neighbourhoods remaining after the SyntenyQC Collect subcommand is called on the cblaster records. 'Sieve' refers to the number of neighbourhoods that remain after the SyntenyQC Sieve subcommand is called on the collected neighbourhoods.

the *Streptomyces* genus. These pBGCs are conserved across multiple species and, in the case of pBGC-1 and pBGC-4, genera. Due to the cblaster search parameters (see Methods Section 1, available as supplementary data at *Bioinformatics* online), each pBGC also contained genes that fell within the same homology groups as the (core biosynthetic) actinorhodin polyketide beta-ketoacyl synthase alpha and beta subunits. This supports the hypothesis that these pBGCs are related to the actinorhodin BGC.

To illustrate the effectiveness with which SyntenyQC can remove redundant neighbourhoods, cluster homologs were identified and dereplicated for a test dataset of *Streptomyces coelicolor* BGCs extracted from the MIBiG database of validated BGCs (Zdouc *et al.* 2025) (see Methods Section 1, available as supplementary data at *Bioinformatics* online). The binary file from each cblaster run was then processed using the SyntenyQC Collect sub-command, with a neighbourhood length of twice the BGC length. This resulted in the loss

of multiple cblaster neighbourhoods that were small or dispersed beyond the specified neighbourhood length (Fig. 3 and Fig. 1B, available as supplementary data at *Bioinformatics* online). The folder of collected GenBank files was then processed with SyntenyQC Sieve, with a minimum similarity filter of 0.7 and the search strategy outlined in Methods Section 2, available as supplementary data at *Bioinformatics* online. This removed an average of 95% (nearest whole figure) of the neighbourhoods identified by cblaster (standard deviation 7%, nearest whole figure). The raw neighbourhood counts from cblaster, Collect and Sieve are shown in Fig. 3. See Table 1, available as supplementary data at *Bioinformatics* online and Fig. 7, available as supplementary data at *Bioinformatics* online for more information on SyntenyQC performance metrics and an exemplary analysis of Sieve information loss.

Finally, in Results Section 1, available as supplementary data at *Bioinformatics* online, we show that under default settings in the published case studies (De Vrieze *et al.* 2025), CAGEcleaner (i) removes diverse neighbourhoods, despite its hit recovery approaches and (ii) fails to remove many redundant neighbourhoods. We also highlight how high internal homology can hinder Sieve filtering in Results Section 2, available as supplementary data at *Bioinformatics* online.

4 Methods

4.1 Collect

The 'Collect' subcommand represents a direct integration point for cblaster (Gilchrist *et al.* 2021), which has become widely used to find putative BGC homologs and derivatives. 'Collect' is used to parse the cblaster results binary file, download neighbourhoods from NCBI (each of a user-defined span and centred on the corresponding cblaster-defined neighbourhood), and then write these neighbourhoods to a local file in GenBank format (Fig. 1A, available as supplementary data at *Bioinformatics* online). Files are named according to either organism or accession (as specified by the user), with the expectation that these filenames will be used to name the associated tracks by whichever synteny visualization software is used, as is done by clinker (Gilchrist and Chooi 2021). This subcommand thus facilitates the extraction of defined neighbourhoods from a cblaster results file and the exclusion of (i) neighbourhoods that are too small (e.g. fragmented contigs) and (ii) neighbourhoods that are too large (indicating cblaster has found hits that are not clustered within the defined neighbourhood span)—see Fig. 1B, available as supplementary data at *Bioinformatics* online.

4.2 Sieve

The 'Sieve' subcommand applies a graph-based algorithm to remove redundant neighbourhoods, in recognition of the fact that neighbourhood similarity is a non-transitive property—two neighbourhoods may be highly similar to each other, yet differ in terms of their relative similarity to a third neighbourhood (Fig. 2, available as supplementary data at *Bioinformatics* online).

First, all neighbourhoods undergo an all-vs-all BLASTP using DIAMOND (Buchfink *et al.* 2021) to identify reciprocal best hits between every pair of neighbourhoods. This is used to generate a similarity score for every pair of neighbourhoods, representing the proportion of genes in the smallest neighbourhood that have a reciprocal best hit within the larger neighbourhood (Fig. 3, available as supplementary data at *Bioinformatics* online). These scores then define edges in a graph with the neighbourhoods as nodes, which is pruned according to Algorithm 1, available as supplementary data at *Bioinformatics* online to remove neighbourhoods that exceed a user defined similarity (default 0.7) to another neighbourhood within the graph (Fig. 3, available as supplementary data at *Bioinformatics* online). Once the graph has been pruned, the surviving neighbourhoods are written to a results folder in GenBank format. Similarity filters above 0.7 typically generate more redundant plots, but large neighbourhoods extending far beyond core conserved regions (such as a BGC) may need lower similarity filters.

To help with optimization of the graph pruning parameters and illustrate a given neighbourhood space, the graph is also written to the results folder as an interactive html file (Fig. 4, available as supplementary data at *Bioinformatics* online), as is a histogram of all non-zero edge values present within the graph. The graph allows users to compare neighbourhood redundancy to host similarity by looking at the diversity of hosts within a given sub-network of the graph. The histogram indicates how changing the similarity filter might impact graph topology—if a similarity filter is 0.7, and the histogram shows most inter-neighbourhood similarities are close to 1, then increasing the similarity filter would not be expected to change the graph structure in a meaningful way (and using Collect to gather larger neighbourhoods that extend beyond conserved regions might be a better approach).

5 Conclusion

In this work, we introduced SyntenyQC, a tool for the preprocessing of genomic neighbourhoods prior to the construction of neighbourhood-scale synteny plots. We then showed how SyntenyQC can be used to collect and objectively reduce the number of neighbourhoods used to construct a given synteny plot, using a collection of verified biosynthetic gene clusters as a test dataset.

Author contributions

Timothy Kirkwood (Conceptualization [lead], Data curation [lead], Investigation [lead], Methodology [lead], Software [lead], Writing—original draft [lead], Writing—review & editing [equal]), Jack Connolly (Conceptualization [equal], Visualization [equal], Writing—review & editing [equal]), Ee Lui Ang (Funding acquisition [equal], Project administration [equal], Resources [equal], Supervision [equal], Writing—review & editing [equal]), Huimin Zhao (Funding acquisition [equal], Project administration [equal], Resources [equal], Supervision [equal], Writing—review & editing [equal]), Eriko Takano (Funding acquisition [equal], Project administration [equal], Resources [equal], Supervision [equal], Writing—original draft [equal], Writing—review & editing [equal]), and Rainer Breitling (Conceptualization [equal], Funding acquisition [equal], Project administration [equal], Resources [equal], Supervision [equal], Writing—original draft [equal], Writing—review & editing [equal])

Supplementary data

Supplementary Information is available at *Bioinformatics* online. Supplementary Data is available at https://github.com/Tim-Kirkwood/SyntenyQC_application_note

Conflict of interest: None declared.

Funding

This work was supported by the University of Manchester—A*STAR Research Attachment Programme (ARAP) to T.D.J. K.; U.S. National Institutes of Health [AI144967] and National Science Foundation [ChE-2019897, DBI-2400058, OISE-2435374] to H.Z.; National Research Foundation, Singapore [NRF-CRP19-2017-05-00] to E.L.A.; and UK Research and Innovation, Biotechnology and Biological Sciences Research Council UK-Japan Partnering Award [BB/X012573/1] to J.A.C., R.B., and E.T.

References

Amos GC, Awakawa T, Tuttle RN *et al.* Comparative transcriptomics as a guide to natural product discovery and biosynthetic gene cluster functionality. *Proc Natl Acad Sci USA* 2017;**114**:E11121–30.

Blin K, Shaw S, Medema MH *et al.* The antiSMASH database version 4: additional genomes and BGCs, new sequence-based searches and more. *Nucleic Acids Res* 2024;**52**:D586–9.

Blin K, Shaw S, Vader L *et al.* antiSMASH 8.0: extended gene cluster detection capabilities and analyses of chemistry, enzymology, and regulation. *Nucleic Acids Res* 2025;**53**:1–7.

Buchfink B, Reuter K, Drost HG. Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat Methods* 2021;**18**:366–8.

Cediel-Becerra JDD, Cumsille A, Guerra S *et al.* Targeted genome mining with GATOR-GC maps the evolutionary landscape of biosynthetic diversity. *Nucleic Acids Res* 2025;**53**:gkaf606.

De Vrieze L, Biltjes M, Lukashevich S *et al.* CAGEcleaner: reducing genomic redundancy in gene cluster mining. *Bioinformatics* 2025;**41**:btaf373.

Gilchrist CL, Booth TJ, Van Wersch B *et al.* cblaster: a remote search tool for rapid identification and visualization of homologous gene clusters. *Bioinform Adv* 2021;**1**:vbab016.

Gilchrist CL, Chooi YH. clinker & clustermap.js: automatic generation of gene cluster comparison figures. *Bioinformatics* 2021;**37**:2473–5.

Hackl T, Ankenbrand M, van Adrichem B *et al.* Gggenomes: effective and versatile visualizations for comparative genomics. arXiv, 2024, preprint: not peer reviewed.

Jensen PR. Natural products and the gene cluster revolution. *Trends Microbiol* 2016;**24**:968–77.

Medema HM, Takano E, Breitling R. Detecting sequence homology at the gene cluster level with MultiGeneBlast. *Mol Biol Evol* 2013;**30**:1218–23.

Navarro-Muñoz JC, Selem-Mojica N, Mullowney MW *et al.* A computational framework to explore large-scale biosynthetic diversity. *Nat Chem Biol* 2020;**16**:60–8.

Salamzade R, Tran PQ, Martin C *et al.* Zol and fai: large-Scale targeted detection and evolutionary investigation of gene clusters. *Nucleic Acids Res* 2025;**53**:gkaf045.

Sullivan MJ, Petty NK, Beatson SA. Easyfig: a genome comparison visualizer. *Bioinformatics* 2011;**27**:1009–10.

van den Belt M, Gilchrist C, Booth TJ *et al.* CAGECAT: the CompArative GEne cluster analysis toolbox for rapid search and visualisation of homologous gene clusters. *BMC Bioinformatics* 2023;**24**:181–8.

Zdouc MM, Blin K, Louwen NLL *et al.*; The MIBiG Consortium. MIBiG 4.0: advancing biosynthetic gene cluster curation through global collaboration. *Nucleic Acids Res* 2025;**53**:D678–90.

Zhang R, Mirdita M, Söding J. De novo discovery of conserved gene clusters in microbial genomes with spacedust. *Nat Methods* 2025;**22**:2065–73.

Bioinformatics, 2025, 41, 1–5
https://doi.org/10.1093/bioinformatics/btaf626
Applications Note