

## Supplementary Text and Figures

Salamzade *et al.* 2022

### Contents

Automated workflows for running <i>IsaBGC</i> .....	2
Updates to <i>IsaBGC</i> since release 1.0.....	2
Assessment of hybrid genomic assemblies constructed for <i>S. epidermidis</i> LK1136 and <i>S. warneri</i> LK413 .....	2
Designation of species belonging to <i>Mammaliicoccus</i> .....	3
Genome annotation, homology determination in predicted proteomes, and clustering of BGCs into GCFs.....	3
High throughput identification of homologous instances of GCFs in assemblies .....	4
Visualization of BGCs across phylogenies.....	8
Inspection of <i>Staphylococcus</i> and <i>Corynebacterium</i> GCFs with high Beta-RD values.....	8
Ancestral inference of gene cluster family carriage.....	9
Identification of homolog groups associated with skin-residing species of <i>Staphylococcus</i> and <i>Corynebacterium</i> .....	9
Understanding GCF conservation and composition through evolutionary and population genetic statistics.....	9
Inference of consensus order and directionality of homolog groups for a GCF .....	10
Metagenomic mining for BGCs, homolog groups, and novel SNVs.....	10
Benchmarking <i>IsaBGC</i> -DiscoVary against assembly-based novel variant detection using <i>M. luteus</i> single-isolate sequencing readsets.....	13
Scrutinization of putative novel variants for <i>IsaBGC</i> -DiscoVary application to cutimycin in <i>C. acnes</i> and the comprehensive set of GCFs for the <i>C. tuberculostearicum</i> species complex .....	14
Determination of whether novel SNVs detected for BGCs of the <i>C. tuberculostearicum</i> species complex would be identified in metagenomic assemblies: .....	15
<i>IsaBGC</i> -DiscoVary based analyses of BGCs from the <i>C. tuberculostearicum</i> species complex: .....	15
Supplementary Text References.....	17
Supplementary Figures .....	22

## Automated workflows for running *lsaBGC*

To automate the application of these programs, we had initially developed three workflow programs called *lsaBGC-AutoProcess.py*, *lsaBGC-AutoExpansion.py*, and *lsaBGC-AutoAnalyze.py*. *lsaBGC-AutoProcess.py* is a preliminary workflow which does not execute any of the aforementioned core programs of the suite. It functions to take in a listing of genomic assemblies and perform gene-calling and basic annotation using Prokka(1), annotate BGCs using antiSMASH(2), and finally run OrthoFinder2(3) for *de novo* delineation of homolog groups. It runs antiSMASH and OrthoFinder2 using the same Prokka output to ensure locus tag identifiers are able to be matched between the output of the two former programs (Fig. S1a). *lsaBGC-AutoExpansion.py* is a wrapper of *lsaBGC-Expansion.py* which performs additional GCF instance identification in draft assemblies across all GCFs identified for a taxa. It is recommended that users run *lsaBGC-AutoExpansion.py* instead of *lsaBGC-Expansion.py* individually because the workflow additionally features a critical consolidation step in which BGC instances identified as potentially belonging to multiple GCFs are re-assessed and assigned to only the single best fitting GCF (Fig. S1c). Finally, *lsaBGC-AutoAnalyze.py* is a major workflow which runs the analytical core programs across each GCF and creates a few consolidated reports after completion. This workflow starts by computing pairwise whole-genome similarity metrics using either CompareM(4) or FastANI(5) and then runs *lsaBGC-Sec.py*, *lsaBGC-PopGene.py*, *lsaBGC-RelativeDivergence.py* and, optionally, *lsaBGC-DiscoVary.py* for each GCF. At the end of the workflow, it generates consolidated report tables and visualizations from GCF-specific results for *lsaBGC-PopGene.py* and *lsaBGC-RelativeDivergence.py*. These workflows, similar to the core programs, are documented on the Github wiki.

## Updates to *lsaBGC* since release 1.0

Changes to *lsaBGC* since its initial release have primarily focused on simplifying usage of the suite through development of *lsaBGC-Ready.py* and *lsaBGC-Easy.py*. Analytical changes have mostly been minimal but include: (i) updated formatting of result files from the *lsaBGC-AutoAnalyze.py* workflow, (ii) changing how gaps are accounted for in calculating the Beta-RD statistic from codon-alignments, (iii) an adjustment to key-word searches used to prevent *lsaBGC-DiscoVary* from calling variants upon genes for which annotation suggests are MGEs, (iv) using MAGUS(6) in place of MAFFT(7) for protein alignment, to allow for better scalability, (v) introducing a more stringent requirement for homolog groups regarded as part of the protcore of BGC predictions by antiSMASH(2), and (vi) use of hierarchical orthogrouping by OrthoFinder. Incorporation of GToTree(8) also provided the opportunity to efficiently and more easily estimate expected similarities between genomes using protein alignments of single copy genes used by the software to construct phylogenies. As such, we have removed support for inferring Beta-RD using ANI or AAI between genomes. We have also added support for GECCO(9) and DeepBGC(10) predictions of BGCs. Finally, we have begun to introduce code to enable application of *lsaBGC* to fungi and plants but are still testing these functionalities.

## Assessment of hybrid genomic assemblies constructed for *S. epidermidis* LK1136 and *S. warneri* LK413

We chose to investigate staphyloxanthin production in *S. epidermidis* and *S. warneri* because they are *Staphylococcus* species commonly isolated from skin. Specifically, the former species *S. epidermidis*, is the most abundant staphylococcal species on skin and the latter species, *S. warneri*, featured two distinct staphyloxanthin encoding GCFs. For the *S. epidermidis* LK1136 hybrid ONT and Illumina genome assembly, we found that the isolate featured eight plasmids, including a 225 kb mega-plasmid (Table S7b). All eight plasmids were regarded as circularized; however, the chromosome was not. For the *S. warneri* LK413 hybrid genome assembly, we found that the isolate featured three plasmids, of which two

were circularized (Table S7b). The chromosome for this isolate was circularized and regarded as complete. The incomplete scaffold in *S. warneri* was found to feature staphyloxanthin encoding GCF-6 and to validate it represented a plasmid, we used the BLASTn(11) to NCBI's nt analysis performed in GAEMR(<https://software.broadinstitute.org/software/gaemr/>) to confirm that all reported alignments were to *Staphylococcus* plasmid sequences. Additionally, this scaffold featured a slightly elevated coverage, ~3X greater than the chromosome.

## Designation of species belonging to *Mammaliicoccus*

The *Staphylococcus*\_A genus designation in GTDB R202 was largely concordant with the recent reclassification of five species from *Staphylococcus* to *Mammaliicoccus*(12). In the latest GTDB R207 release, this genus has now been reclassified to *Mammaliicoccus*. The recent update to taxonomic names also includes the reclassification of species *S. pasteurii*\_A to *M. fleurettii*. Notably, we do not regard *S. schleiferi* as part of *Mammaliicoccus* because it is categorized as *Staphylococcus* by GTDB (both R202 and R207) and groups within the *Staphylococcus* genus in our ribosomal protein-based maximum-likelihood phylogeny. Similar to our assessment, the reclassification of this species to *Mammaliicoccus* was recently questioned(13).

## Genome annotation, homology determination in predicted proteomes, and clustering of BGCs into GCFs

Complete or chromosome level assemblies were run through `IsaBGC-AutoProcess.py` which invokes Prokka(1) for gene calling and annotation, antiSMASH for BGC detection (v6)(2), and OrthoFinder2(3) for delineating homologous clusters of proteins (Fig. S1a). Since version 1.1, we no longer recommend usage of `IsaBGC-AutoProcess.py` and instead suggest using the `IsaBGC-Easy.py` workflow or `IsaBGC-Ready.py`. `IsaBGC-Ready.py` can perform many of the same functions as `IsaBGC-AutoProcess.py` with expanded options to: (i) automatically perform clustering of BGCs into GCFs using `IsaBGC-Cluster.py` or to incorporate user-provided BiG-SCAPE(14) clustering results, (ii) identify additional instances of GCFs directly in draft quality assemblies using `IsaBGC-AutoExpansion.py`, and (iii) construct a species phylogeny using `GToTree`(8).

Subsequent clustering of detected BGCs into GCFs was performed using `IsaBGC-Cluster.py`. For consistency and simplicity, we ran both the genus-level and species-level analyses using identical parameters for `IsaBGC-Cluster.py`. The MCL inflation parameter was set to 4.0, the minimal syntenic similarity threshold was set to 0.7, and the Jaccard similarity of homolog groups shared vs. observed in union between two BGCs threshold was set to 20.

`IsaBGC-Cluster.py` leverages homology information from OrthoFinder2(3) together with syntenic similarity akin to the approach taken in BiG-SCAPE(14). However, unlike BiG-SCAPE, which is reliant on domains, `IsaBGC-Cluster.py` uses entire protein sequences designated to discrete homolog groups, offers a tailored report for users on appropriate parameter selection, and employs Markov chain clustering (MCL)(15) instead of affinity propagation for granular GCF delineation. Similar to BiG-SCAPE, `IsaBGC-Cluster.py` also uses both the presence of coding units (domains for BiG-SCAPE; homolog groups for `IsaBGC-Cluster.py`) as well as sequence similarity between such units to appropriately cluster BGCs. In `IsaBGC-Cluster.py`, the presence of homolog groups and their sequence similarity is implicit to the OrthoFinder2 algorithm which determines appropriate thresholds for designating homologous, ideally orthologous, protein instances by accounting for genome-wide similarity(3). Also, similar to BiG-SCAPE in concept(14), `IsaBGC-Cluster.py` can be set to require syntenic similarity between BGC instances for clustering, but the methodology in `IsaBGC-Cluster.py` is based on global syntenic similarity measured using the absolute value of the Spearman correlation for sets of three homolog groups shared between BGC instances. BiG-SCAPE on the contrary uses more localized information on adjacent-pairs of

domains to infer overall syntenic similarity. Finally, both `lsaBGC-Cluster.py` and `BiG-SCAPE` offer flexibility for whether to perform preliminary partitioning of BGCs based on the predicted class by `antiSMASH`. By default, `lsaBGC-Cluster.py` does not perform such partitioning.

Unique to `lsaBGC-Cluster.py` is an option to produce a user-friendly PDF report guiding users for optimal parameters for clustering. This report showcases how different values of the MCL inflation parameter and the Jaccard similarity threshold for shared homologs between pairs of BGCs affects the final clustering. In this report, users can see how different parameter combinations can influence the number of singleton GCFs, GCFs with a single BGC member, the number of the core homolog groups observed in all BGCs belonging to a particular GCF (Table S2, S3). Such a report is critical for users to be able to move forward with an appropriate delineation of GCFs and not to re-track and manually experiment with different clustering configurations. An example report, pertaining to clustering BGCs from 77 *Staphylococcus* genomes with completed or chromosome-level assemblies, as well as descriptions for each of the figures featured in its 52 pages can be found on the Github wiki page for the `lsaBGC-Cluster.py` program.

To additionally provide users with versatility to better define the boundaries of a GCF, which could be particularly useful when dealing with hybrid BGCs or multiple BGCs co-located nearby each other, we further provide `lsaBGC-Refiner.py`. `lsaBGC-Refiner.py` takes in as input a GCF listing file of BGC instances belonging to it and a pair of user-defined boundary homolog groups. It then filters BGC Genbanks to retain only genes found in between the two boundary homolog groups. This functionality is particularly useful for instances where `antiSMASH` is inconsistent in defining hybrid BGCs due to variable inter-protocore content which has recently been highlighted(16).

For biosynthetic class designations of GCFs in this study, we regarded GCFs as a single class if at least 90% of BGC instances from complete genomes were predicted by `antiSMASH` as encoding for the class; otherwise, the GCF was regarded as a hybrid.

## High throughput identification of homologous instances of GCFs in assemblies

To mine metagenomic datasets for base-resolution novelty within BGCs previously unobserved from available assemblies for a given taxa, it is first necessary to comprehensively profile all allelic variants of a homolog group from such a set of assemblies. We thus developed `lsaBGC-Expansion.py` to efficiently and systematically identify orthologous instances of GCFs across the comprehensive set of assemblies available for a taxa, which for certain *Staphylococcus* species could range in the thousands on NCBI's GenBank database (Fig. 1cd, S1b). In concept, `lsaBGC-Expansion.py` is similar to the now common bioinformatics practice of defining homolog groups for proteins upfront and then searching for their presence in new genomes(17–19). In fact, re-identifying instances of homolog groups found within GCFs is the first component of `lsaBGC-Expansion.py`. This is performed by first constructing profile HMMs for such homolog groups using MAFFT local alignment(7) with standard settings and HMMER3(20), followed by searching genomic assemblies directly using HMMER3 or emitting the consensus sequence and searching via DIAMOND(21), as inspired by the approach developed by Melnyk et al. 2019(18). For the analyses presented in this paper, we used the DIAMOND-based approach (specified by `--quick_mode` argument).

In a preliminary step, profile HMMs are similarly aligned to the predicted proteome of genomes from the initial set of genomes which were run through `antiSMASH` and used to establish GCFs. This reflexive alignment enables the determination of appropriate E-value thresholds to gauge the presence or absence of homolog groups in new genomes. Specifically, the E-value thresholds for each homolog group is determined as either: (i) the lowest E-value of false positive alignments multiplied by a factor of 1E-5 or (ii) a default threshold of 1E-10 if E-values for some true positive alignments are found to be higher than those for false positive alignments. Importantly, this methodology for selecting E-value thresholds assumes that new genomes being searched are phylogenetically interspersed with genomes used for initial

analysis and construction of the profile HMMs. We additionally required proteins which aligned to homolog group profile HMMs to also be of similar length to the proteins used to construct the profile HMMs, requiring them to be at most 1.5X the max(25, median absolute divergence) base-pairs shorter or longer than the median length of the representative proteins of the homolog group.

A classical HMM framework, leveraging the pomegranate library(22), is used to scan predicted coding genes across assembly scaffolds, where coding genes are regarded to follow a binary state based on whether they exhibit homology to a GCF-associated homolog group. Our approach mimics the algorithm of ClusterFinder(23), but features several important differences to identify fragmented instances matching well-characterized BGCs rather than search for novel BGCs with remote homology to known instances. We estimate the emission probability of each homolog group based on whether its profile HMM is able to distinguish true hits from false hits from reflexive alignments (see previous paragraph). If it is able to distinguish true-hits from false-hits, then detecting a homolog group at the E-value threshold described above is very likely to represent a predicted protein belongs to the GCF and the homolog group will have an emission probability of 0.99 for the "GCF State" and a emission probability of 0.01 for the "Background State". If the profile HMM is unable to distinguish true-hits from false-hits, then the emission probability for the "Background State" is set to a maximum of either: (i)  $1.0 - (\# \text{ of GCF instances} / \# \text{ of total instances})$  or (ii) 0.2. The emission probability for the "GCF State" in this case will be set to the complement of the "Background State" probability. The transition probabilities between states are set to 0.1 by default and to 0.9 for transitioning to the same state. The start and end of a scaffold are equally likely to correspond to either the states of GCF or Background.

Genomic neighborhoods of predicted coding genes detected by the HMM as corresponding to the GCF (herein referred to as "potential GCF segments") are next conditionally assessed to avoid reporting false positive neighborhoods which are likely not related to the GCF in question. Each potential GCF segment must feature at least 3 homolog groups found in a known instance of the GCF. Additionally, all potential GCF segments must display reasonable syntenic similarity (gene positioning) to a known instance of the GCF found in the high quality genomic assemblies. Unlike the syntenic similarity filtering applied in the clustering of BGCs into GCFs, here we use Pearson's correlation instead of Spearman's correlation to more reliably assess syntenic similarity of shorter segments to known GCF instances. Correlations are only calculated if the genes in the segment under consideration and the comparing known BGC instance have genes in the same relative direction. A default correlation of 0.8 to at least one known GCF instance is required by each segment (with p-value < 0.1).

Segments are reported as part of the GCF automatically if any of the following criteria are met: (i) The segment features  $\geq 5$  homolog groups and segment features  $\geq 3$  "core" homolog groups ("core" homolog groups are those which are observed in all known instances of a GCF from the high-quality genomic assemblies), (ii) the segment features a GCF "specific" homolog group (these are homolog groups which are only observed within the GCF in the high-quality genome assemblies, thus their presence is a good indication the segment belongs to the GCF and is not a false positive), or (iii) the segment features homolog groups which overlap with the core of protocusters in GCFs as delineated by antiSMASH. If segments do not meet any of the three criteria above, they can still be reported if they are on the edge of scaffolds (within 500 bp from the end of a scaffold). These segments do not need to meet criteria (i) individually, but do need to feature  $\geq 5$  homolog groups and  $\geq 3$  core homolog groups of the GCF in unison. Additionally, only one edge segment per scaffold is allowed.

To ensure we are not missing more evolutionarily diverged variants of homolog groups not represented in the initial lsaBGC analysis, we perform a final "polishing" step for segments which are considered to belong to a GCF. This involves reassessing genes on GCF segments which are unassigned to any homolog group and additionally searching for surrounding genes (up to 10 genes on each side) which display homology to GCF-associated homolog groups. A gene only needs to display homology at a level of less than  $1E-10$  E-value to be assigned to a homolog group. Critically, we only assign these genes to

homolog groups after the segment is determined as belonging to the focal GCF to ensure our detection of segments is based on more concrete/significant homology-based evidence

Ultimately, we do not recommend running `IsaBGC-Expansion.py` individually on a GCF-by-GCF basis, but to instead use the wrapper workflow program `IsaBGC-AutoExpansion.py`, to automatically search for homologous instances across all GCFs. This is because `IsaBGC-AutoExpansion.py` is able to resolve potential conflicts where some part of a genomic assembly might independently be assigned to two separate GCFs (Fig. S1c). After running `IsaBGC-Expansion.py` individually per GCF, it consolidates results and resolves any conflict in genes from the same genomic assembly being assigned to multiple GCFs. It resolves such conflicts by performing pairwise comparisons of the gene sets across every BGC instance from every GCF. Overlap between two BGCs from distinct GCFs is considered a conflict if they overlap with more than 5% of the number genes of one of the BGCs. If this is the case, the sum of the exponents of E-values of each BGC's genes to their respective homolog group profile HMMs are compared and the BGC with the lower sum (indicating more genes in the BGC match a GCF profile or that genes match better to the GCF profile) is retained while the other is discarded. `IsaBGC-AutoExpansion.py` also creates consolidated result files, including: (i) an updated/expanded sample vs. homolog group gene matrix, (ii) an updated/expanded sample listings file (listing the path to each sample's GenBank assembly and predicted proteome), and (iii) an updated listing file of BGCs for each GCF.

*Rapid determination of 63 GCFs in >15K Staphylococcus genomes:* We were able to use `IsaBGC-AutoExpansion.py` to rapidly identify GCFs determined from complete *Staphylococcus* genomes in all ~15K *Staphylococcus* genomes represented in GTDB release R202. As expected, we found that fewer homolog groups in BGCs were detected for low quality assemblies ( $N50 < 10K$ ; median of 105 homolog groups in BGCs per genome) as compared to high-quality assemblies ( $N50 > 100K$ ; median of 164 homolog groups in BGCs per genome) ( $p=8.58E-36$ ; two-sided Wilcoxon rank sum test). Furthermore, we found a lower percentage of homolog groups found together on the same BGC fragment with core biosynthetic machinery for low quality assemblies ( $N50 < 10K$ ; median of 54%) as compared to high-quality assemblies ( $N50 > 100K$ ; median of 100%) ( $p=8.64E-39$ ; two-sided Wilcoxon rank sum test) (Fig. 1c). Note, the time taken to run `IsaBGC-AutoExpansion.py` on all draft-quality *Staphylococcus* genomes described in the main text is not inclusive of preliminary gene-calling using Prokka(1) performed via `IsaBGC-AutoProcess.py`.

*M. luteus - Benchmarking comprehensive antiSMASH vs. IsaBGC framework:* To benchmark the sensitivity and specificity of the *IsaBGC* framework, whereby we perform antiSMASH(2) based identification on a subset of complete genomes for a taxa and then identify homologous instances in draft quality genomes, against simply running antiSMASH on all the available genomes for the taxa, we performed two comparative analyses using *M. luteus* genomes (Fig. 1c, S3, S4). Because `IsaBGC-Expansion.py` accounts for all homolog groups found in a GCF rather than being dependent on a smaller subset of BGC-associated domains, `IsaBGC-Expansion.py` should have increased sensitivity for detection of genomic regions belonging to a GCF compared to antiSMASH. For instance, if in truth a BGC exists across two separate scaffolds, but only one segment features all the core domains encoding for the secondary metabolite biosynthesis machinery, then a rule-based approach seeking specific domains might struggle with detection of the second segment consisting of only auxiliary cargo, even if some of those genes encode enzymes critical for the production of the final metabolite.

In the first experiment, to assess the sensitivity of `IsaBGC-Expansion.py`, we compared the final GCFs of two different *IsaBGC* analyses: (i) an expansion based analysis where we ran initial processing, antiSMASH based BGC identification, and GCF clustering for 14 high-quality "complete" / "chromosome" quality *M. luteus* assemblies and then perform expansion of the GCFs with 213 additional *M. luteus* draft genomic assemblies of lower quality, and (ii) a comprehensive clustering analysis where we ran initial processing, antiSMASH based BGC identification, and GCF clustering across all 227 genomic assemblies in consideration (Fig. S3a).

This comparative analysis essentially allowed us to assess how many instances (in this case contiguous segments) of a particular GCF are missed by the *lsaBGC* expansion-based framework (Fig. S3b). We found that almost all instances of BGCs found when running antiSMASH on the draft assemblies directly, were also identified through *lsaBGC-AutoExpansion.py* using BGCs from the 14 high-quality genomic assemblies as references (97.8%; 1206/1233). Nearly all the exceptions not detected in the expansion-based analysis (88.9%; 24 of 27) corresponded to rare GCFs which had no representatives in the high-quality assemblies. These cases demonstrate a significant limitation of the *lsaBGC* core framework in which BGCs are identified *de novo* in only a subset of genomic assemblies and is an important consideration for users.

The analysis also allowed us to assess how many instances of a particular GCF were only found by *lsaBGC-AutoExpansion.py* and not by running antiSMASH directly on draft genomes (Fig. S3c). Given that our approach is a taxa-focused analysis and utilizes all homolog groups associated with a GCF, not just those containing core BGC domains, it is not surprising that we were able to find substantially more instances of BGC segments in the draft assemblies using *lsaBGC-AutoExpansion.py* which were missed by antiSMASH. This is likely because of assembly fragmentation, as described in the example scenario above. In total, we found 689 new GCF instances only through *lsaBGC-Expansion*. Of these, most (87.1%, 600 of 689) correspond to true "expansions" (the GCF was detected as present in the samples in the comprehensive clustering analysis using antiSMASH, but additional segments containing auxiliary content on different scaffolds of the assembly were only detected by *lsaBGC-AutoExpansion.py*).

In the second benchmarking experiment, we again use the 14 completed *M. luteus* genomes available, but in this setup our aim was to compare how the *lsaBGC* approach compared to running *de novo* antiSMASH when applied to fragmented versions of the original genomic assemblies (Fig. S4a). To generate the fragmented assemblies, paired-end reads were first simulated from assemblies using ART(24) at 5X coverage and then re-assembled using Unicycler(25). Five fragmented assemblies were created for each of the original 14 completed *M. luteus* genomes. On average fragmented assemblies had an N50 of 13,935 bp. This benchmarking setup allowed us to assess how many coding genes initially identified as part of distinct GCFs on complete genomes are re-identified after genomes are artificially fragmented. We performed *in silico* simulation of assembly fragmentation for each of the 14 genomes in five replicates. For each such replicate simulation, we randomly fragmented genomic assemblies at a single point within each BGC predicted for it by antiSMASH. This resulted in 70 fragmented assemblies (5 for each of the 14 genomes), which were then searched *de novo* for BGCs using antiSMASH or searched based on similarity to the initial BGCs identified in the completed (unfragmented) genomes using *lsaBGC-AutoExpansion.py*. Coding genes were matched between the initial completed genomes with their fragmented versions based on exact sequence matching and also checking if ORFs from the fragmented genome were subsequences of ORFs from the complete genomes.

As expected, based on the algorithms employed, we found that *lsaBGC-AutoExpansion.py* was able to re-identify more of the initial coding genes associated with BGCs in complete genomes compared to rerunning antiSMASH *de novo* (Fig. 1c, S4b). Rerunning antiSMASH on fragmented assemblies detected only 45.3% on average of the proteins it had originally associated with BGCs in complete genomes, while *lsaBGC-AutoExpansion.py* was able to recover 77.1% of them on average. However, *lsaBGC-AutoExpansion.py* did identify multiple instances of homolog groups which were not present in the initial BGC predictions on complete genomes (Fig. S1c, S4bc). This is because *lsaBGC-AutoExpansion.py* used as reference all BGC instances from across the 14 genomes to identify GCFs in the fragmented genomes and thus increased the boundary of GCFs beyond what they might have initially been set to by the initial antiSMASH run on the complete (unfragmented) genomes. This can be seen as another key advantage of *lsaBGC-AutoExpansion.py*, in that it allows smoothing of BGC boundaries across homologous instances upon identification and can enable more comprehensive and robust comparative genomics of GCFs downstream.

## Visualization of BGCs across phylogenies

We have provided the *lsaBGC-See.py* program to allow users to visualize GCF instances across either a user-provided species phylogeny or automatically generated GCF-specific phylogeny. While the latter functionality is present in CORASON(14), the former feature, which can provide a differing and important evolutionary perspective, is currently unique to *lsaBGC*. *lsaBGC* creates automated PDF reports using R visualization libraries *ggplot2*(26) and *ggtree*(27) as well as creates a track for visualization in iTol(28). Additionally, the constructed GCF phylogeny and user-provided species phylogeny are reformatted to a Newick file in which genome identifiers are expanded if multiple instances of a GCF are found for it, which is critical for visualizing fragmented BGCs from draft genomes. The program generates a GCF specific phylogeny if requested by creating codon alignments (as described in the section “Understanding GCF Conservation and Composition through Evolutionary and Population Genetic Statistics”), concatenating these, filtering for conserved SNVs and generating an approximate maximum-likelihood phylogeny using *FastTree2*(29).

Since version 1.3, we also provide the largely standalone program *GSeeF.py* within the *lsaBGC* suite which allows for construction of species phylogenies using *GToTree*(8) and phylogenomic heatmaps showcasing the presence of GCFs, defined by *lsaBGC-Cluster.py* or *BiG-SCAPE*, with the coloring of the heatmap corresponding to annotation types of BGCs extracted from antiSMASH BGC GenBanks. *GseeF.py* can be used to create similar views such as what we have depicted in Figure 1B.

For generating comparative views between different *crt* encoding GCF representatives (Fig. S9a) we used *clinker*(30). For visualization of staphyloxanthin encoding GCFs across the *Staphylococcus* phylogeny (Fig. 3a, S8b), only a single representative genome from each species, as classified by GTDB(31), was selected and used to prune the original ribosomal phylogeny created from the diversity representation genome set. The percentage of species members with GCFs was determined using *lsaBGC-AutoExpansion.py* results for all ~15K *Staphylococcus* genomes and was shown in log<sub>10</sub> scale.

## Inspection of *Staphylococcus* and *Corynebacterium* GCFs with high Beta-RD values

Investigation into High Beta-RD of lugdunin encoding NRPS GCF-46 in *Staphylococcus*: Among staphylococcal GCFs, GCF-46 depicted the highest posterior Beta-RD distribution (Fig. S5c). This GCF encodes for the lugdunin NRPS, which was determined to be an antibiotic produced by *S. lugdunensis* which is active against *S. aureus*(32). The ubiquity of this GCF being regarded as present in multiple staphylococcal species by *lsaBGC-AutoExpansion.py* was unexpected; however, further investigation revealed a rare instance of a small insertion-sequence element, which is ubiquitous across *Staphylococcus*, within the protcore region of the lugdunin BGC, thus resulting in *lsaBGC-AutoExpansion.py* over-classifying the presence of the GCF as it assumes the genes within the element are highly BGC-associated. Although we suspect such cases to be rare, we have since introduced more stringent requirements for defining homolog groups as part of the protcore of BGCs predicted by antiSMASH in more recent releases of *lsaBGC*.

Investigation into *Corynebacterium* GCF with the second highest Beta-RD value: In addition to *Corynebacterium* NRPS encoding GCF-50, the NRPS encoding GCF-9 exhibited a similarly high posterior Beta-RD distribution (Fig. S4d). GCF-9 was found to also be present in multiple species on the skin and was most prevalent in the well-known skin pathogen *C. diphtheria*(33). This GCF exhibited a similar scenario to GCF-50, in which it featured a highly conserved region with NRP synthase(s) flanked by MGEs in addition to a nearby phage and toxin/antitoxin system. GCF-9 was not further investigated because the NRP synthase (OG0002384) within GCF-9 could be found in multiple copies within certain genomes. To address such limitations, in version 1.3 of *lsaBGC*, we have switched the default settings to use more resolute orthogroup classifications by *OrthoFinder* (hierarchical orthogroups), which we found are able to better partition paralogous groups of NRP synthases.



## Ancestral inference of gene cluster family carriage

We performed ancestral inference of GCF carriage to predict vertical descent of GCFs within the skin-associated phylogenetic clades of *Staphylococcus* and *Corynebacterium* (Fig. S5ab)(34). GCF carriage across genomes was encoded as a binary trait and AncestralGeneRator (<https://github.com/broadinstitute/AncestralGeneRator>) was used to perform ancestral state reconstruction to infer carriage for inner-nodes of the *Corynebacterium* and *Staphylococcus* ribosomal phylogenies. Briefly, PAUP (v4b)(35) was used to perform maximum parsimony with the ACCTRAN algorithm, setting gain and loss costs to 10 and 5, respectively. Afterwards, result files from AncestralGeneRator were further processed and were visualized using iTol(28). This analysis revealed that four GCFs were ancestral to the *S. epidermidis/aureus* clade (predicted siderophore, terpene/T3PKS, cyclic lactone autoinducer, and hserlactone) and five GCFs were ancestral to the *C. tuberculostearicum* species complex (predicted siderophore, terpene, T1PKS, NAPAA, and NAPAA/betalactone).

## Identification of homolog groups associated with skin-residing species of *Staphylococcus* and *Corynebacterium*

We developed a script called crawlingFisher.py, provided in the *lsaBGC* suite, which tests for enrichment or depletion of homolog group presence amongst genomes in focal clades using Fisher's exact test. Provided with a phylogenetic tree and matrix specifying the presence of homolog groups across genomes, the program automatically performs such tests for each homolog group at each inner-node of the phylogeny. Multiple testing correction was performed comprehensively at the end using Benjamini-Hochberg false discovery rate and cases were considered statistically significant if the adjusted p-value was less than 0.05. Results were further filtered to retain only homolog groups which were present in at least 80% of genomes under the focal node and at most 20% of other genomes or, alternatively, at most 20% of genomes under the focal node and at least 80% of other genomes. Finally, only results pertaining to focal nodes where at least 80% of genomes were classified as skin-associated were reported (Fig. S5ab, Table S5).

## Understanding GCF conservation and composition through evolutionary and population genetic statistics

A core program of the *lsaBGC* suite is *lsaBGC-PopGene.py* which generates a table report of conservation and evolutionary statistics for each homolog group found within a specific GCF. To calculate certain homolog group statistics within this report, *lsaBGC-PopGene.py* begins by constructing protein alignments(7) for each homolog group and translating those to codon-based alignments using PAL2NAL(36). Codon-based alignments are additionally used to visualize conservation and domain structure for homolog groups and downstream in *lsaBGC-DiscoVary.py*. Three of the evolutionary statistics which codon alignments are used for computing are: (i) homolog-group specific variants of the Beta-Relative Divergence statistic, (ii) the rate of non-synonymous mutations relative to the rate of synonymous mutations (dN/dS), and (iii) Tajima's D statistic, which can be used to detect signatures of sweeping vs. balancing selection. *lsaBGC-PopGene.py* also allows users to specify population designations for each isolate with the focal GCF and computes additional statistics per homolog group pertaining to this information.

Assuming users provide genome-wide similarity estimates, either ANI or AAI, to *lsaBGC-PopGene.py*, it will calculate how pairwise homolog group similarities in codon alignments or protein alignments compare to such genome-wide expectations. This is reported as the median Beta-RelativeDivergence (analogous to the BGC-wide metric described in the *Materials and Methods*) across all pairs of genomes with the homolog group found in the focal GCF.

Tajima's D is a simple statistic where values below -2 typically indicate conservation or sweeping selection and a low ratio of rare minor alleles to high-frequency minor alleles while values above 2 indicate balancing selection. To calculate Tajima's D we modified a previous implementation(37) into a function within *IsaBGC*. Namely, we adjusted the calculation of Tajima's D to better reflect the mathematical derivation described by Simonsen, Churchill and Aquadro 1995(38). Additionally, we only considered sites as segregating or pairwise differences if both sequences being compared had a valid allele at the position within the alignments (alignment sites with gaps were ignored). As an alternate assessment of sequence variation to Tajima's D, we also report the proportion of sites across homolog group multiple sequence alignments where multiple alleles exist (major allele < 98%) and the proportion of sites where the major allele is non-dominant (<75%).

The rate of non-synonymous mutations relative to the rate of synonymous mutations is a classical statistic used to infer the effect of positive versus negative selection. If the rate for non-synonymous mutations is higher than the rate for synonymous mutations across different instances of a gene ( $dN/dS > 1$ ), it could suggest positive selection; whereas, if the reverse is true ( $dN/dS < 1$ ), it could suggest negative or purifying selection. We used Biopython's `codonalign.codonseq` module(39) to calculate  $dN/dS$  using the method described by Nei and Gojobori 1986(40). To more robustly calculate the statistic, we replace singleton codon instances (those observed in only one in one BGC) with gaps. Additionally, to ensure we avoid excessive computation when the number of sequences with a homolog group is large, we have implemented an empirical but non-exhaustive framework in which we calculate the median  $dN/dS$  between 1000 randomly selected pairs of samples. This random sampling and calculation of  $dN/dS$  is performed for 20 iterations and the median of  $dN/dS$  estimates across iterations, along with the absolute median deviation to assess robustness, is reported. Because we are actively seeking to improve our calculations of  $dN/dS$  in *IsaBGC* and Biopython's `codonalign` module is under development currently, we do not discuss results related to  $dN/dS$  in this study.

## **Inference of consensus order and directionality of homolog groups for a GCF**

We developed an algorithm to infer the consensus order of homolog groups relative to each other in the GCF, as well as their relative consensus directionality (sense vs. antisense). This algorithm works by first computing how many times a homolog group proceeds another homolog group across BGC instances belonging to a GCF. To gather this information, a single BGC instance, with the most homolog groups, is selected as the reference and used for configuring the general direction of the remaining GCF instances, deciding whether to flip them to better align with the order of the reference. The gene order information for each BGC, encoded as a dictionary, is then used to construct a primary path, starting from the homolog group most often found at one edge of a GCF to the homolog group most often found on the opposite edge. Afterwards, homolog groups which were not featured in this consensus path, potentially because they are infrequently found and not core to the GCF, are attempted to be placed in their most appropriate locations. The core ordering path will be structured primarily by more prevalent homolog groups. When looking at the *IsaBGC*-PopGene.py report table sorted by the consensus ordering of the homolog groups, it is thus important to consider the proportion of samples with the GCF which have a particular homolog group. The consensus directionality of homolog groups is simply based on whether most instances of it are forward or reverse relative to a reference gene from the chosen representative BGC.

## **Metagenomic mining for BGCs, homolog groups, and novel SNVs**

Metagenomic methods to explore microdiversity within microbiomes are continually advancing(41,42). We developed *IsaBGC*-DiscoVary.py to explore the micro-diversity of BGCs beyond the limited set of single-isolate genomes available for a lineage or taxa of interest and allow users to identify BGC genes

and base-resolution novel SNVs within metagenomic datasets (Fig. S2). Briefly, lsaBGC-DiscoVary.py serves two roles:

1. Assessing the presence of GCFs, individual homolog groups and if requested, phasing them (uses a custom approach leveraging DESMAN(43)), and
2. If users construct a comprehensive database of homolog groups alleles observed across all available assemblies for a taxonomy, lsaBGC-DiscoVary.py can also be used to identify putatively novel SNVs never previously observed at a particular site in a homolog group.

*Selection of Representative Alleles for Reference Database:* lsaBGC-DiscoVary.py begins by identifying representative allelic sequences for each homolog group in the focal GCF from the codon alignments listing file provided. To select representative alleles, it parses each homolog group's MSA FASTA file, considers sequences within a single median absolute deviation from the median length, and determines the number of differences between pairs of sequences. Sequences are deemed to be members of the same allelic cluster if they exhibit  $\geq 99\%$  identity of the shorter sequence and they differ at less than 10 sites from each other. Afterwards, pairs of such similar sequences are joined into larger clusters through single-linkage clustering and a representative sequence is chosen based on the minimal summed differences to the other sequences in the allelic cluster (e.g. the centroid). A Bowtie2(44) reference database is finally constructed from the representative sequences.

*Read Alignment to Reference Database and Alignment Parsing:* Processed sequencing reads are next aligned against the database of representative sequences for homolog group alleles using Bowtie2 in “--very-sensitive-local” mode(44). Because our database is currently based on individual genes (which are rather short), we found that aligning paired-end reads individually (as unpaired reads), increased sensitivity and did not significantly compromise specificity. This is similar to the approach for mapping used in MetaMLST(45). After alignment, sorting and indexing of BAM files is performed using samtools(46). Alignments for each homolog group are then processed and investigated using the pysam library in Python(47). An allele of a homolog group is considered potentially present if 90% of its sites are covered by at least one read, accounting for whether a read has lower than 30 base-quality and whether a site is a skipped region in the reference or corresponds to a deletion. Each alignment of a read to an allelic representative of a homolog group is then assessed as to whether it exhibits:

1. Alignment to multiple allelic representatives of the homolog group. Only alignments with the top/maximum alignment score of the read to any of the allelic representatives of the homolog group will be considered. This allows partitioning a read into one or more allelic representatives.
2. The read displays at least 95% identity to the reference allele sequence within the core alignment, where the core alignment is defined as the part of the alignment in between the first and last positions where reference and query sequences both have valid nucleotides (even if non-matching). If the core alignment length is  $\geq 100$  bp, 95% identity is required, while if it is shorter,  $\geq 60$  bp, 99% identity is required.
3. The total indel length within the core alignment is  $< 5$  bp. This allows for some leniency around small deletions and insertions, so as to not discard otherwise high-quality alignments. Reads with indels will be used for phasing or determining consensus alleles for homolog groups within metagenomes, but are not used as support for potentially novel SNVs reported.

If an alignment meets the above criteria, it is next assessed at each position for high base quality ( $\geq 30$  PHRED). If so, then the reference allele site is considered to be covered and the base of the read/query is noted. Further, because the reference allelic sequences were already aligned to each other and provided as the codon alignment inputs, we can translate the position of a site on the reference allele to a position in the codon multiple sequence alignment and ultimately gain a universal tally of base counts

at particular sites of a homolog group. The synchronization of reference gene positions to codon alignment positions is a key feature of *lsaBGC-DiscoVary.py* which enables it to identify novel SNVs.

*Final Assessment of Homolog Group Presence:* As is often cautioned in gene-based metagenomic analysis, faulty alignment of reads belonging to the lineage of interest or from other taxa can lead to a misinterpretation of enzyme presence or association with the lineage of interest. To further filter out faulty alignments, we parse the codon alignments of each homolog group, featuring reference and representative allelic sequence, and mark regions along it which are particularly "gappy", >10% of sequences have no allelic residue, as troublesome, including +/- 50bp around the start and end of each region. These regions were observed to present problems as default gap opening and extension costs in the aligning algorithm can occasionally lead to faulty alignments. Based on a similar logic that alignment and alignment scores could behave irrationally when only part of reads should properly align to a reference, we also deem the first and last 50 bp of a MSA as "troublesome".

Additionally, we self-align the full predicted proteome of the genomes used to establish representative alleles for homolog groups with DIAMOND(21) to identify regions along the homolog group MSAs which are similar at high-identity to potential paralogs. The criteria for defining these regions matches our criteria for mapping reads to allelic sequences. Thus, alignments which are  $\geq 20$  residues long and exhibit  $\geq 99\%$  amino acid identity or  $\geq 33$  residues long and exhibit  $\geq 90\%$  amino acid identity for  $\geq 5\%$  of the initial sample set are marked as "troublesome" for accurate alignment of reads.

For each sequencing read set, each homolog group is next more thoroughly assessed for carriage in the context of the full BGC. Up to this stage, the criteria for consideration of a BGC homolog group as present is pretty lenient and simply requires 1X coverage at 90% of sites for one of the representative alleles of the homolog group. Here, we further refine this criteria to require 1X coverage at  $\geq 90\%$  of sites in the codon alignment of the homolog group which are deemed as non-troublesome to align. For homolog groups which meet this requirement, the median depth of the middle 80% of positions is computed. The median of these homolog group specific median depths is then calculated along with the median absolute deviance. Because these homolog groups are expected to be co-located together in a BGC, potential variability in sequencing coverage across the genome, associated with active replication(48), should not result in a difference in coverage between homolog groups of the BGC. Based on this assumption, we next aim to identify and disregard homolog groups which are outliers in terms of their coverage ( $> 2$  median absolute deviances from the median of 80% trimmed median depths). These homolog groups will be difficult to gauge from raw metagenomics sequencing data as they will likely either lack enough coverage for resolved allele typing or have too much coverage and correspond to potentially multi-copy or common enzymes where undesired reads (from outside the focal BGC or lineage) are being aligned to the homolog group. Homolog groups are also disregarded if one of their predicted products contains mobile genetic element MGE suggestive keywords 'integrase' or 'transp'. In the most recent release of *lsaBGC*, we have updated the second keyword to 'transpos' to still allow for novel SNV detection on transporters.

If there are at least 5 homolog groups which are deemed present and not filtered by the above criteria, then BGC presence is assessed as a whole based on whether  $\geq 70\%$  of the core homolog groups are present (where the core homolog groups are those found in all BGCs from the initial *lsaBGC* processing/clustering analysis - e.g. the BGCs from high-quality genomes) or if just a single GCF specific homolog group is observed. Homolog groups are also disregarded if  $\geq 5\%$  of the initial set of the high-quality genomes used for initial *lsaBGC* analysis featured multiple copies of the homolog group (paralogs were common). Finally, a report file will be generated featuring only homolog groups deemed present (not filtered by above criteria) for samples which are regarded as featuring the BGC.

Note for investigations of cutimycin in *C. acnes*, we manually specified the core homolog groups involved in the thiopeptides biosynthesis(49) to increase sensitivity through an optional setting in *lsaBGC-DiscoVary.py*.

Consensus allele determination, allelic phasing, and phylogenetic visualization: For the set of retained homolog groups deemed as present within a sequencing readset for the focal GCF, the proportion of positions which are heterozygous are next computed and used to determine whether allelic phasing is needed or appropriate (current default  $\geq 5\%$  of sites along present homolog groups in the sequencing dataset need to be heterozygous to turn on phasing mode). If phasing mode is initiated and specified by users, DESMAN(43) is used to first determine the most likely number of strains and then phase the distinct haplotypes. If phasing mode is not initiated, then the consensus / majority-rule allele is selected for each site along the homolog group. Critically, the output from emission of the consensus sequence or multiple phased alleles for the homolog groups is not an independent sequence but rather an allele call for the sequencing / metagenomic read set at each position in the homolog group codon alignment. This allows us to avoid inferring faulty frameshifts in our sample-specific sequence(s) of homolog groups and allows direct incorporation into the codon alignment to build phylogenetic views. Additionally, we require that each base emitted / inferred has a minimum depth (current default is 5) and for the total depth at each position to be within a reasonable range of the median depth observed across all homolog groups regarded as present. For each inferred sequence, if an in-frame stop codon is observed, then downstream sites are automatically emitted as gaps.

The sequences inferred for each homolog group are brought into the context of the full codon alignments of reference alleles, after which filtering is performed to remove sequences which have gaps at more than 25% of non-"troublesome" sites of the alignments. Of the sequences retained after this filtering, individual sites are filtered to retain only those in which at most 10% of sequences have gaps / ambiguity. The resulting FASTA files are input into FastTree2(29) to infer a quick homolog group specific phylogeny which is then visualized in R and used to display the similarity of newly identified alleles / sequences to known / reference sequences.

Identification of novel SNVs: The cornerstone feature of lsaBGC-DiscoVary.py is its ability to search metagenomic / raw-read datasets and then assess whether they possess any potential novel SNVs not previously observed in the comprehensive set of known alleles gathered from all available assemblies for a taxa. Currently, only SNVs which are supported by at least 5 reads and not located in sites along the codon-alignment marked as "troublesome" are reported. Additionally, SNVs are not reported after the first in-frame stop codon observed from preliminary scanning. A comprehensive report of putatively novel SNVs is produced for all sequencing samples and homolog groups. A subset of reads which are supportive of putatively novel SNVs (last column in the report) are written in gzipped FASTQ format for each sequencing / metagenomic sample. This allows users to quickly taxonomically profile and assess that reads in fact belong to the lineage of interest (see subsection below).

## **Benchmarking lsaBGC-DiscoVary against assembly-based novel variant detection using *M. luteus* single-isolate sequencing readsets**

The correspondence for novel single nucleotide variants (SNVs) reported by lsaBGC-DiscoVary.py was assessed using whole-genome sequencing readsets for single isolates and compared to SNV identification based on an assembly-based approach (Fig. S11). lsaBGC-AutoProcess.py was run on all complete instances of *M. luteus* genomes present in GTDB to identify BGCs which were then clustered into 9 GCFs using lsaBGC-Cluster.py. Afterwards, lsaBGC-AutoExpansion.py was used to search all remaining, draft-quality, genomes from the species for homologous instances of the 9 GCFs identified and lsaBGC-PopGene.py was run to generate codon-based alignments for homolog groups associated with each GCF. We then used the sequencing reads from 132 *M. luteus* isolates sequenced by our lab, and absent in the current GTDB release, to run lsaBGC-DiscoVary.py and identify novel variants not previously observed in available genomes for the species. Because we had constructed draft assemblies for these same samples, we also identified GCF and homolog group instances in each assembly using

lsaBGC-AutoExpansion.py. MAFFT(7) was then used to incorporate the homolog group sequences identified in the draft assemblies into the codon alignments used for lsaBGC-DiscoVary.py analysis and the expanded codon alignments were subsequently parsed and assessed for novel SNVs. We found a high concordance between the lsaBGC-DiscoVary.py and assembly-based approaches for novel SNV detection, with 1788 of 1798 (99.5%) novel SNVs reported by DiscoVary also being found by the assembly based approach and 1788 of 1858 (96.3%) novel SNVs found by the assembly based approach also being reported by lsaBGC-DiscoVary (Table S11). Of the 69 novel SNVs only found by the assembly based approach, 30 were identified by lsaBGC-DiscoVary.py but not reported due to alleles exhibiting more or less coverage than expected given the median coverage of the BGC. Among the remaining 39 novel SNVs, manual examination of a common SNV (homolog group OG0001039 - position 65 in the codon alignments; found in 15 samples), revealed that it involved a cytosine allele within a lengthy stretch of 21 C/Gs, which could lead to less confident read alignment in lsaBGC-DiscoVary.py. Of the nine novel SNVs found by lsaBGC-DiscoVary, six correspond to minor alleles, explaining why they were not represented in sample assemblies.

### **Scrutinization of putative novel variants for lsaBGC-DiscoVary application to cutimycin in *C. acnes* and the comprehensive set of GCFs for the *C. tuberculostearicum* species complex**

By default, lsaBGC-DiscoVary.py only considers reads aligning at 95% identity to reference gene sequences and avoids reporting novel variants on genes if either annotation or coverage (relative to other genes in the BGC context) suggests they might be MGEs. The program also automatically extracts the subset of reads supporting the existence of novel variants, which can then be further screened with high stringency filters using otherwise computationally expensive methods. This can involve mapping reads supporting putative novel SNVs to a representative database of whole-genomes to observe for higher quality alignments or checking whether reads are classified as a particular taxonomy by Kraken2 (v2.0.8-beta)(50). As described previously(51), we used a customized database for Kraken2 analysis constructed from chromosomally complete bacterial, viral, archaeal, fungal, protozoan genomes as well as the human genome available on RefSeq, where plasmid sequences were distinguished to alleviate potential misclassifications of reads to a particular taxonomy.

We generally do not recommend screening variant-supporting reads for an appropriate taxonomic match with Kraken2 as it can generally lead to false negatives (on average only 92.8% of read-pairs per sample were taxonomically classified and misclassification can also occur; Fig. S14). For reads supporting putative novel variants within the cutimycin GCF of *C. acnes* or the six GCFs of the *C. tuberculostearicum* species complex, we required them to not align with a higher alignment score to a comprehensive database of all *Cutibacterium* or *Corynebacterium* genomes, respectively.

We further validated that reads supporting the existence of variants at conserved sites in the mycolic acid biosynthesis gene mapped best to the *C. tuberculostearicum* species and confirmed our suspicion that they could be misclassified by Kraken2 given the representation in the underlying database (Fig. S14). This *ad hoc* analysis was performed using paired-end information when mapping in Bowtie2 and accounting for concordant read pair alignments. While many reads supporting the existence of novel SNVs in the biosynthesis gene were classified as *C. acnes* by Kraken2, we found that none of the reads concordantly aligned to the comprehensive *Cutibacterium* genomes Bowtie 2 database whereas 85.07% concordantly aligned to the comprehensive *Corynebacterium* genomes Bowtie 2 database.

Note, the reasoning behind mapping SNV supporting reads to comprehensive genomic databases to purge false positive SNV calls, is that if a read truly aligns best to the BGC-associated homolog group where it was used to call a SNV, then it should at most generate a mapping to this comprehensive database of equal mapping quality. We align paired-end reads individually as was performed in lsaBGC-DiscoVary.py to have mapping scores be comparable with those from the original alignment. Furthermore, because our initial database within lsaBGC-DiscoVary.py consisted of allelic representatives of homolog groups, we realized that reads mapping to the start and end of genes could

result in a lower mapping score than if a larger genomic context was provided. To ensure we do not remove such reads mapping to gene edges, `scrutinizeNovelSNVSupport.py`, the program we developed to perform Kraken 2 and genomic database alignment based filtering of novel SNV reports from `IsaBGC-DiscoVary.py`, also compares the reference sequences when it finds a better Bowtie 2 mapping for SNV-supporting reads in genomic databases. If the reference sequence in the genomic database encompasses the reference sequence in the original alignment to the `IsaBGC-DiscoVary.py` homolog group database and the read was previously noted to map to a gene edge, then reads are still retained and SNV read support is not decremented. Five retained and supportive reads are required for an SNV to be featured in the filtered report produced by the program.

### **Determination of whether novel SNVs detected for BGCs of the *C. tuberculostearicum* species complex would be identified in metagenomic assemblies:**

Because the *C. tuberculostearicum* species complex can be found at low abundance at certain body sites within individuals, we wanted to determine whether the 34,545 novel SNVs found in BGC contexts of the species complex, after filtering SNVs where reads mostly map with higher scores to other regions of *Corynebacterium* genomes (as described in the previous subsection), would also be detected through metagenomic assembly. MEGAHIT(52) assembly was performed individually for all metagenomic readsets using default options and filtered to only retain 2 kb or longer contigs. Sample specific metagenomic assemblies were then combined into one consolidated FASTA file and used to construct a Bowtie 2 database(44). No pooling of samples for metagenomic assembly or subsequent dereplication was performed to ensure strain-specific variability is retained. After assessing the presence of putative novel SNVs and filtering those where supportive reads are mostly mapped to alternate regions in *Corynebacterium* genomes at higher scores (as described in the *Materials and Methods*), we performed similar investigations through Bowtie 2 mapping of reads supporting novel SNV presence to the concatenated database of contigs from individual metagenomes. Exact mappings of SNV-supporting reads to contigs were identified and used to assess whether novel SNVs are represented in metagenomic assemblies. Of the 34,545 novel SNVs deemed as reliable after mapping to the comprehensive database of *Corynebacterium* genomes, we found that 22,886 would also be identified from metagenomic assembly. Thus, 11,659 SNVs detected by `IsaBGC-DiscoVary.py` would be missed by metagenomic assembly and are not represented in contigs of considerable length. This is expected to be an underestimate as a concatenated database of metagenomic contigs from multiple samples was used. We additionally tested mapping SNV supporting reads to only a subset of the metagenomic contigs which we deemed as belonging to *Corynebacterium*. These contigs were identified by BLASTn analysis(11) of the metagenomic contigs to the comprehensive *Corynebacterium* genomics database used to filter putative novel SNVs in the previous subsection. Contigs with HSPs with query coverage greater than 25% and sequence identity greater than 85% or query coverage greater than 70% and sequence identity greater than 70% to one of the known *Corynebacterium* genomes were classified as *Corynebacterium*. Of the 556,030 total concatenated metagenomic contigs, 91,713 were classified as *Corynebacterium*. However, using this subset of contigs instead of the full metagenomic assembly database resulted in a minor increase in the number of SNVs which would be undetected by metagenomic assembly (11,763 instead of 11,659; an increase of 0.9%), suggesting that reads supporting novel SNVs detected by `IsaBGC-DiscoVary.py` were >99% *Corynebacterium*.

### ***IsaBGC-DiscoVary* based analyses of BGCs from the *C. tuberculostearicum* species complex:**

For the following analyses, we used 5,802 instances of 2,343 unique novel SNVs found across 66 homolog groups which were core or adjacent to core biosynthesis machinery of BGCs from the *C. tuberculostearicum* species complex. A total of 68 context-distinct homolog groups were determined after

partitioning multi-GCF occurring homolog groups based on their GCF context (OG0000094 and OG0000100 were each found in two separate GFCs and thus each was partitioned as two separate homolog groups). The 2,343 novel SNVs had site and allele-specific coverage values less than two median absolute deviations from the median coverage for the entire BGC.

Comparing SNVs shared across body sites and subjects: To comprehensively assess whether novel SNVs were shared more frequently for microbiomes of the same body site or from the same individual, we performed a pairwise, multi-iteration resampling analysis. For each pair of metagenomes, where each sample had at least 30 novel SNVs, we calculated the average Jaccard similarity across 1,000 simulations in which we randomly drew sets of 30 SNVs from each metagenomic sample to control for sequencing depth. Pairs of metagenomes were classified as one of three categories: “different body site, different participant”, “different body site, same participant”, and “same body site, different participant” (Fig. S13e). A two-sided Wilcoxon rank sum test for differences between the three distributions of average Jaccard similarities revealed that there was a statistically significant difference between “different body site, same participant” and “same body site, different participant” ( $p=6.01E-3$ ). Of greater relevance, there were statistically strong significant differences between the “different body site, different participant” distribution and the two other distributions: “different body site, same participant” and “same body site, different participant” ( $p=3.67E-09$  and  $p=1.39E-116$ , respectively). Thus, pairs of metagenomes were more likely to share novel SNVs if they were either from the same body site or same participant compared to if they were from different participants and different body sites. Two metagenomic samples for participant S002 were not used because they corresponded to a resampling of two of their body sites and we did not have enough of such samples to pursue a temporal analysis. We assessed this testing using different considerations/cutoffs for: (i) the number of novel SNVs needed for a metagenome to be considered in the analysis and (ii) whether or not to account for singleton SNVs. In all cases, statistical testing for differences between distributions of pairwise metagenome categories yielded similar conclusions.

Assessing trends for novel SNV predicted synonymous to non-synonymous rate with metagenome ubiquity and site conservation: Using reports from *IsaBGC-DiscoVary.py*, we performed a systematic prediction of whether novel SNVs identified on the 68 context-distinct homolog groups nearby or overlapping with protocore regions of BGCs in *C. tuberculostearicum* corresponded to synonymous or non-synonymous substitutions. As described in other sections, SNV alleles were compared to corresponding alleles observed at the site on the reference gene sequences which SNVs were called upon. We aimed to check whether the rate of synonymous to non-synonymous novel SNVs showed a relationship to: (i) how common SNVs were across metagenomic samples and (ii) conservation levels across protein sequences.

Novel SNV predicted synonymous to non-synonymous rate increases with SNV metagenome ubiquity amongst metagenomic samples: The occurrence of novel SNVs, a novel allele observed at a specific site in the multiple sequence codon alignment of a particular homolog group, was tabulated across metagenomic samples. While novel SNVs found in a single metagenome were only 2.4X as likely to correspond to a synonymous change as opposed to a non-synonymous change, more prevalent novel SNVs, found in ten or more samples, were 8.4X more likely to correspond to a synonymous change (Fig. S13d).

Novel SNV predicted synonymous to non-synonymous rate increases at conserved sites in protein sequences: We searched RefSeq’s bacterial NR database for remote homologs of the 68 context-distinct homolog groups within or adjacent to protocore regions of BGCs. The top 20 homologs which belonged to classified bacterial species outside of the *Corynebacterium* genus or *Corynebacteriales* family were identified using *hmmsearch* from HMMER3 with profile-HMMs gathered from *IsaBGC-Expansion* for each homolog group and an E-value threshold of  $1E-20$  (Table S12). Of the 2,343 unique novel SNVs identified in protocore or protocore adjacent homolog groups of BGCs from the *C. tuberculostearicum* species complex, 2,244 were found in 34 context-distinct homolog groups with conservation information calculated from alignments featuring distantly related homologs from diverse bacteria. The sequences of representative homologs were added to protein alignments for homolog groups in *IsaBGC-PopGene*



results using the ‘--add’ option in MAFFT(7). Afterwards, we used the alignments for scoring protein sequence conservation with Jensen-Shannon divergence(53). Conservation scores were used to determine conservation percentiles of sites along protein alignments for individual homolog groups.

Rates of predicted synonymous to non-synonymous novel SNVs were calculated for each conservation percentile across all homolog groups. Novel SNVs found in the top five percentile of conserved sites for a homolog group were approximately 9.8X more likely to correspond to a synonymous change as opposed to a non-synonymous change (Fig. 5b). In contrast, the bottom 20% of conserved sites (the least conserved sites), were only 1.7X as likely to represent non-synonymous substitutions as synonymous substitutions. Further, the number of novel SNVs, either synonymous or non-synonymous observed at conserved sites, amongst the top 20% of conserved sites was 412, and lower than the number of novel SNVs found in the 20-40 (n=459), 40-60 (n=490), and 60-80 (n=493). Only 388 novel SNVs were found in the 80-100 percentile ranges of conserved sites and we suspect that this decrease is due to read alignments to such regions becoming more challenging at the thresholds required by lsaBGC-DiscoVary.py.

For the three SNVs predicted to result in non-synonymous differences and in the top five percentile of conserved sites along the mycolic acid PKS, we inspected their codon contexts to check that there were no additional variants which, in aggregate, might result in a synonymous change (Fig. 5c; Table S13).

## Supplementary Text References

1. Seemann T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics* [Internet]. 2014 Mar 18 [cited 2021 Dec 28];30(14):2068–9. Available from: <https://academic.oup.com/bioinformatics/article/30/14/2068/2390517>
2. Blin K, Shaw S, Kloosterman AM, Charlop-Powers Z, van Wezel GP, Medema MH, et al. antiSMASH 6.0: improving cluster detection and comparison capabilities. *Nucleic Acids Res* [Internet]. 2021 Jul 2;49(W1):W29–35. Available from: <http://dx.doi.org/10.1093/nar/gkab335>
3. Emms DM, Kelly S. OrthoFinder: phylogenetic orthology inference for comparative genomics. *Genome Biol* [Internet]. 2019 Nov 14;20(1):238. Available from: <http://dx.doi.org/10.1186/s13059-019-1832-y>
4. Parks D. CompareM: A toolbox for comparative genomics [Internet]. Github; [cited 2022 Mar 5]. Available from: <https://github.com/dparks1134/CompareM>
5. Jain C, Rodriguez-R LM, Phillippy AM, Konstantinidis KT, Aluru S. High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. *Nat Commun* [Internet]. 2018 Nov 30 [cited 2022 Mar 1];9(1):1–8. Available from: <https://www.nature.com/articles/s41467-018-07641-9>
6. Smirnov V. Recursive MAGUS: Scalable and accurate multiple sequence alignment. *PLoS Comput Biol* [Internet]. 2021 Oct;17(10):e1008950. Available from: <http://dx.doi.org/10.1371/journal.pcbi.1008950>
7. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol Biol Evol* [Internet]. 2013 Apr;30(4):772–80. Available from: <http://dx.doi.org/10.1093/molbev/mst010>
8. Lee MD. GToTree: a user-friendly workflow for phylogenomics. *Bioinformatics* [Internet]. 2019 Oct 15;35(20):4162–4. Available from: <http://dx.doi.org/10.1093/bioinformatics/btz188>

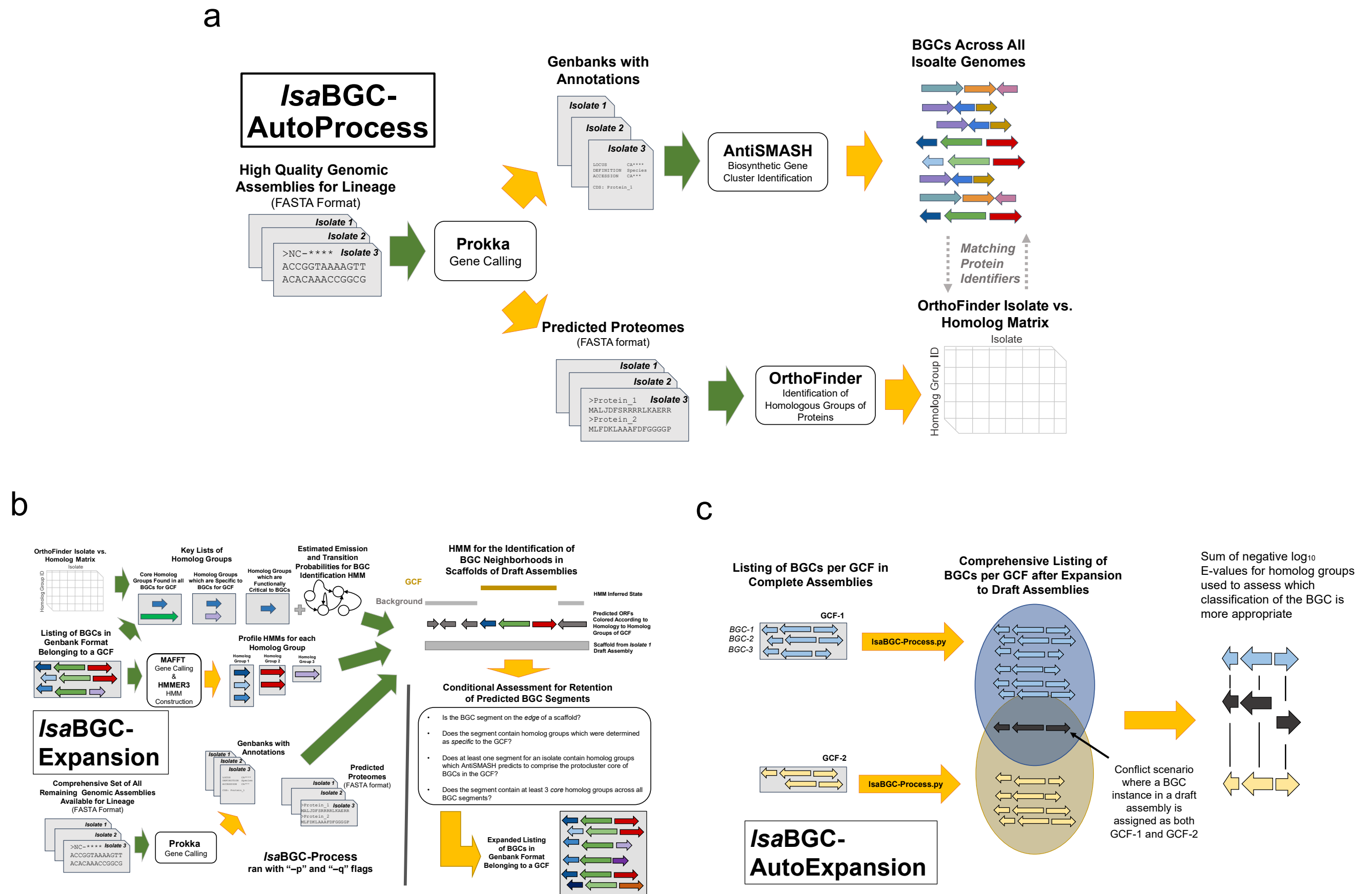
9. Carroll LM, Larralde M, Fleck JS, Ponnudurai R, Milanese A, Cappio E, et al. Accurate de novo identification of biosynthetic gene clusters with GECCO [Internet]. *bioRxiv*. 2021 [cited 2022 Jun 23]. p. 2021.05.03.442509. Available from: <https://www.biorxiv.org/content/10.1101/2021.05.03.442509v1>
10. Hannigan GD, Prihoda D, Palicka A, Soukup J, Klempir O, Rampula L, et al. A deep learning genome-mining strategy for biosynthetic gene cluster prediction. *Nucleic Acids Res* [Internet]. 2019 Oct 10;47(18):e110. Available from: <http://dx.doi.org/10.1093/nar/gkz654>
11. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. *BMC Bioinformatics* [Internet]. 2009 Dec 15;10:421. Available from: <http://dx.doi.org/10.1186/1471-2105-10-421>
12. Madhaiyan M, Wirth JS, Saravanan VS. Phylogenomic analyses of the Staphylococcaceae family suggest the reclassification of five species within the genus *Staphylococcus* as heterotypic synonyms, the promotion of five subspecies to novel species, the taxonomic reassignment of five *Staphylococcus* species to *Mammaliococcus* gen. nov., and the formal assignment of *Nosocomiicoccus* to the family Staphylococcaceae. *Int J Syst Evol Microbiol* [Internet]. 2020 Nov;70(11):5926–36. Available from: <http://dx.doi.org/10.1099/ijsem.0.004498>
13. Kania SA. Reclassification of *Staphylococcus schleiferi* by Madhaiyan et al. lacks key supporting data. *Int J Syst Evol Microbiol* [Internet]. 2022 Apr;72(4). Available from: <http://dx.doi.org/10.1099/ijsem.0.005289>
14. Navarro-Muñoz JC, Selem-Mojica N, Mullaney MW, Kautsar SA, Tryon JH, Parkinson EI, et al. A computational framework to explore large-scale biosynthetic diversity. *Nat Chem Biol* [Internet]. 2020 Jan;16(1):60–8. Available from: <http://dx.doi.org/10.1038/s41589-019-0400-9>
15. van Dongen S, Abreu-Goodger C. Using MCL to extract clusters from networks. *Methods Mol Biol* [Internet]. 2012;804:281–95. Available from: [http://dx.doi.org/10.1007/978-1-61779-361-5\\_15](http://dx.doi.org/10.1007/978-1-61779-361-5_15)
16. Steinke K, Mohite OS, Weber T, Kovács ÁT. Phylogenetic Distribution of Secondary Metabolites in the *Bacillus subtilis* Species Complex. *mSystems* [Internet]. 2021 Mar 9;6(2). Available from: <http://dx.doi.org/10.1128/mSystems.00057-21>
17. Aramaki T, Blanc-Mathieu R, Endo H, Ohkubo K, Kanehisa M, Goto S, et al. KofamKOALA: KEGG Ortholog assignment based on profile HMM and adaptive score threshold. *Bioinformatics* [Internet]. 2019 Nov 19 [cited 2021 Dec 28];36(7):2251–2. Available from: <https://academic.oup.com/bioinformatics/article/36/7/2251/5631907>
18. Melnyk RA, Hossain SS, Haney CH. Convergent gain and loss of genomic islands drive lifestyle changes in plant-associated *Pseudomonas*. *ISME J* [Internet]. 2019 Jun;13(6):1575–88. Available from: <http://dx.doi.org/10.1038/s41396-019-0372-5>
19. Cantalapiedra CP, Hernández-Plaza A, Letunic I, Bork P, Huerta-Cepas J. eggNOG-mapper v2: Functional Annotation, Orthology Assignments, and Domain Prediction at the Metagenomic Scale. *Mol Biol Evol* [Internet]. 2021 Oct 1 [cited 2021 Dec 28];38(12):5825–9. Available from: <https://academic.oup.com/mbe/article/38/12/5825/6379734>
20. Eddy SR. Accelerated Profile HMM Searches. *PLoS Comput Biol* [Internet]. 2011 Oct;7(10):e1002195. Available from: <http://dx.doi.org/10.1371/journal.pcbi.1002195>

21. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. *Nat Methods* [Internet]. 2014 Nov 17 [cited 2021 May 21];12(1):59–60. Available from: <https://www.nature.com/articles/nmeth.3176>
22. Schreiber J. Pomegranate: fast and flexible probabilistic modeling in python. *J Mach Learn Res* [Internet]. 2017; Available from: <https://www.jmlr.org/papers/volume18/17-636/17-636.pdf>
23. Cimermancic P, Medema MH, Claesen J, Kurita K, Wieland Brown LC, Mavrommatis K, et al. Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. *Cell* [Internet]. 2014 Jul 17;158(2):412–21. Available from: <http://dx.doi.org/10.1016/j.cell.2014.06.034>
24. Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. *Bioinformatics* [Internet]. 2012 Feb 15;28(4):593–4. Available from: <http://dx.doi.org/10.1093/bioinformatics/btr708>
25. Wick RR, Judd LM, Gorrie CL, Holt KE. Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Comput Biol* [Internet]. 2017 Jun;13(6):e1005595. Available from: <http://dx.doi.org/10.1371/journal.pcbi.1005595>
26. Wickham H. *ggplot2: Elegant Graphics for Data Analysis* [Internet]. Springer, New York, NY; 2009. Available from: <https://link.springer.com/book/10.1007/978-0-387-98141-3>
27. Yu G, Smith DK, Zhu H, Guan Y, Lam TT-Y. Ggtree : An r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods Ecol Evol* [Internet]. 2017 Jan;8(1):28–36. Available from: <https://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12628>
28. Letunic I, Bork P. Interactive Tree Of Life (iTOL) v4: recent updates and new developments. *Nucleic Acids Res* [Internet]. 2019 Apr 1 [cited 2022 Feb 27];47(W1):W256–9. Available from: <https://academic.oup.com/nar/article/47/W1/W256/5424068>
29. Price MN, Dehal PS, Arkin AP. FastTree 2--approximately maximum-likelihood trees for large alignments. *PLoS One* [Internet]. 2010 Mar 10;5(3):e9490. Available from: <http://dx.doi.org/10.1371/journal.pone.0009490>
30. Gilchrist CLM, Chooi Y-H. clinker & clustermap.js: automatic generation of gene cluster comparison figures. *Bioinformatics* [Internet]. 2021 Jan 18 [cited 2022 Mar 6];37(16):2473–5. Available from: <https://academic.oup.com/bioinformatics/article-abstract/37/16/2473/6103786>
31. Parks DH, Chuvochina M, Rinke C, Mussig AJ, Chaumeil P-A, Hugenholtz P. GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic Acids Res* [Internet]. 2021 Sep 14; Available from: <http://dx.doi.org/10.1093/nar/gkab776>
32. Zipperer A, Konnerth MC, Laux C, Berscheid A, Janek D, Weidenmaier C, et al. Human commensals producing a novel antibiotic impair pathogen colonization. *Nature* [Internet]. 2016 Jul 28;535(7613):511–6. Available from: <http://dx.doi.org/10.1038/nature18634>
33. Hadfield TL, McEvoy P, Polotsky Y, Tzinserling VA, Yakovlev AA. The pathology of diphtheria. *J Infect Dis* [Internet]. 2000 Feb;181 Suppl 1(s1):S116–20. Available from: <http://dx.doi.org/10.1086/315551>

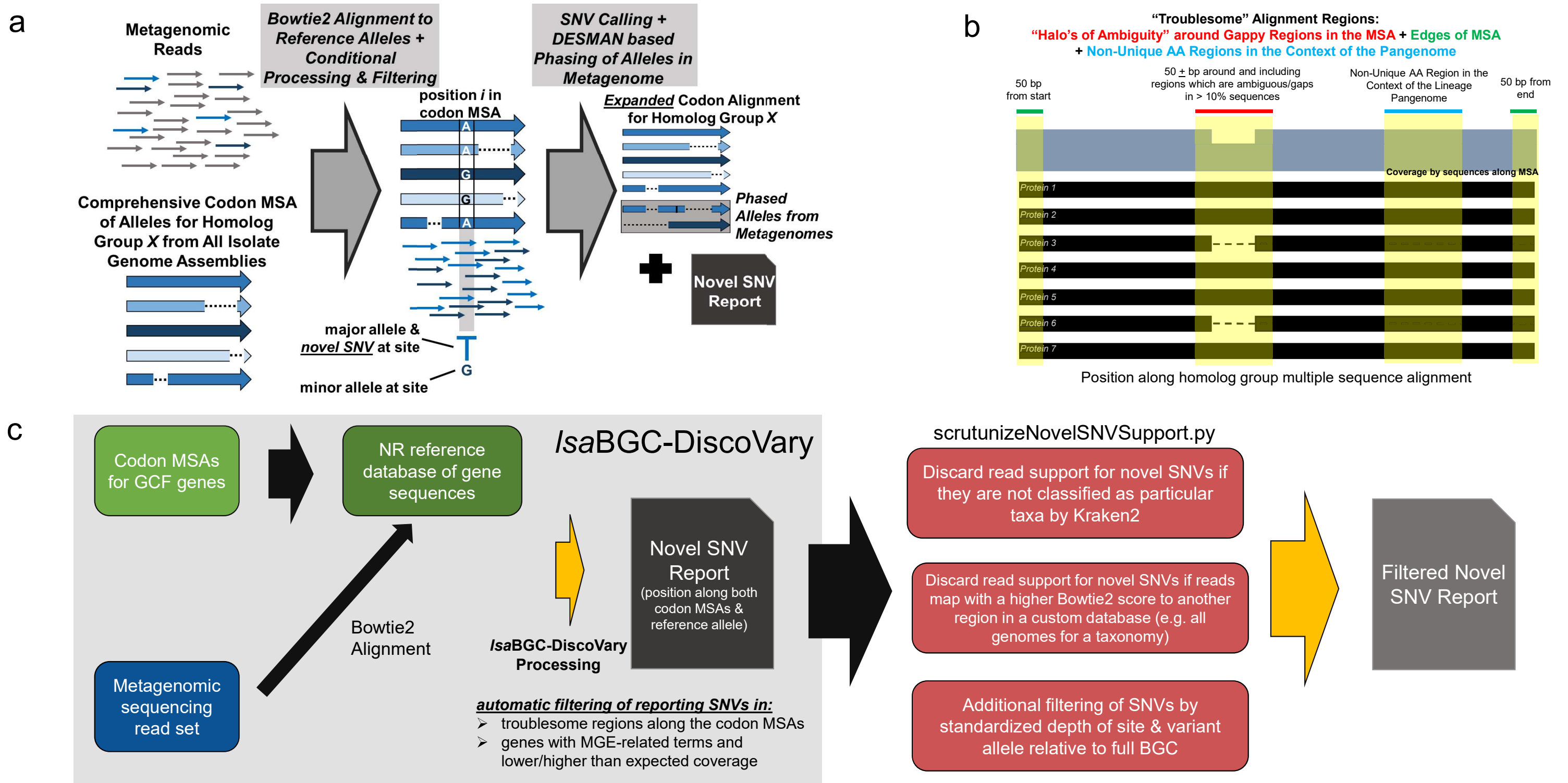
34. van Dijk LR, Walker BJ, Straub TJ, Worby CJ, Grote A, Schreiber HL 4th, et al. StrainGE: a toolkit to track and characterize low-abundance strains in complex microbial communities. *Genome Biol* [Internet]. 2022 Mar 7;23(1):74. Available from: <http://dx.doi.org/10.1186/s13059-022-02630-0>
35. Swofford DL. PAUP: phylogenetic analysis using parsimony. Mac Version 3 1 1(Computer program and manual) [Internet]. 1993; Available from: <https://ci.nii.ac.jp/naid/10003335753/>
36. Suyama M, Torrents D, Bork P. PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. *Nucleic Acids Res* [Internet]. 2006 Jul 1;34(Web Server issue):W609-12. Available from: <http://dx.doi.org/10.1093/nar/gkl315>
37. Whalley T. LICENSE at master · WhalleyT/tajima [Internet]. Github; [cited 2022 Mar 4]. Available from: <https://github.com/WhalleyT/tajima>
38. Simonsen KL, Churchill GA, Aquadro CF. Properties of statistical tests of neutrality for DNA polymorphism data. *Genetics* [Internet]. 1995 Sep;141(1):413–29. Available from: <http://dx.doi.org/10.1093/genetics/141.1.413>
39. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* [Internet]. 2009 Mar 20 [cited 2022 Mar 4];25(11):1422–3. Available from: <https://academic.oup.com/bioinformatics/article/25/11/1422/330687>
40. Nei M, Gojobori T. Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Mol Biol Evol* [Internet]. 1986 Sep;3(5):418–26. Available from: <http://dx.doi.org/10.1093/oxfordjournals.molbev.a040410>
41. Olm MR, Crits-Christoph A, Bouma-Gregson K, Firek BA, Morowitz MJ, Banfield JF. inStrain profiles population microdiversity from metagenomic data and sensitively detects shared microbial strains. *Nat Biotechnol* [Internet]. 2021 Jan 18; Available from: <http://dx.doi.org/10.1038/s41587-020-00797-0>
42. Gregory AC, Gerhardt K, Zhong Z-P, Bolduc B, Temperton B, Konstantinidis KT, et al. MetaPop: a pipeline for macro- and microdiversity analyses and visualization of microbial and viral metagenome-derived populations. *Microbiome* [Internet]. 2022 Mar 15;10(1):49. Available from: <http://dx.doi.org/10.1186/s40168-022-01231-0>
43. Quince C, Delmont TO, Raguideau S, Alneberg J, Darling AE, Collins G, et al. DESMAN: a new tool for de novo extraction of strains from metagenomes. *Genome Biol* [Internet]. 2017 Sep 21 [cited 2021 May 17];18(1):1–22. Available from: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-017-1309-9>
44. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. *Nat Methods* [Internet]. 2012 Mar 4;9(4):357–9. Available from: <http://dx.doi.org/10.1038/nmeth.1923>
45. Zolfo M, Tett A, Jousson O, Donati C, Segata N. MetaMLST: multi-locus strain-level bacterial typing from metagenomic samples. *Nucleic Acids Res* [Internet]. 2017 Jan 25;45(2):e7. Available from: <http://dx.doi.org/10.1093/nar/gkw837>
46. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* [Internet]. 2009 Aug 15;25(16):2078–9. Available from: <http://dx.doi.org/10.1093/bioinformatics/btp352>

47. Gilman P, Janzou S, Guittet D, Freeman J, DiOrio N, Blair N, et al. PySAM (Python Wrapper for System Advisor Model “SAM”) [Internet]. National Renewable Energy Lab. (NREL), Golden, CO (United States); 2019 Aug [cited 2022 Mar 5]. Report No.: PySAM. Available from: <https://www.osti.gov/servlets/purl/1559931>
48. Bremer H, Churchward G. An examination of the Cooper-Helmstetter theory of DNA replication in bacteria and its underlying assumptions. *J Theor Biol* [Internet]. 1977 Dec 21;69(4):645–54. Available from: [http://dx.doi.org/10.1016/0022-5193\(77\)90373-3](http://dx.doi.org/10.1016/0022-5193(77)90373-3)
49. Claesen J, Spagnolo JB, Ramos SF, Kurita KL, Byrd AL, Aksenov AA, et al. A *Cutibacterium acnes* antibiotic modulates human skin microbiota composition in hair follicles. *Sci Transl Med* [Internet]. 2020 Nov 18;12(570). Available from: <http://dx.doi.org/10.1126/scitranslmed.aay5445>
50. Wood DE, Lu J, Langmead B. Improved metagenomic analysis with Kraken 2. *Genome Biol* [Internet]. 2019 Nov 28 [cited 2021 May 21];20(1):1–13. Available from: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1891-0>
51. Swaney MH, Sandstrom S, Kalan LR. Cobamide Sharing Is Predicted in the Human Skin Microbiome. *mSystems* [Internet]. 2022 Aug 15;e0067722. Available from: <http://dx.doi.org/10.1128/msystems.00677-22>
52. Li D, Liu C-M, Luo R, Sadakane K, Lam T-W. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* [Internet]. 2015 Jan 20 [cited 2022 Mar 10];31(10):1674–6. Available from: <https://academic.oup.com/bioinformatics/article/31/10/1674/177884>
53. Capra JA, Singh M. Predicting functionally important residues from sequence conservation. *Bioinformatics* [Internet]. 2007 May 22 [cited 2022 Mar 11];23(15):1875–82. Available from: <https://academic.oup.com/bioinformatics/article/23/15/1875/203579>

## Supplementary Figures

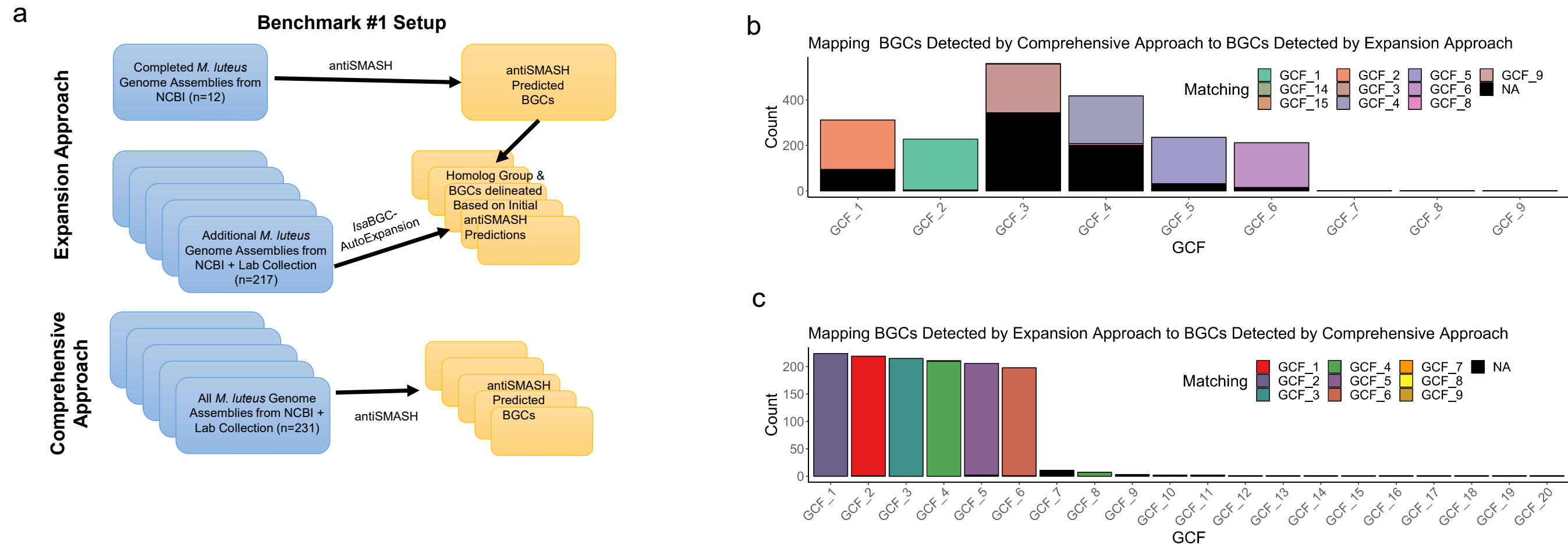


**Figure S1: Schematics of select IsaBGC programs and workflows.** a) A schematic of the *IsaBGC-AutoProcess* workflow which generates the inputs required for IsaBGC analyses. b) An overview of the algorithm behind *IsaBGC-Expansion* used to identify homologous instances of GCFs in a sensitive and efficient manner from potentially draft-quality assemblies. c) A schematic of the *IsaBGC-AutoExpansion* workflow which automatically runs *IsaBGC-Expansion* for each GCF and then resolves conflicts of overlap and consolidates results.



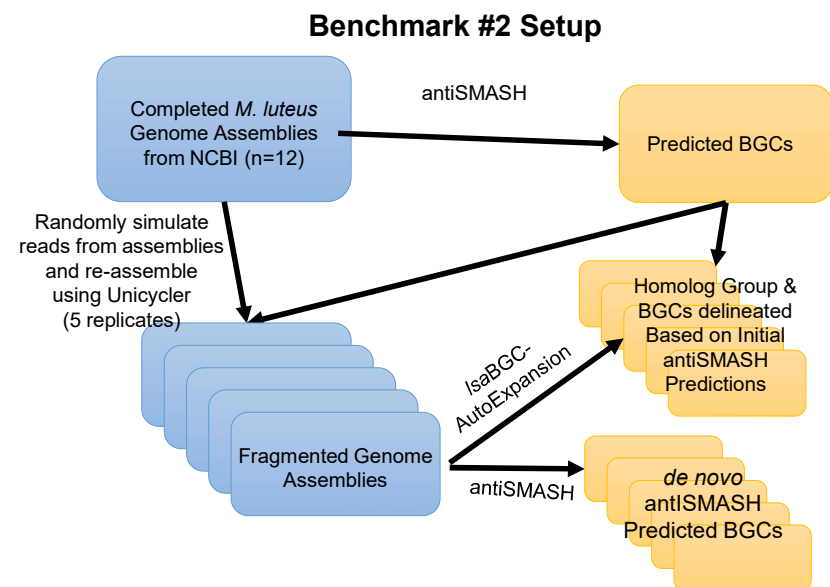
**Figure S2: An overview of IsaBGC-DiscoVary.** **a)** An overview of the IsaBGC-DiscoVary algorithm is shown. Codon alignments for each BGC homolog group, as constructed by IsaBGC-PopGene, are used to dereplicate and select representative alleles for mapping metagenomic readsets with Bowtie2. Afterwards, alignments are processed and used to identify whether homolog groups are represented in metagenomic samples. SNV sites along individual representative genes for homolog groups are mapped to comprehensive codon alignments and used to assess whether the allele represented by the SNV has previously been observed at the codon alignment position. Optional phasing of multiple alleles for a homolog group within a lineage or taxa can also be performed using DESMAN. **b)** Novel SNVs are not reported if they are within specific regions along codon alignments, including regions which are towards the beginning or end of the alignment, regions which are highly redundant, and regions where >10% of gene instances have deletions or lack sequence. **c)** An overview of IsaBGC-DiscoVary for determining putative novel SNVs which can then be further filtered using exhaustive methods to assert that reads supporting their presence do not map better to other taxa or genomic regions.



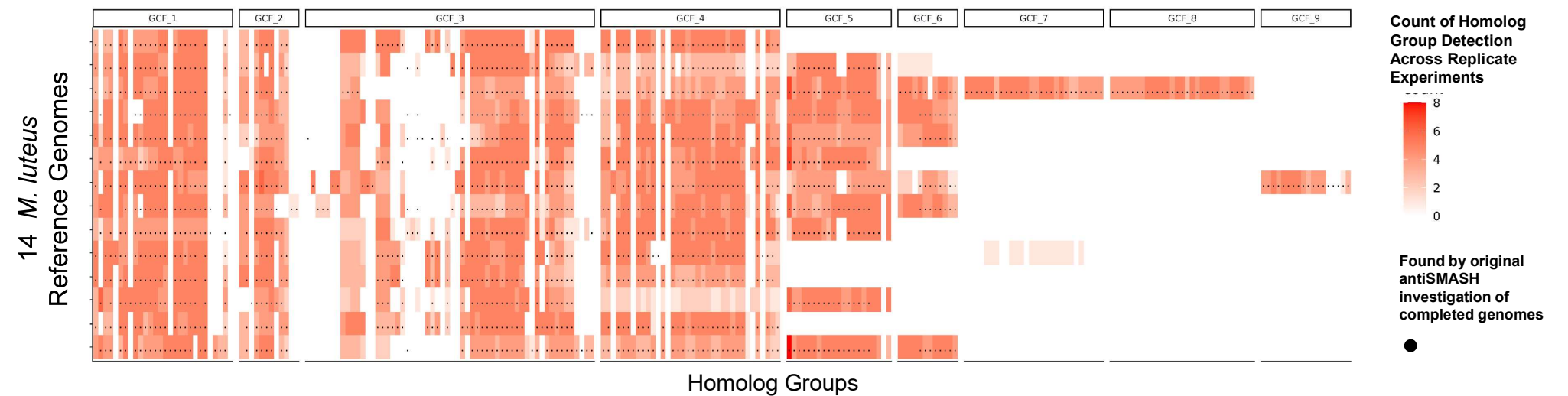


**Figure S3: Benchmarking *IsaBGC-AutoExpansion* using actual draft-quality *M. luteus* assemblies.** **a)** An overview of the first benchmarking experiment is shown comparing the use of *IsaBGC-AutoExpansion* to comprehensive antiSMASH to profile GCFs in 132 *M. luteus* genomes. **b)** Results from the first benchmarking experiment are shown. BGC instances identified by antiSMASH in genomes were mapped to BGC segments identified by running *IsaBGC-AutoExpansion* trained on GCFs from complete *M. luteus* genomes. “NA”, shown in black, represent GCF segments which were undetected by antiSMASH. **c)** Results from the first benchmarking experiment are shown. BGC instances identified by *IsaBGC-AutoExpansion* were mapped to BGC segments identified by running *IsaBGC-AutoExpansion*. “NA”, shown in black, represent GCF segments which were undetected by *IsaBGC-AutoExpansion*.

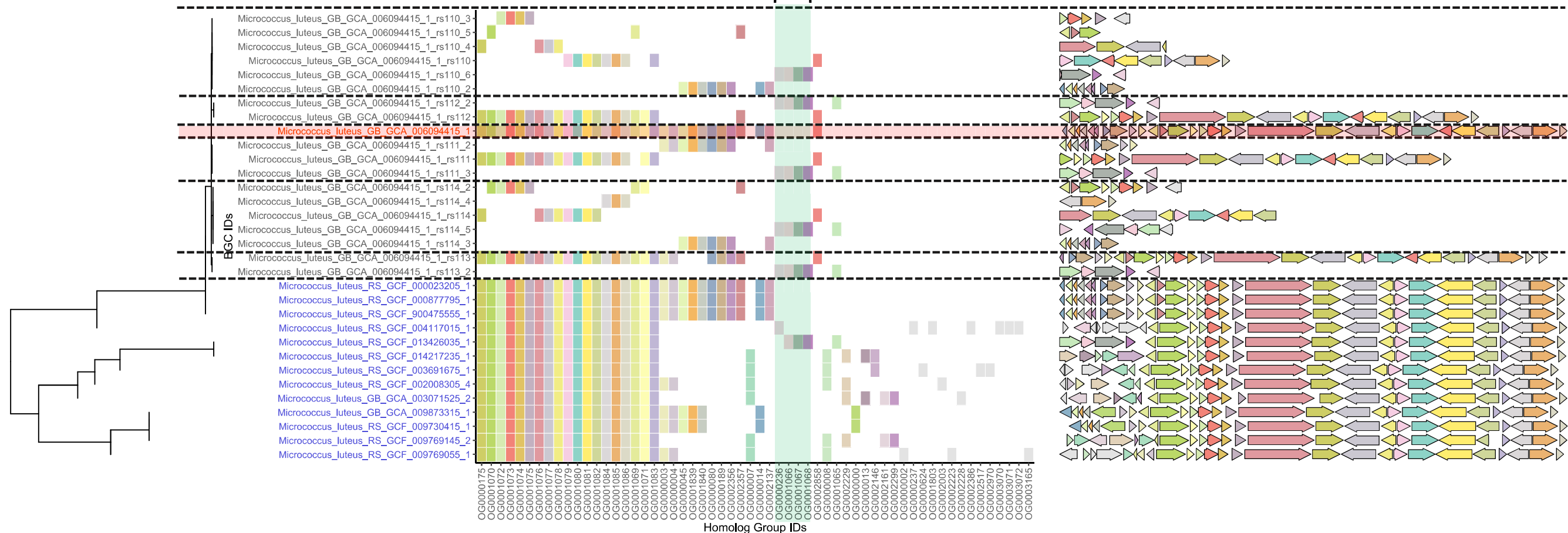
a



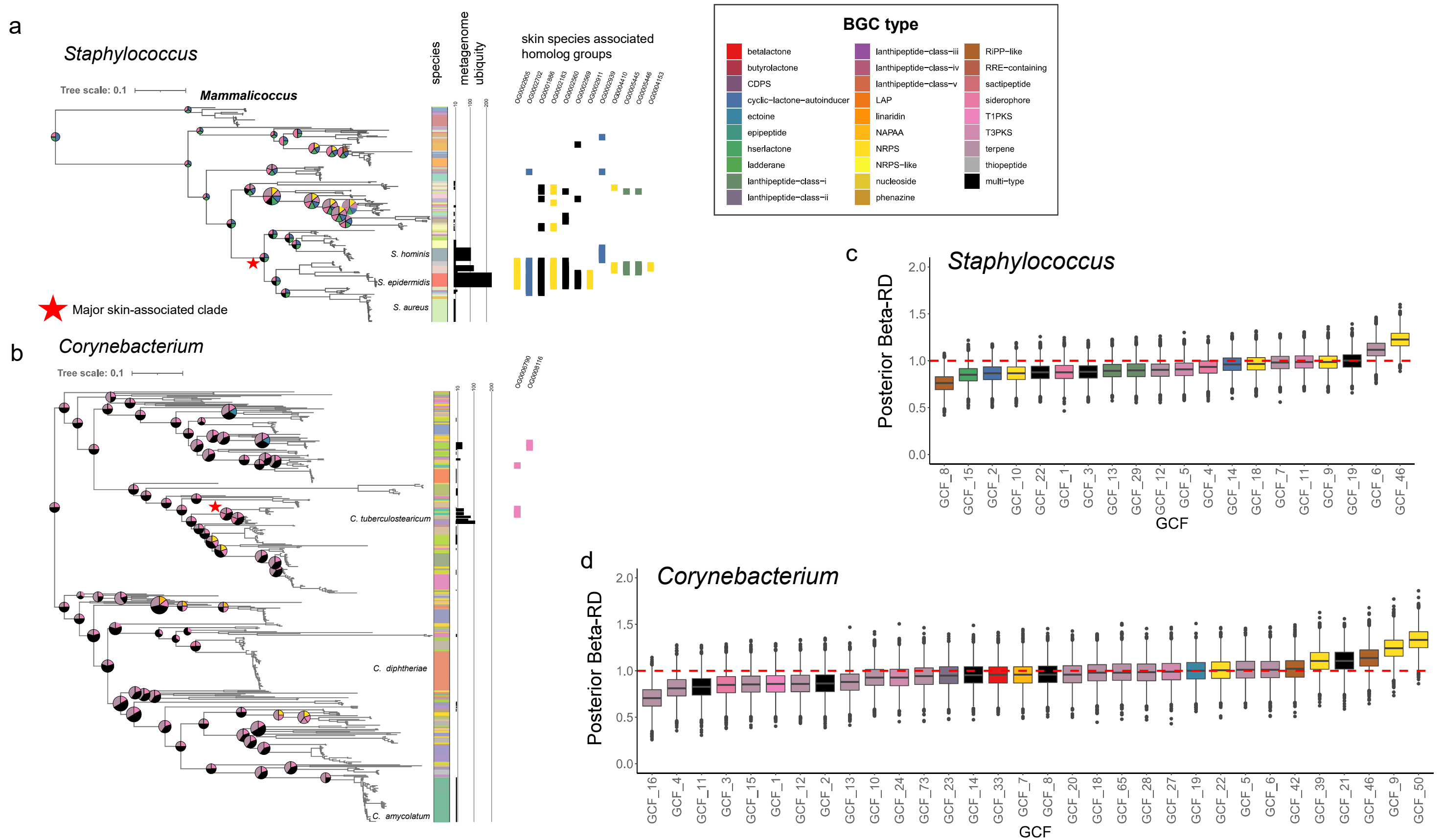
b



c



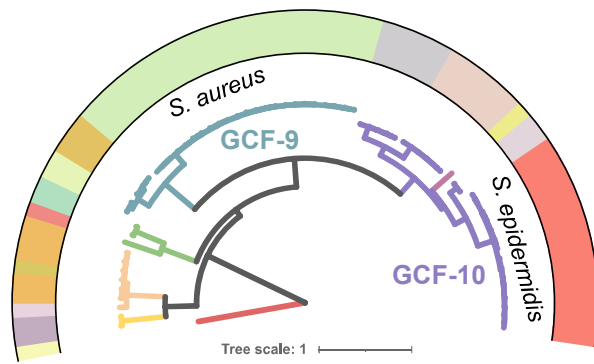
**Figure S4: Benchmarking *IsaBGC-AutoExpansion* on simulated fragmented assemblies showcases ability to synchronize GCF boundaries.** **a)** An overview of the second benchmarking experiment is shown comparing the use of *IsaBGC-AutoExpansion* to antiSMASH for predicting BGCs in fragmented versions of 14 complete genomes of *M. luteus*. **b)** The heatmap shows how many times each homolog group from different GCFs were identified by *IsaBGC-AutoExpansion* across 5 replicate simulations where fragmented genomic assemblies were generated from simulated reads for each of the corresponding complete *M. luteus* genomes. Dots signify whether homolog groups were detected as part of a BGC for a particular genome in the original antiSMASH annotation of the completed genome (unfragmented). **c)** *IsaBGC-See* was used to illustrate segments detected for a single GCF, GCF-3, by *IsaBGC-AutoExpansion* in fragmented versions of a single genome, *M. luteus* GCA\_006094415, in relation to the BGC detected on the completed genome (unfragmented; highlighted in red) and homologous instances of the GCF in the other 13 completed *M. luteus* genomes (blue labels). Dotted black lines separate GCF instances predicted for the five different replicate fragmented assemblies based on simulated reads from the focal completed genome. The first 6 set of BGC segments illustrates how *IsaBGC-AutoExpansion* is able to find and put together multiple BGC segments into the same GCF despite their dispersion across six different scaffolds. One segment (highlighted in teal) includes genes which were not part of the original BGC delineation on the completed focal genome but were part of homologous BGCs from different completed genomes. We validated these genes are adjacent to the original BGC delineation in the completed focal genome and were simply not included as part of the BGC because they were slightly more distant from core genes/domains of the BGC.



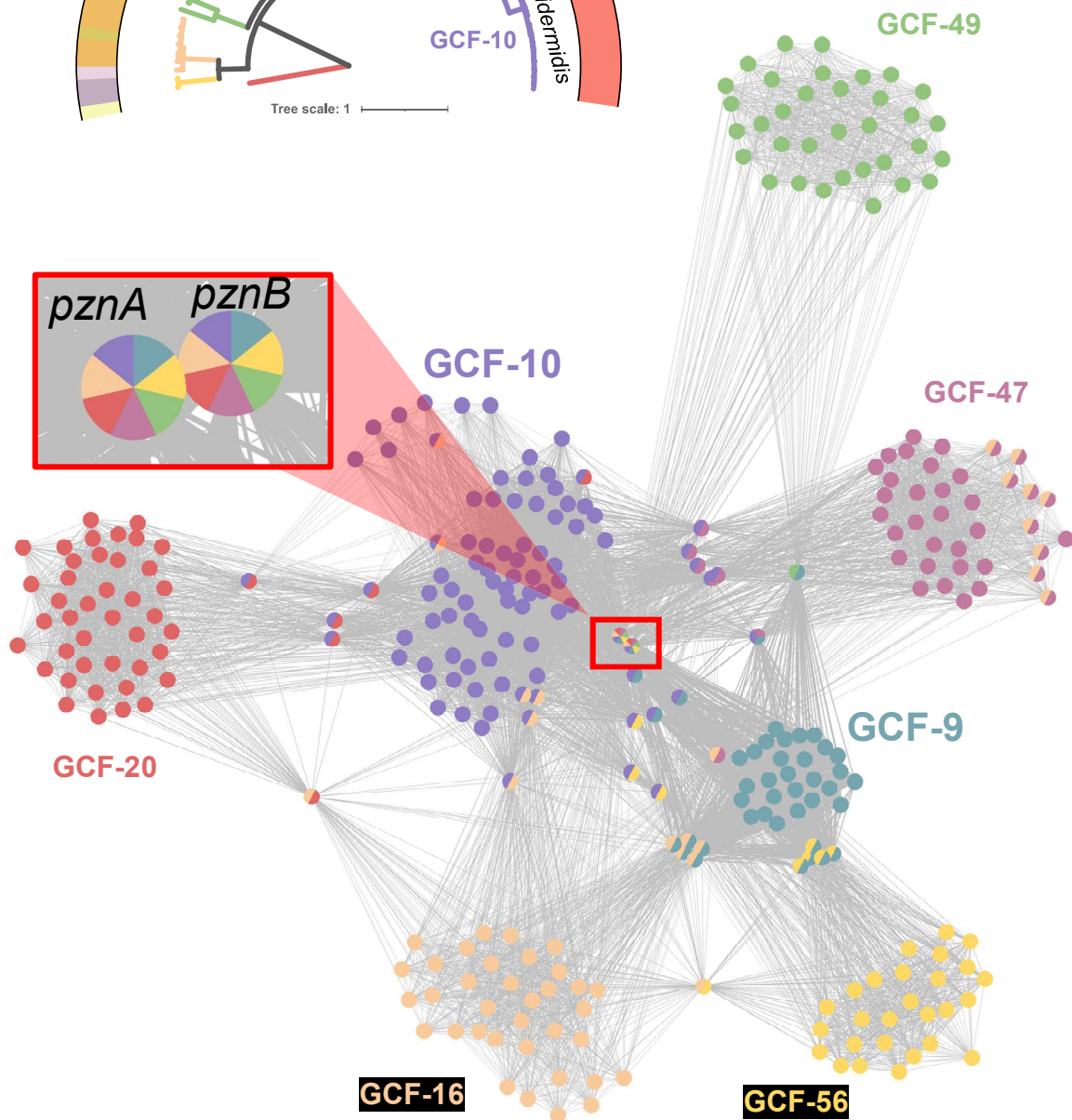
**Figure S5: Systematic identification of skin-associated BGC genes and signatures of intra-GCF horizontal gene transfer.** A maximum likelihood phylogeny was constructed from ribosomal protein encoding genes for **a)** 229 representative *Staphylococcus* genomes and **b)** 456 representative *Corynebacterium* genomes. Ancestral state reconstruction for GCF carriage was performed using maximum parsimony with the ACCTRAN algorithm and shown as piecharts for innernodes which encapsulate five or more species. Color strips to the right of the phylogenies correspond to the species classification of the genomes. The bar chart to the right of the species color strip corresponds to the number of metagenomes which were found to feature the species. The final track shows the presence of homolog groups found to be enriched in phylogenetic clades where  $\geq 80\%$  of genomes are classified as skin-associated species. IsaBGC-Divergence was used to calculate the Beta-RD statistic between pairs of genomes carrying a particular GCF in representative **c)** *Staphylococcus* and **d)** *Corynebacterium* genomes. Bayesian shrinking analysis was performed of raw Beta-RD calculates using STAN.



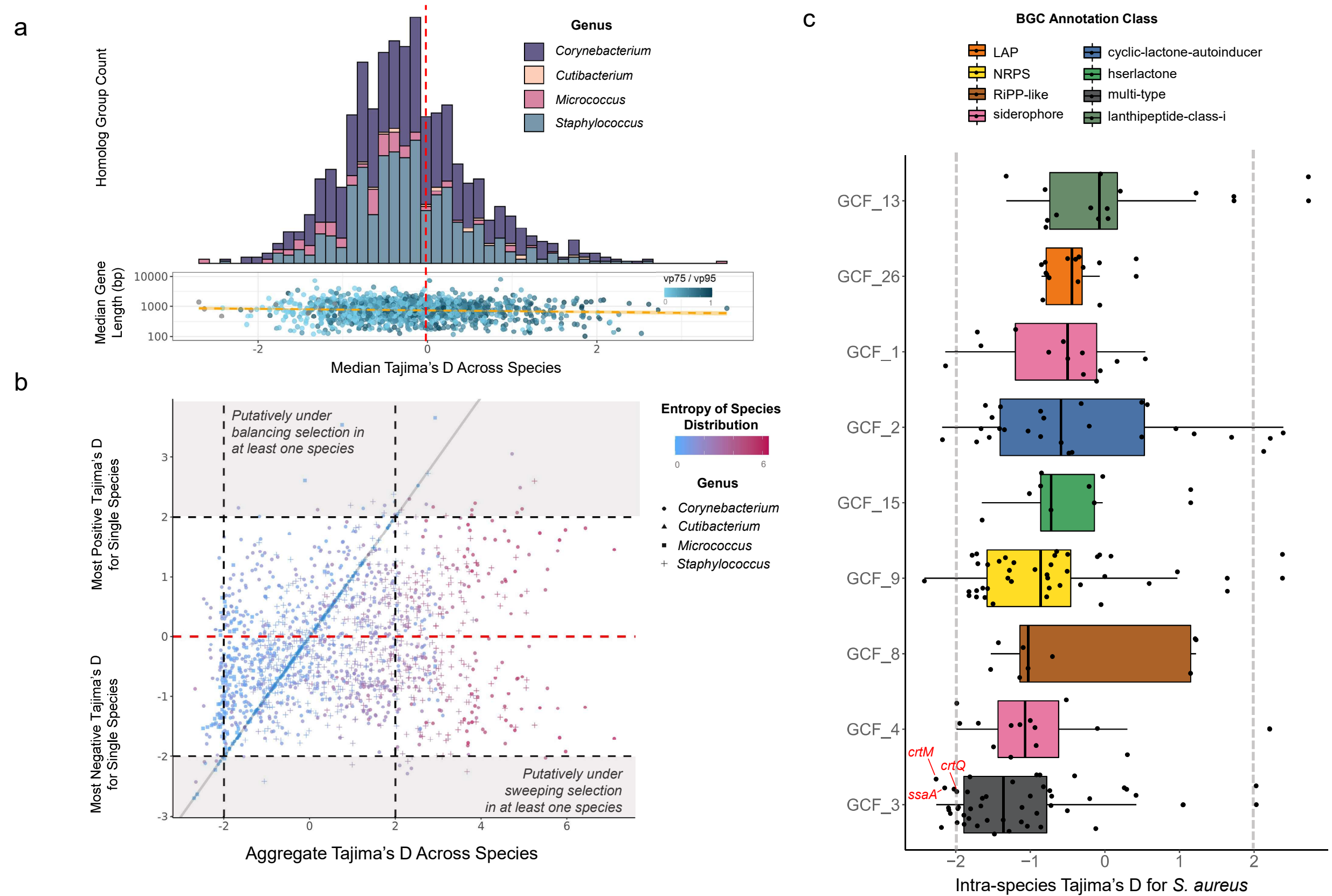
a



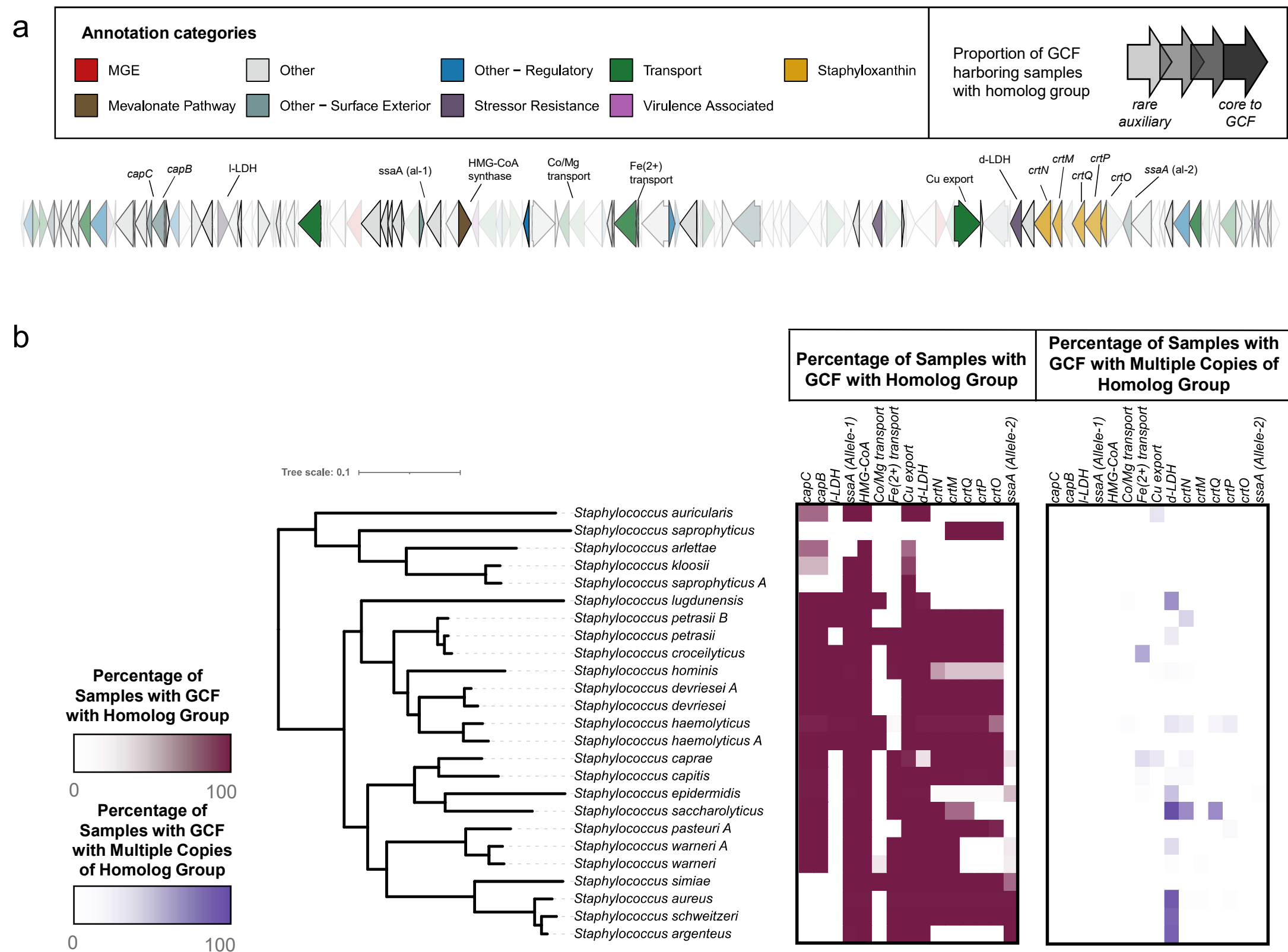
b



**Figure S6: *IsaBGC* expands the known diversity of pyrazinones.** a) A maximum-likelihood phylogeny was constructed from PznA and PznB showing the relationship between GCFs encoding for pyrazinones. Branch color corresponds to different GCFs and the outer circular color denotes the species classification for the genome from which the sequences were gathered. b) Homolog groups (nodes) across the seven pyrazinone encoding GCFs are depicted as a network where edges indicate that two homolog groups co-occur in the same GCF. The color of each node depicts the distinct pyrazinone GCFs in which the homolog group was found.



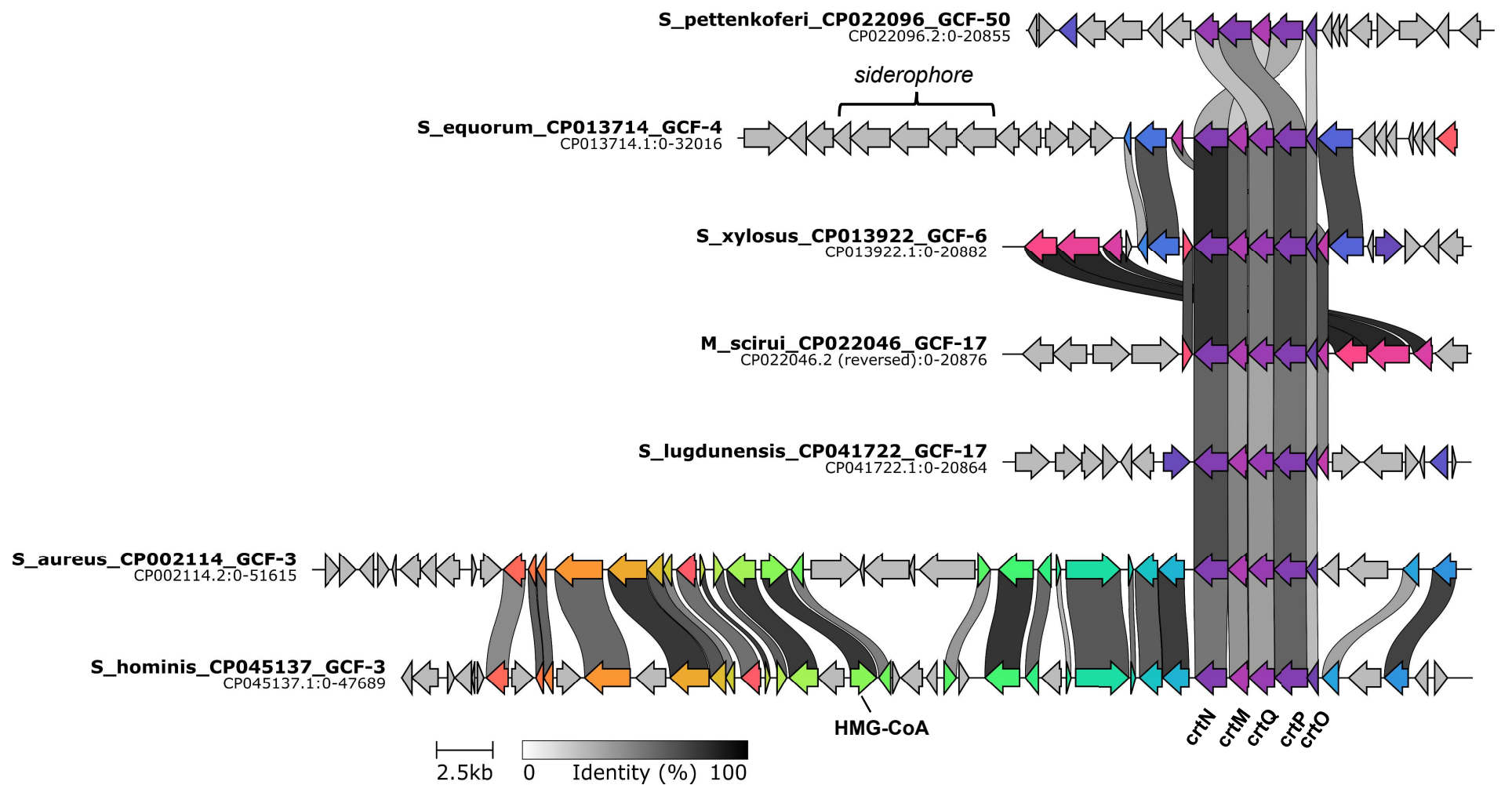
**Figure S7: Tajima's D statistic highlights GCFs and homolog groups under selective pressure.** **a)** For each homolog group found in a GCF, Tajima's D statistic was independently calculated per species and the median value across species was determined. The histogram of the median Tajima's D across species exhibits a normal distribution roughly centered around 0. The scatterplot below the histogram showcases that the median Tajima's D across species per homolog group exhibited no correlation with the median length of homolog groups. Each homolog group in the scatterplot is colored according to an analogous statistic to Tajima's D which is the ratio of sites along the homolog group's codon alignment where the major allele is found in  $\geq 75\%$  of sequences to sites where the major allele is found in  $\geq 95\%$  of sequences. **b)** The aggregate Tajima's D statistic was calculated over all sequences per homolog group and found to be biased by the number of species the homolog group was found in and thus the maximum and minimum Tajima's D per species was investigated instead. **c)** Intra-species Tajima's D calculations using 24 distinct *S. aureus* genomes highlight GCF-3, predicted to encode a hybrid terpene/T3PKS BGC, as feature seven homolog groups with Tajima's D below -2.0, including *crtM*, involved in staphyloxanthin biosynthesis, and *ssaA*, encoding a staphylococcal secretory antigen.



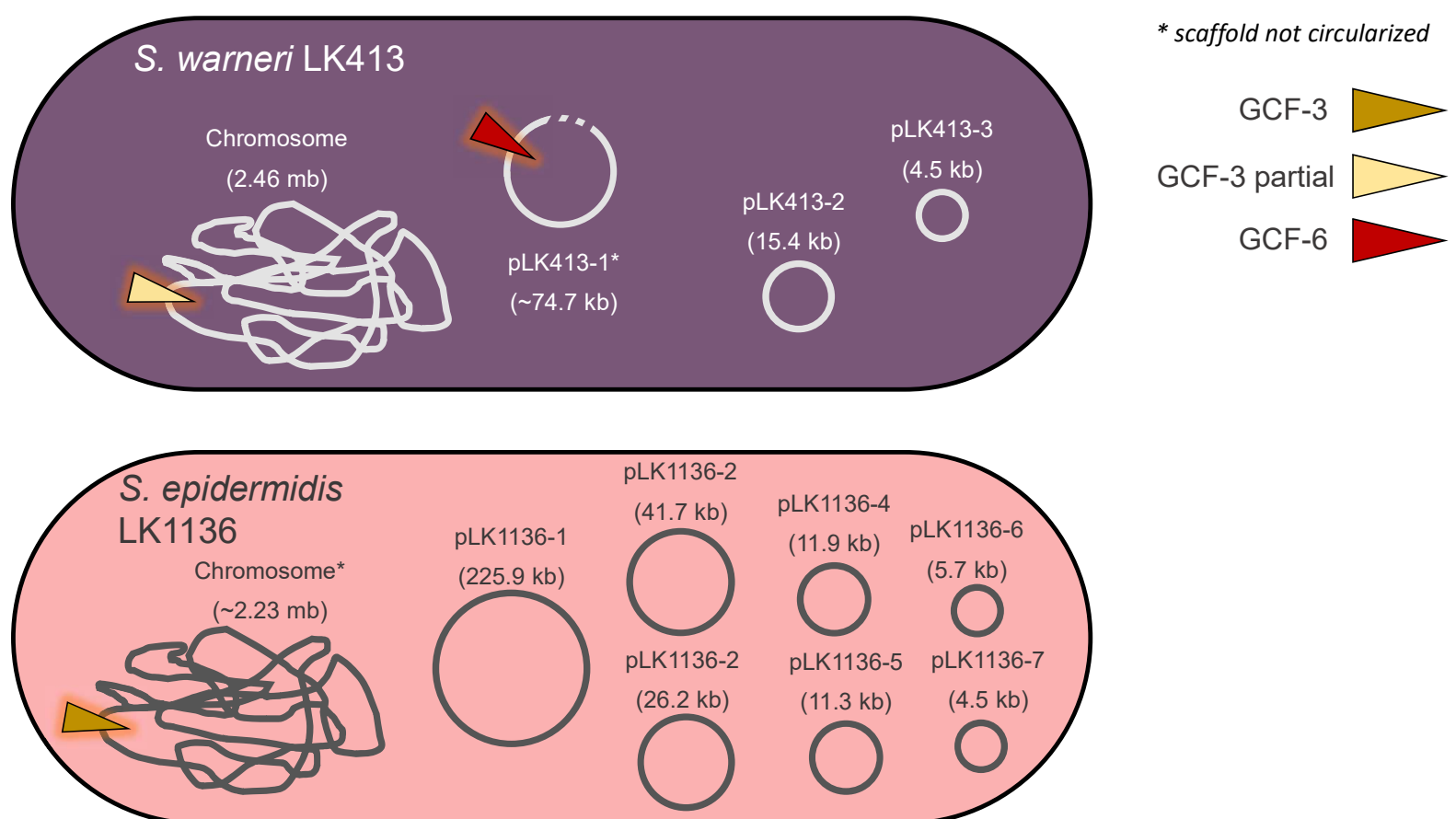
**Figure S8: Signatures of conservation and speciation in the surrounding context of staphyloxanthin biosynthesis encoding GCF-3.** GCF-3 is a predicted hybrid BGC which includes the staphyloxanthin biosynthesis encoding *crt* operon and the mevalonate pathway related enzyme hydroxymethylglutaryl-CoA synthase, a false positive detection due to homology with type-III polyketide synthases. **a)** A consensus schematic of GCF-3 is shown, generated from individual instances from 103 representative staphylococci found to feature it. Genes are colored according to broad annotation categories and transparency illustrates the proportion of GCF-3 carrying samples found to possess a specific homolog group. **b)** A maximum-likelihood phylogeny for the 25 species within the *S. aureus* / *S. epidermidis* clade found to carry GCF-3 is shown alongside three heatmaps depicting species-specific metrics for select homolog groups. The left heatmap showcases the percentage of the total species genomes found to carry select genes from GCF-3, including the staphyloxanthin encoding *crt* genes. The right heatmap depicts whether homolog groups are found in multiple-copies within the GCF-3 context for different species.



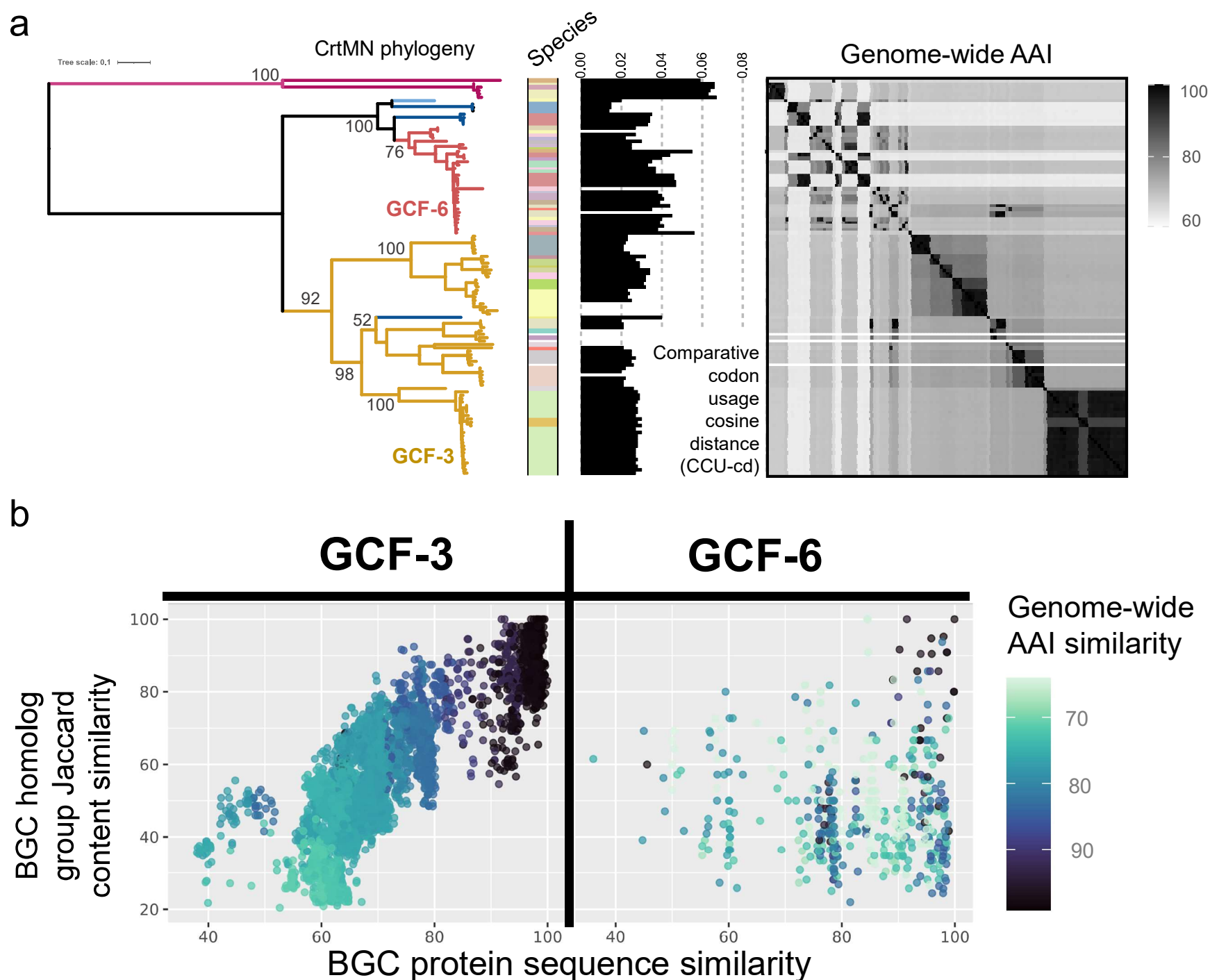
a



b



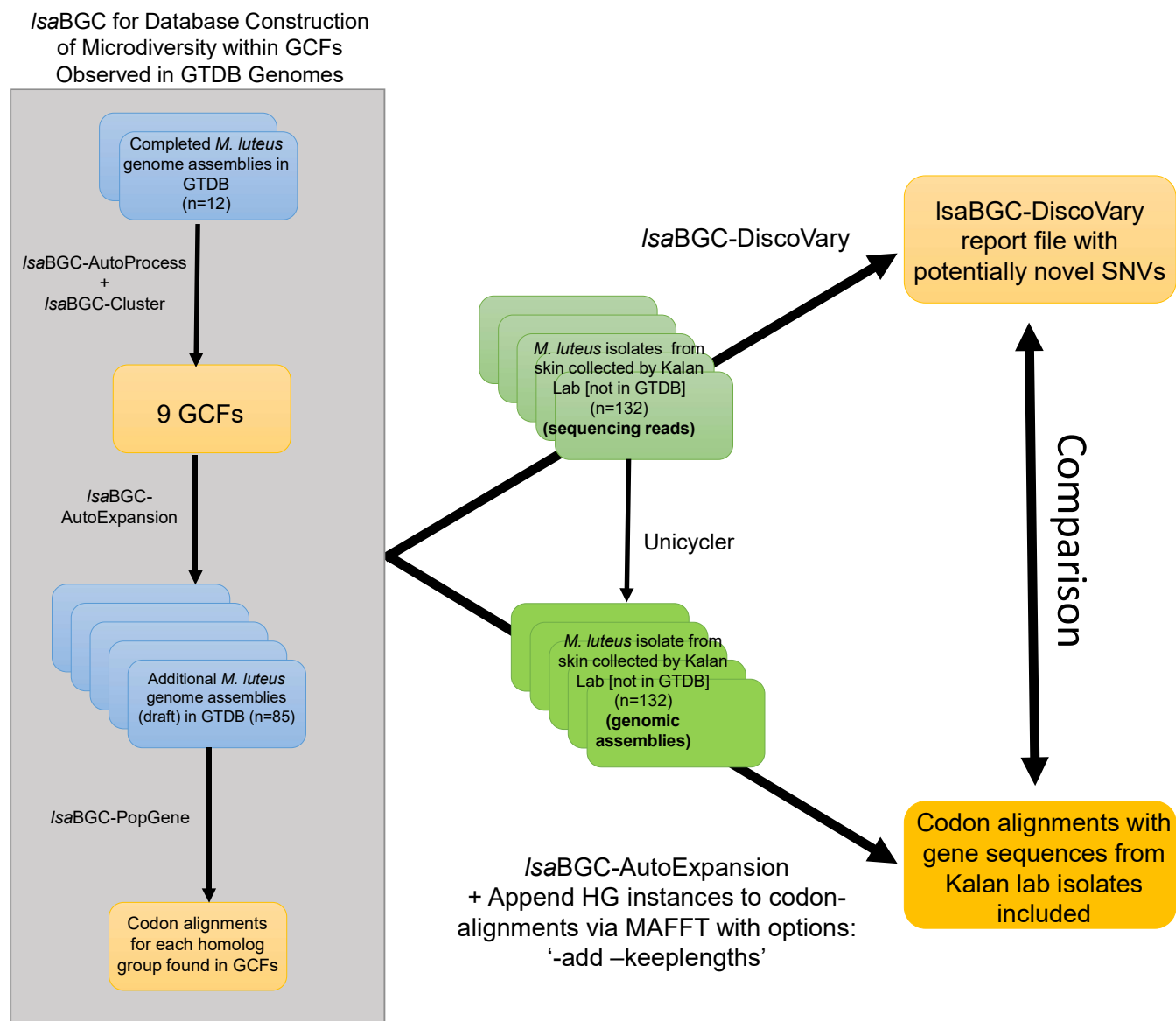
**Figure S9: The staphyloxanthin encoding *crt* operon is found in multiple GCFs.** **a)** Sequence and synteny comparisons between representative instances of GCFs featuring the *crt* operon encoding for staphyloxanthin were performed and illustrated using clinker. Based on phylogenetic analysis of CrtMN sequences (Figure 3b), the *crt* operon is likely misclassified as GCF-17 for *S. lugdunensis* and should instead be GCF-3. The cause of the misclassification is because the *crt* operon is in a different genomic context within the species as compared to other species in the *S. aureus/epidermidis* clade with GCF-3. **b)** Schematics of the near completed genomic assemblies for *S. epidermidis* LK1136 and *S. warneri* LK413 isolated from skin showcasing the location of staphyloxanthin encoding GCFs.



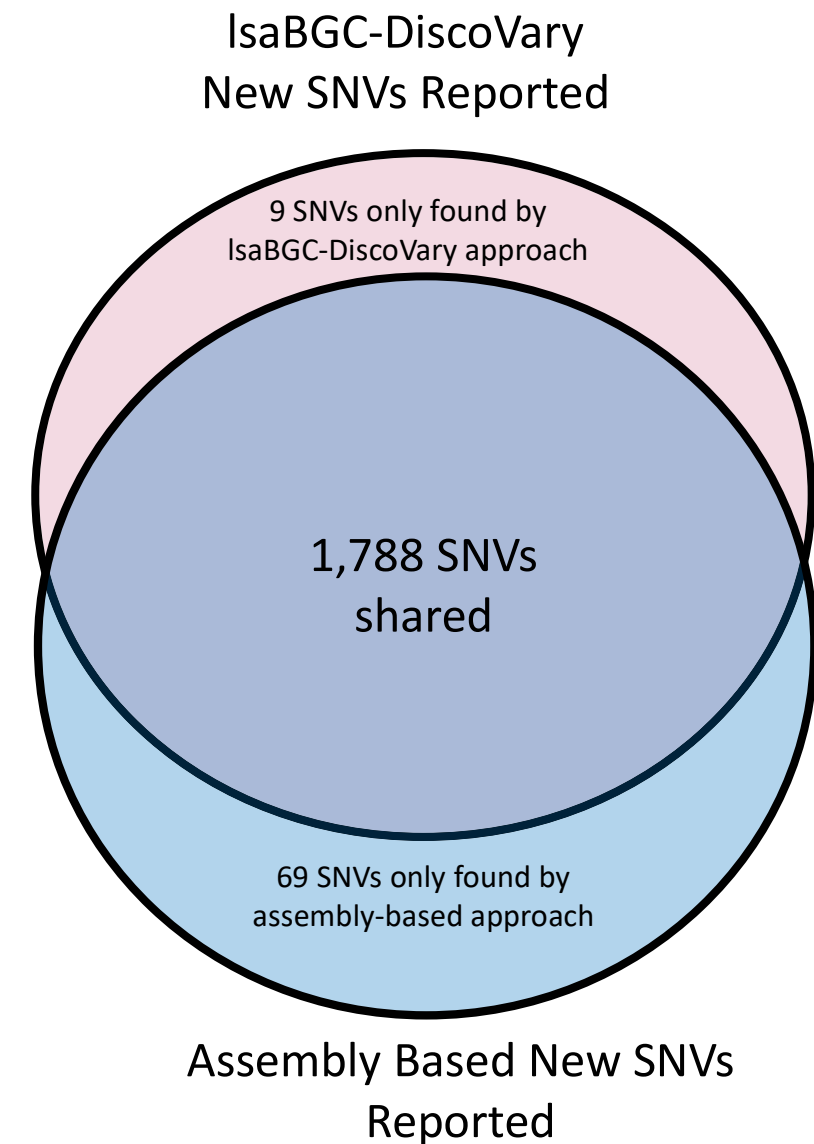
**Figure S10: Staphyloxanthin encoding GCF-6 exhibits signatures of mobilization.** **a)** A maximum-likelihood phylogeny was built from CrtM and CrtN protein alignments. Branch colors represent the GCF classification of the CrtMN sequences and bootstrap values are shown for key nodes where GCFs partition. The species the CrtMN sequence was extracted from is shown as a color strip followed by a bar plot depicting the comparative codon usage cosine distance (CCU-cd). CCU-cd was only calculated for the full five gene *crt* operon and represents the codon frequency dissimilarity with the codon frequency of the background genome. The heatmap showcases the genome-wide average amino acid identity (AAI) between pairs of genomes from which CrtMN sequences were gathered. **b)** AAI and shared homolog group content were calculated between pairs of GCF instances from different genomes for GCF-3 and GCF-6 individually. The coloring represents genome-wide AAI similarity.



a

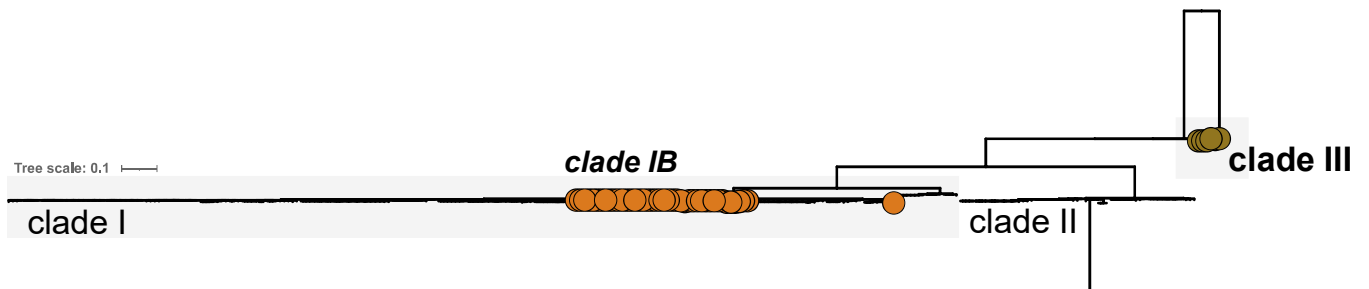


b

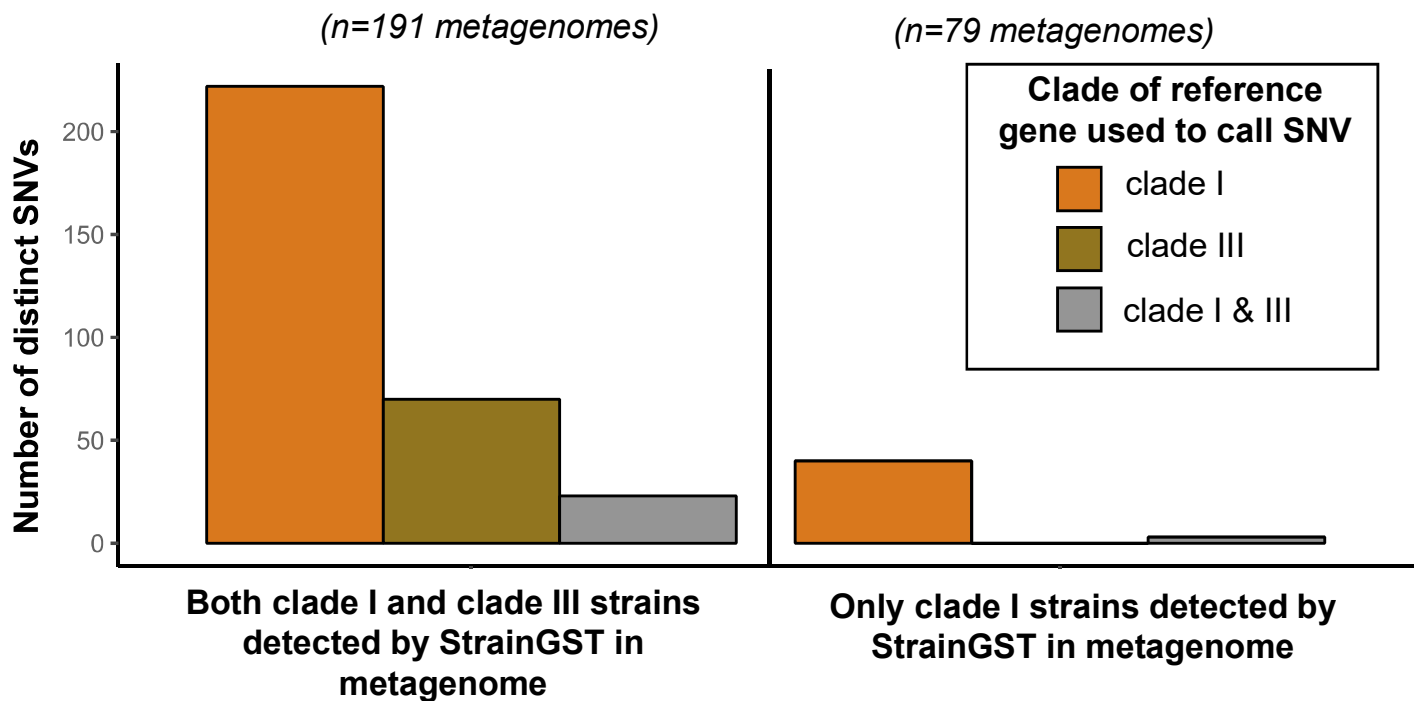


**Figure S11: Benchmarking *IsaBGC-DiscoVary* to assembly based SNV identification using single isolate genomic sequencing data for 132 *M. luteus*.** a) A schematic of the benchmarking setup for comparing *IsaBGC-DiscoVary* identification of novel SNVs from sequencing reads for 132 *M. luteus* compared to an assembly based identification of novel SNVs for the same isolates. Novel SNVs corresponded to alleles which were not previously represented at specific sites in homolog group codon alignments constructed from publicly available *M. luteus* genomes gathered from NCBI. b) A venn diagram showcasing the number of novel SNVs reported by *IsaBGC-DiscoVary* as compared to the novel SNVs found by the assembly-based approach.

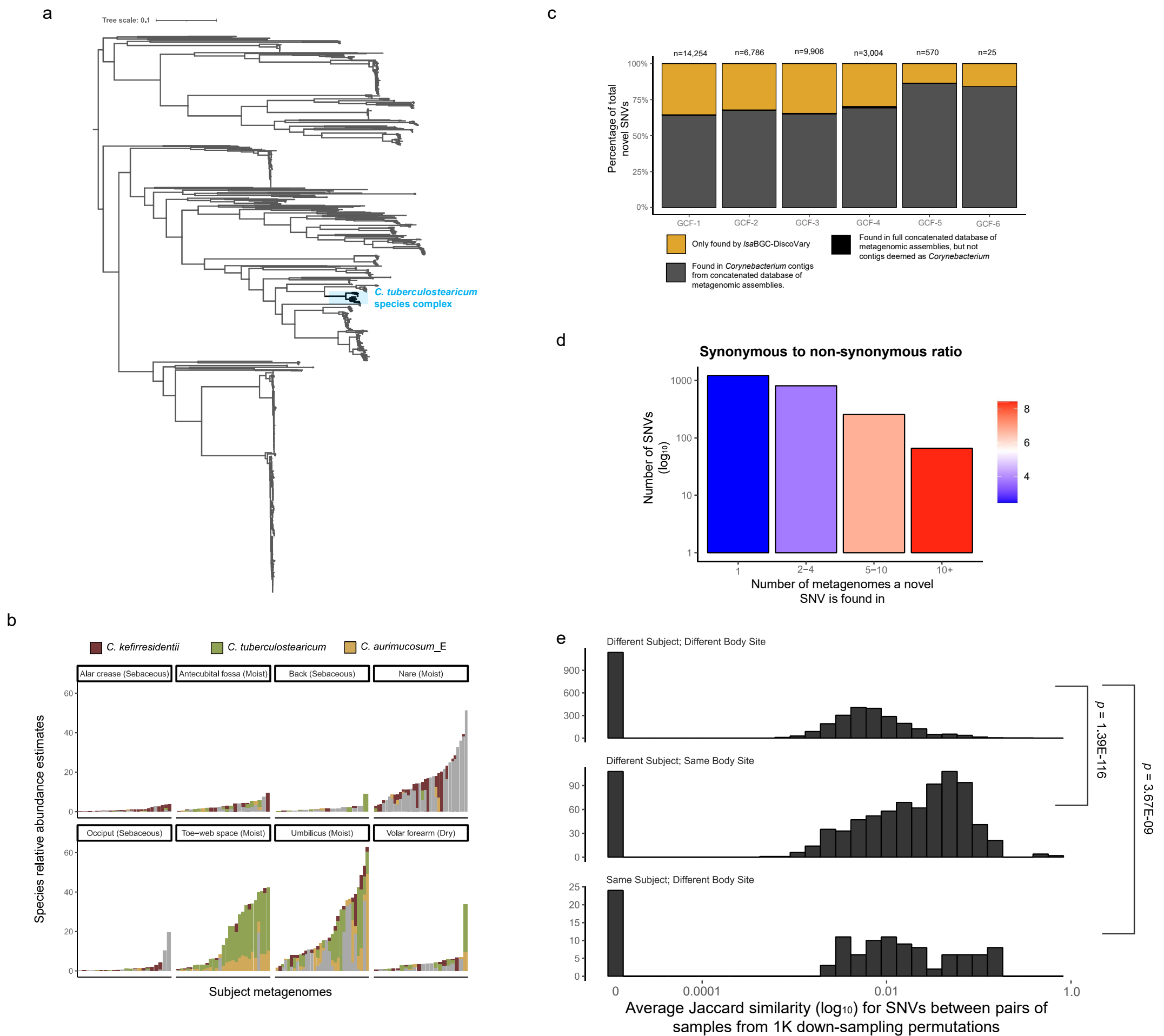
a



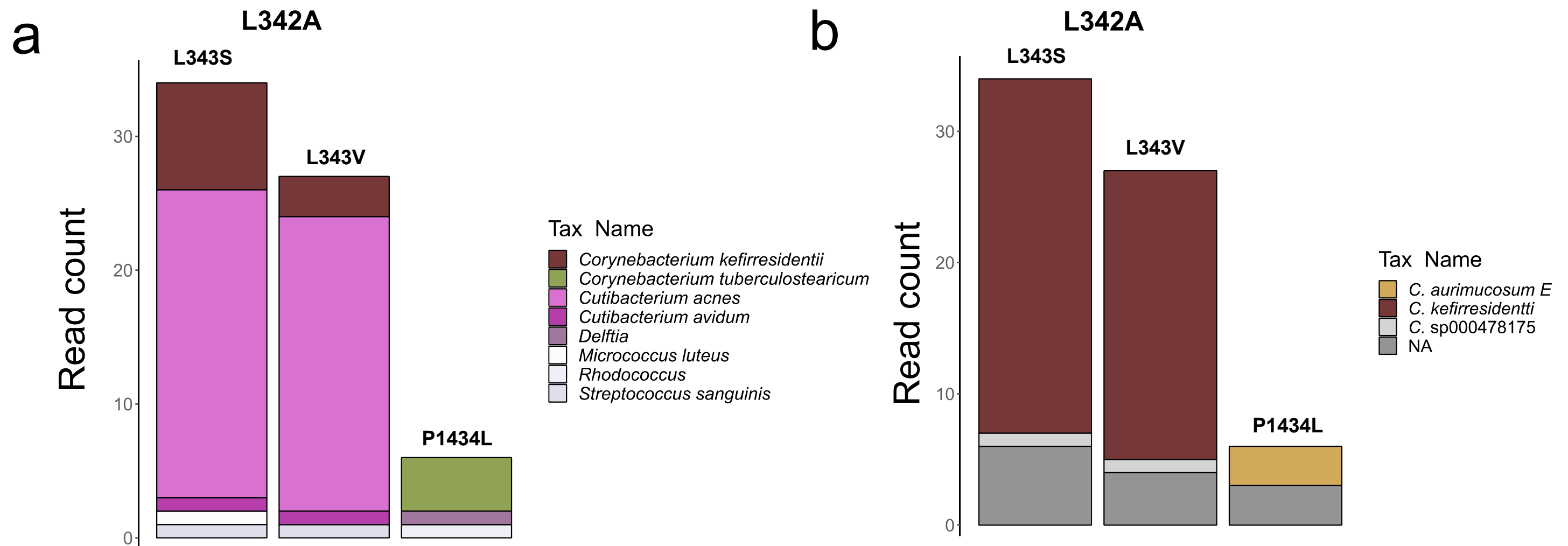
c



**Figure S12: IsaBGC-DiscoVary SNV reporting on BGCs is congruent with whole-genome strain detection tools.** a) The cutimycin encoding GCF is found primarily in clade I (subclade IB; orange) and clade III (tan) *C. acnes*. b) The number of distinct SNVs identified by DiscoVary (post-filtering) which were identified upon reference genes belonging to clade I, clade III, or both clades. Results are shown separately for metagenomes in which StrainGST detected both clade I and clade III strains and metagenomes in which StrainGST only detected clade I strain(s) as present.



**Figure S13: The *C. tuberculostrictum* species complex is highly prevalent across skin metagenomes.** **a)** A more resolute and comprehensive phylogeny of 1,118 *Corynebacterium*, including all 22 genomes found to belong to the *C. tuberculostrictum* species complex in this study, was constructed using a set of 138 genes determined as largely single-copy in Actinomycetota with GToTree. The *C. tuberculostrictum* species complex was found to form a monophyletic clade. **b)** The relative abundance of *Corynebacterium* and representative strains of the *C. tuberculostrictum* species complex in the StrainGST database are shown across skin metagenomes from different body sites and individuals. **c)** Assessment of novel SNVs identified by IsaBGC-DiscoVary, post-filtering, for each BGC from the *C. tuberculostrictum* species complex for presence within metagenomic assemblies. **d)** Novel SNVs were tabulated by the number of metagenomes they were observed in. The color of bars corresponds to the ratio of suspected synonymous to non-synonymous SNVs for the sets of novel SNVs. **e)** For metagenomic samples where 30 or more novel SNVs were identified, a multi-iteration, down-sampling based approach was used to compute the average Jaccard similarity for number of shared novel SNVs between pairs of metagenomes. Pairwise comparisons of metagenomes were categorized by whether samples were from the same body-site or subject.



**Figure S14: Validation of taxonomic origin for reads supporting the presence of non-synonymous SNVs at highly conserved sites as identified by *IsaBGC-DiscoVary*.** **a)** Reads supporting the existence of three novel SNVs in highly conserved sites of the mycolic acid biosynthesis polyketide synthase (OG0001691) were classified taxonomically with Kraken2. Despite many of the reads supporting novel SNV existence being classified as *Cutibacterium*, paired-end alignment to a comprehensive database of all *Cutibacterium* genomes featured in GTDB R202 showed that none aligned concordantly. **b)** Of the reads supporting the existence of novel SNVs, 85.07% aligned as concordant pairs to a comprehensive database of all *Corynebacterium* genomes, with the majority aligning to species belonging to the *C. tuberculostearicum* species complex.