Synteny Plot quality control with SyntenyQC

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

[1] Manchester Institute of Biotechnology, Department of Chemistry, School of Natural Sciences, Faculty of Science and Engineering, University of Manchester, 131 Princess Street, Manchester, M1 7DN, UK

[2] Synthetic Biology Translational Research Program, Yong Loo Lin School of Medicine, National University of Singapore, 10 Medical Drive, 117597, Singapore ,

[3] Singapore Institute of Food and Biotechnology Innovation (SIFBI), Agency for Science, Technology and Research (A*STAR), 31 Biopolis Way, Nanos #04-01, 138669, Singapore

[4] Singapore Integrative Biosystems and Engineering Research (SIBER) Strategic Research Translational Thrust (SRTT), Agency for Science, Technology and Research (A*STAR), 2 Fusionopolis Way, Kinesis, #8-05, 138635, Singapore

[5] Bioinformatics Institute (BII), Agency for Science, Technology and Research (A*STAR), 30 Biopolis Street, #07-01 Matrix, 138671, Singapore

[6] Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, 600 South Mathews Avenue, Urbana, 61801, Illinois USA

[7] Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, 1206 W. Gregory Dr., 61801, Illinois USA

[8] NSF Molecular Maker Lab Institute, University of Illinois at Urbana-Champaign, 1206 W. Gregory Dr., 61801, Illinois USA

[9] NSF iBiofoundry, University of Illinois at Urbana-Champaign, 1206 W. Gregory Dr., 61801, Illinois USA

*Corresponding authors. timothy.kirkwood@manchester.ac.uk, Rainer_Breitling@bii.a-star.edu.sg

Contents

Supplementary Methods 1
S1: Collecting a BGC test data set and processing with SyntenyQC 1
Supplementary Figures 2
Supplementary Figure 1: The Collect subcommand. *2*
Supplementary Figure 2: Region similarity is non-transitive. *3*
Supplementary Figure 3: The Sieve subcommand. *3*
Supplementary Figure 4: The neighbourhood graph for actinorhodin MIBIG entry BGC000194. ..... *3*
Supplementary Algorithms 4
Algorithm 1 4
Supplementary Bibliography 4

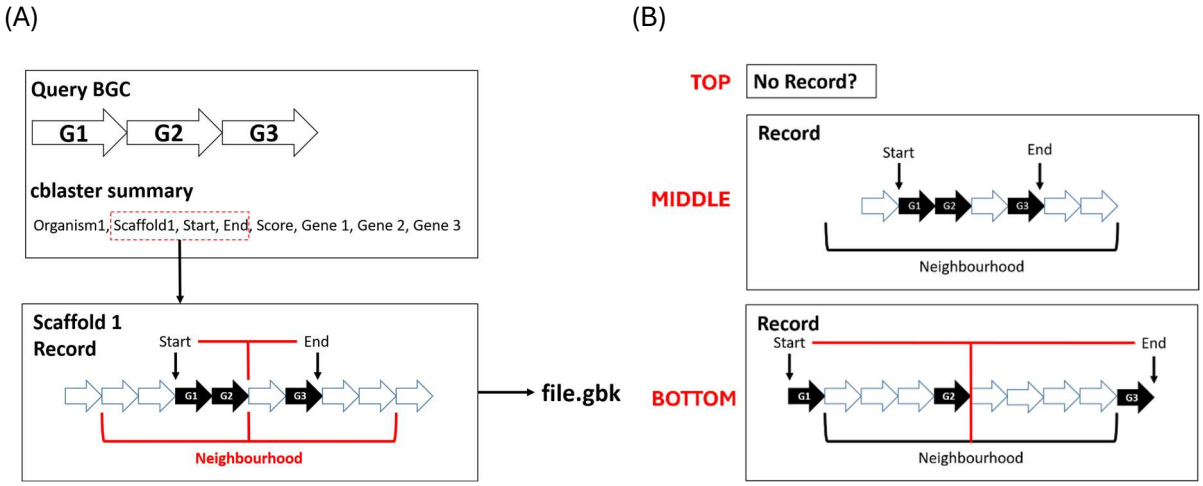
Supplementary Methods

S1: Collecting a BGC test data set and processing with SyntenyQC

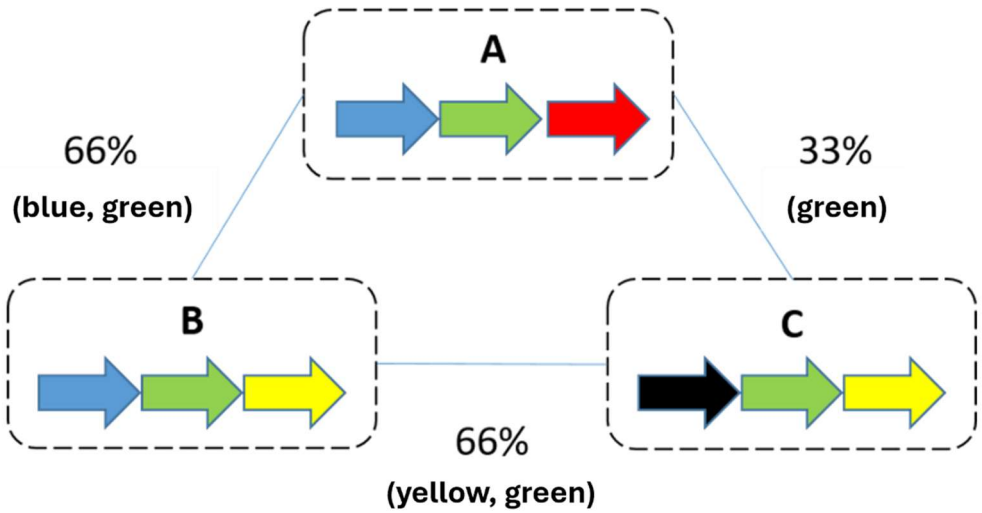
The MIBIG database (Zdouc et al., 2025) of verified Biosynthetic Gene Clusters (BGCs) was searched for entries from *Streptomyces coelicolor*. Each BGC protein set was used as a query in a cblaster (Gilchrist et al., 2021) search to identify putative BGC homologs. All hit neighbourhoods were required to have N unique hits to the BGC query, where N was either 5 or half the number of proteins in the query (whichever was larger), and all core biosynthetic proteins were required. For queries with 5 or fewer proteins, N was the number of proteins encoded by the BGC, and no specific proteins were required. Hits were restricted to those within Actinomycete genomes.

The commands used for cblaster, SyntenyQC Collect, and SyntenyQC Sieve are given in **Supplementary Data 1**. The fasta-format query files used for cblaster are given in **Supplementary Data 2**. The binary files output by cblaster are available at **Supplementary Data 3**. The GenBank files generated by SyntenyQC Collect and Sieve are available upon request. All supplementary files are available at Tim-Kirkwood/SyntenyQC_application_note.

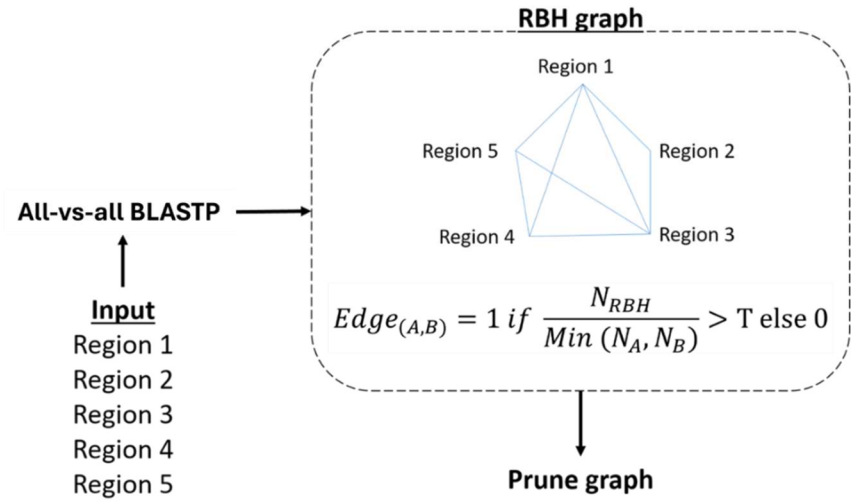
Supplementary Figures



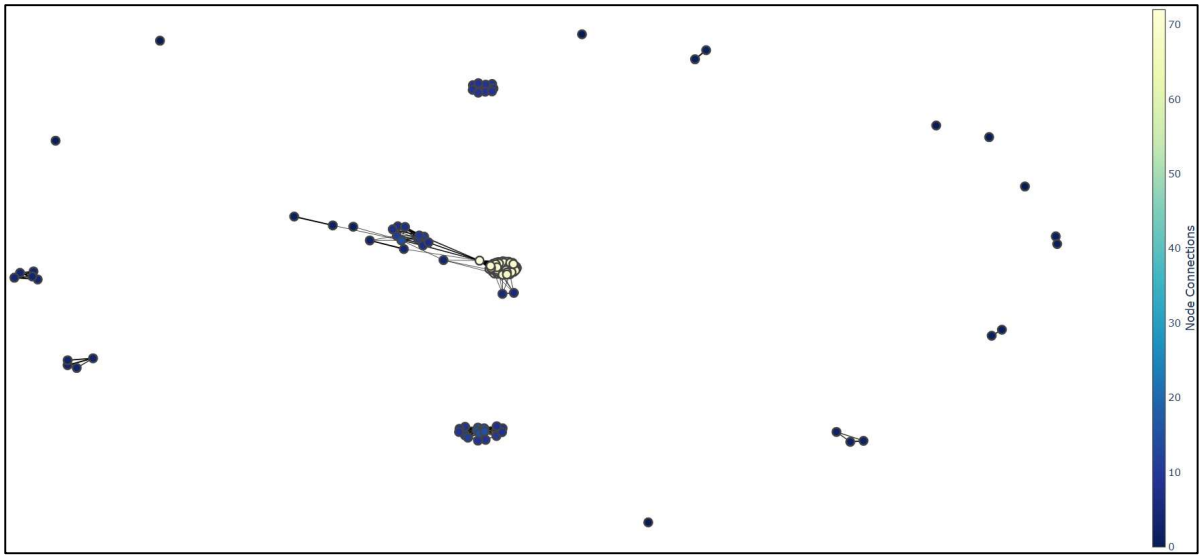
Supplementary Figure 1: The Collect subcommand. (A) The workflow. A neighbourhood is downloaded from NCBI using the accession number supplied in the cblaster result file. The loci of the first (Gene 1) and last (Gene 3) cluster gene homologs in this neighbourhood are used to define a neighbourhood of a user-specified size, with a mid-point that lies between the loci. This neighbourhood is then written to a local GenBank file. **(B) Rejected records.** The neighbourhood is rejected if (TOP) its accession is not recognised, (MIDDLE) making a neighbourhood of a user-specified size would involve extending the neighbourhood beyond a contig edge – note this is optional, or (BOTTOM) if the hits identified by cblaster do not fall within a neighbourhood of user-specified size.



Supplementary Figure 2: Region similarity is non-transitive. Boxes indicate neighbourhoods, arrows indicate genes, colours indicate homolog groups. A/B and B/C are fairly similar in terms of homolog composition (66%), but A and C are much less similar (33%).



Supplementary Figure 3: The Sieve subcommand. A collection of input regions are subjected to an all-vs-all BLASTP to identify the number of reciprocal best hits (RBHs) between each pair of input regions. The neighbourhoods are then represented as a graph, where an edge is drawn between two nodes if their respective regions have a proportion of RBHs that exceed a user-defined threshold. N_{RBH} is the number of reciprocal best hits between two node regions (A and B), N_A and N_B are the number of proteins in regions A and B respectively. T is a threshold proportion that is set by the user (typically 0.5 to 0.7). This graph is then pruned according to Supplementary Algorithm 1. Following pruning, the remaining neighbourhoods are returned to the user.



Supplementary Figure 4: The neighbourhood graph for actinorhodin MIBIG entry BGC000194. Shown is the neighbourhood graph described in Supplementary Figure 5,

with neighbourhoods as nodes and edges indicating that two neighbourhoods have a similarity score that exceeds the user threshold. Prior to pruning via Supplementary Algorithm 1, the neighbourhood graph is written to a dynamic HTML file, with hover labels describing the details associated with each node and edge. Node colour indicates the degree of a given node, and edge thickness indicates the similarity score magnitude.

Supplementary Algorithms

Algorithm 1

Data: RBH graph

Result: Nodes from pruned RBH graph

Procedure:

```
while max(node degrees in RBH graph) > 0:
    delete nodes = []
    for node in RBH graph:
        if node degree = max(node degrees in RBH graph):
            delete nodes + node
    delete node = random node from delete nodes
    RBH graph = RBH graph - delete node
return nodes in RBH graph
```

Supplementary Bibliography

Gilchrist, C. L. M., Booth, T. J., Van Wersch, B., Van Grieken, L., Medema, M. H., & Chooi, Y. H. (2021). cblaster: a remote search tool for rapid identification and visualization of homologous gene clusters. *Bioinformatics Advances*, *1*(1). https://doi.org/10.1093/BIOADV/VBAB016

Zdouc, M. M., Blin, K., Louwen, N. L., Navarro, J., Loureiro, C., Bader, C. D., Bailey, C. B., Barra, L., Booth, T. J., Bozhueyuek, K. A., Cediel-Becerra, J. D., Charlop-Powers, Z., Chevrette, M. G., Chooi, Y. H., D'Agostino, P. M., de Rond, T., Pup, E. D., Duncan, K. R., Gu, W., … Dillen, J. (2025). MIBiG 4.0: advancing biosynthetic gene cluster curation through global collaboration. *Nucleic Acids Research*, *53*(D1), 678–690. https://doi.org/10.1093/nar/gkae1115