# Welcome to the zol (&fai) wiki!

Interested in finding homologous instances of a gene-cluster in target genomes? Check out how to prepare target genomes and perform such searches on the prepTG and fai wiki pages.

Interested in evolutionary analysis of a set of related gene-clusters in GenBank format with CDS features available? Check out the zol wiki page.

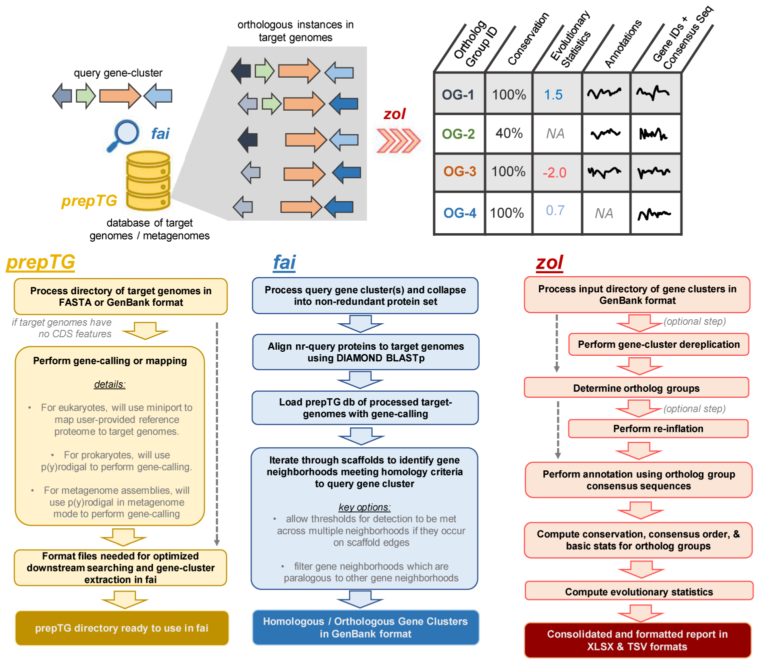
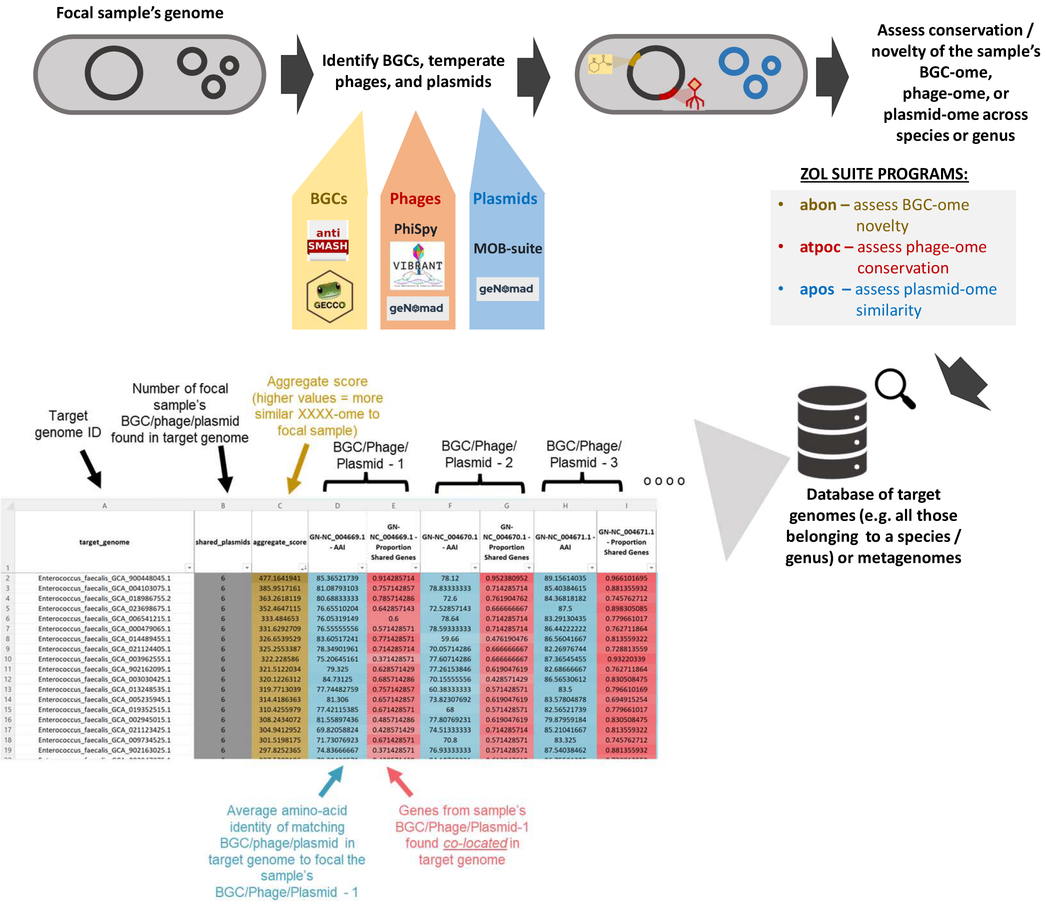

orthologous instances in target genomes
query gene-cluster
fai
prepTG
database of target genomes / metagenomes
zol
Ortholog Group ID
Conservation
Evolutionary Statistics
Annotations
Gene IDs + Consensus Seq
OG-1
100%
1.5
OG-2
40%
NA
OG-3
100%
-2.0
OG-4
100%
0.7
NA
prepTG
Process directory of target genomes in FASTA or GenBank format
if target genomes have no CDS features
Perform gene-calling or mapping
details:
• For eukaryotes, will use miniport to map user-provided reference proteome to target genomes.
• For prokaryotes, will use p(y)rodigal to perform gene-calling.
• For metagenome assemblies, will use p(y)rodigal in metagenome mode to perform gene-calling
Format files needed for optimized downstream searching and gene-cluster extraction in fai
prepTG directory ready to use in fai
fai
Process query gene cluster(s) and collapse into non-redundant protein set
Align nr-query proteins to target genomes using DIAMOND BLASTp
Load prepTG db of processed target-genomes with gene-calling
Iterate through scaffolds to identify gene neighborhoods meeting homology criteria to query gene cluster
key options:
• allow thresholds for detection to be met across multiple neighborhoods if they occur on scaffold edges
• filter gene neighborhoods which are paralogous to other gene neighborhoods
Homologous / Orthologous Gene Clusters in GenBank format
zol
Process input directory of gene clusters in GenBank format
(optional step)
Perform gene-cluster dereplication
Determine ortholog groups
(optional step)
Perform re-inflation
Perform annotation using ortholog group consensus sequences
Compute conservation, consensus order, & basic stats for ortholog groups
Compute evolutionary statistics
Consolidated and formatted report in XLSX & TSV formats

Figure 1: zol_overview

## Contents


Focal sample's genome
Identify BGCs, temperate phages, and plasmids
Assess conservation / novelty of the sample's BGC-ome, phage-ome, or plasmid-ome across species or genus
BGCs
anti SMASH
GECCO
Phages
PhiSpy
VIBRANT
geNomad
Plasmids
MOB-suite
geNomad
ZOL SUITE PROGRAMS:
abon – assess BGC-ome novelty
atpoc – assess phage-ome conservation
apos – assess plasmid-ome similarity
Database of target genomes (e.g. all those belonging to a species / genus) or metagenomes
Target genome ID
Number of focal sample's BGC/phage/plasmid found in target genome
Aggregate score (higher values = more similar XXXX-ome to focal sample)
BGC/Phage/ Plasmid - 1
BGC/Phage/ Plasmid - 2
BGC/Phage/ Plasmid - 3
o o o o
Average amino-acid identity of matching BGC/phage/plasmid in target genome to focal the sample's BGC/Phage/Plasmid - 1
Genes from sample's BGC/Phage/Plasmid-1 found co-located in target genome

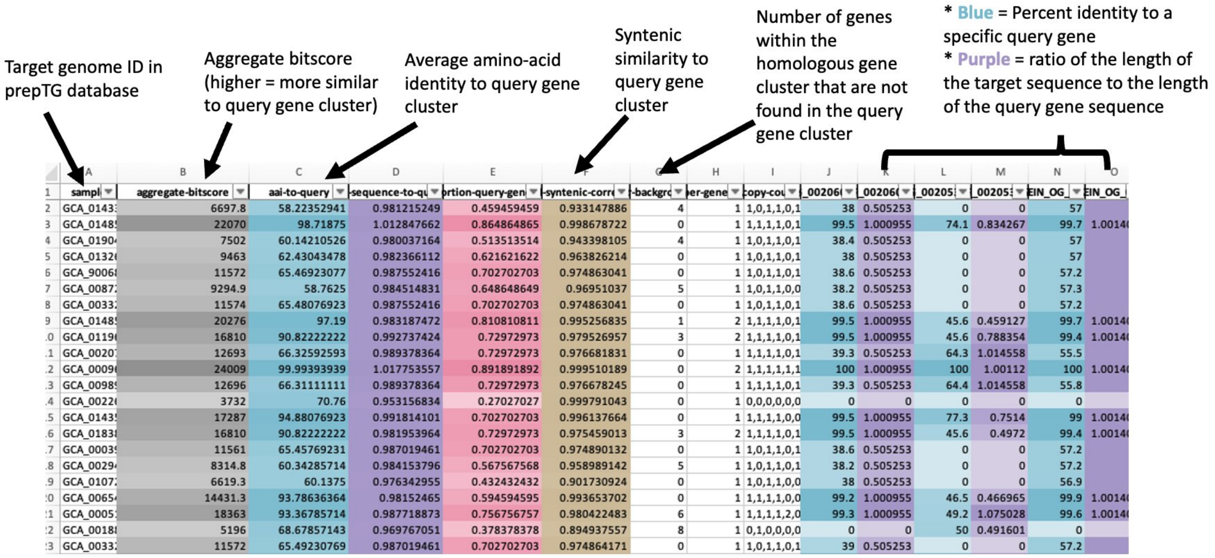
| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | target_genome | shared_plasmids | aggregate_score | GN-NC_004669.1 - AAI | GN-NC_004669.1 - Proportion Shared Genes | GN-NC_004670.1 - AAI | GN-NC_004670.1 - Proportion Shared Genes | GN-NC_004671.1 - AAI | GN-NC_004671.1 - Proportion Shared Genes |
| 2 | Enterococcus_faecalis_GCA_900448045.1 | 6 | 477.1641941 | 85.36521739 | 0.914285714 | 78.12 | 0.952380952 | 89.15614035 | 0.966101695 |
| 3 | Enterococcus_faecalis_GCA_004103075.1 | 6 | 385.9517161 | 81.08793103 | 0.757142857 | 78.83333333 | 0.714285714 | 85.40384615 | 0.881355932 |
| 4 | Enterococcus_faecalis_GCA_018986755.2 | 6 | 363.2618119 | 80.68833333 | 0.785714286 | 72.6 | 0.761904762 | 84.36818182 | 0.745762712 |
| 5 | Enterococcus_faecalis_GCA_023698675.1 | 6 | 352.4647115 | 76.65510204 | 0.642857143 | 72.52857143 | 0.666666667 | 87.5 | 0.898305085 |
| 6 | Enterococcus_faecalis_GCA_006541215.1 | 6 | 333.484653 | 76.05319149 | 0.6 | 78.64 | 0.714285714 | 83.29130435 | 0.779661017 |
| 7 | Enterococcus_faecalis_GCA_000479065.1 | 6 | 331.6292709 | 76.55555556 | 0.571428571 | 78.59333333 | 0.714285714 | 86.44222222 | 0.762711864 |
| 8 | Enterococcus_faecalis_GCA_014489455.1 | 6 | 326.6539529 | 83.60517241 | 0.771428571 | 59.66 | 0.476190476 | 86.56041667 | 0.813559322 |
| 9 | Enterococcus_faecalis_GCA_021124405.1 | 6 | 325.2553387 | 78.34901961 | 0.714285714 | 70.05714286 | 0.666666667 | 82.26976744 | 0.728813559 |
| 10 | Enterococcus_faecalis_GCA_003962555.1 | 6 | 322.228586 | 75.20645161 | 0.371428571 | 77.60714286 | 0.666666667 | 87.36545455 | 0.93220339 |
| 11 | Enterococcus_faecalis_GCA_902162095.1 | 6 | 321.5122034 | 79.325 | 0.628571429 | 77.26153846 | 0.619047619 | 82.68666667 | 0.762711864 |
| 12 | Enterococcus_faecalis_GCA_003030425.1 | 6 | 320.1226312 | 84.73125 | 0.685714286 | 70.15555556 | 0.428571429 | 86.56530612 | 0.830508475 |
| 13 | Enterococcus_faecalis_GCA_013248535.1 | 6 | 319.7713039 | 77.74482759 | 0.757142857 | 60.38333333 | 0.571428571 | 83.5 | 0.796610169 |
| 14 | Enterococcus_faecalis_GCA_005235945.1 | 6 | 314.4186363 | 81.306 | 0.657142857 | 73.82307692 | 0.619047619 | 83.57804878 | 0.694915254 |
| 15 | Enterococcus_faecalis_GCA_019352515.1 | 6 | 310.4255979 | 77.42115385 | 0.671428571 | 68 | 0.571428571 | 82.56521739 | 0.779661017 |
| 16 | Enterococcus_faecalis_GCA_002945015.1 | 6 | 308.2434072 | 81.55897436 | 0.485714286 | 77.80769231 | 0.619047619 | 79.87959184 | 0.830508475 |
| 17 | Enterococcus_faecalis_GCA_021123425.1 | 6 | 304.9412952 | 69.82058824 | 0.428571429 | 74.51333333 | 0.714285714 | 85.21041667 | 0.813559322 |
| 18 | Enterococcus_faecalis_GCA_009734525.1 | 6 | 301.5198175 | 71.73076923 | 0.671428571 | 70.8 | 0.571428571 | 83.325 | 0.745762712 |
| 19 | Enterococcus_faecalis_GCA_902163025.1 | 6 | 297.8252365 | 74.83666667 | 0.371428571 | 76.93333333 | 0.571428571 | 87.54038462 | 0.881355932 |

Figure 2: zol_utility_programs_overview

## fai results

The two major results of fai are:

1. A directory of homologous or orthologous gene clusters (in GenBank format) from target genomes to the query gene cluster provided as input to the program. These can be found in the directory `fai_Results/Final_Results/Homologous_Gene_Cluster_GenBanks/`
2. An XLSX spreadsheet which allows users to assess homologous hits to the query gene cluster at scale. Most columns feature automatic conditional formatting to ease user assessment of quantitative fields.



Target genome ID in prepTG database

Aggregate bitscore (higher = more similar to query gene cluster)

Average amino-acid identity to query gene cluster

Syntenic similarity to query gene cluster

Number of genes within the homologous gene cluster that are not found in the query gene cluster

* Blue = Percent identity to a specific query gene
* Purple = ratio of the length of the target sequence to the length of the query gene sequence

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | sampl | aggregate-bitscore | aai-to-query | -sequence-to-qu | ortion-query-gen | -syntenic-corre | -backgrc | ier-gene | copy-cou | _00206( | _00206( | _00205: | _00205: | EIN_OG_ | EIN_OG_ |
| 2 | GCA_0143: | 6697.8 | 58.22352941 | 0.981215249 | 0.459459459 | 0.933147886 | 4 | 1 | 1,0,1,1,0,1 | 38 | 0.505253 | 0 | 0 | 57 | |
| 3 | GCA_0148! | 22070 | 98.71875 | 1.012847662 | 0.864864865 | 0.998678722 | 0 | 1 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 74.1 | 0.834267 | 99.7 | 1.0014( |
| 4 | GCA_0190 | 7502 | 60.14210526 | 0.980037164 | 0.513513514 | 0.943398105 | 4 | 1 | 1,0,1,1,0,1 | 38.4 | 0.505253 | 0 | 0 | 57 | |
| 5 | GCA_0132( | 9463 | 62.43043478 | 0.982366112 | 0.621621622 | 0.963826214 | 0 | 1 | 1,0,1,1,0,1 | 38 | 0.505253 | 0 | 0 | 57 | |
| 6 | GCA_9006 | 11572 | 65.46923077 | 0.987552416 | 0.702702703 | 0.974863041 | 0 | 1 | 1,0,1,1,0,1 | 38.6 | 0.505253 | 0 | 0 | 57.2 | |
| 7 | GCA_0087: | 9294.9 | 58.7625 | 0.984514831 | 0.648648649 | 0.96951037 | 5 | 1 | 1,0,1,1,0,0 | 38.2 | 0.505253 | 0 | 0 | 57.3 | |
| 8 | GCA_0033: | 11574 | 65.48076923 | 0.987552416 | 0.702702703 | 0.974863041 | 0 | 1 | 1,0,1,1,0,1 | 38.6 | 0.505253 | 0 | 0 | 57.2 | |
| 9 | GCA_0148! | 20276 | 97.19 | 0.983187472 | 0.810810811 | 0.995256835 | 1 | 2 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 45.6 | 0.459127 | 99.7 | 1.0014( |
| .0 | GCA_0119( | 16810 | 90.82222222 | 0.992737424 | 0.72972973 | 0.979526957 | 3 | 2 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 45.6 | 0.788354 | 99.4 | 1.0014( |
| .1 | GCA_0020 | 12693 | 66.32592593 | 0.989378364 | 0.72972973 | 0.976681831 | 0 | 1 | 1,1,1,1,0,1 | 39.3 | 0.505253 | 64.3 | 1.014558 | 55.5 | |
| .2 | GCA_0009( | 24009 | 99.99393939 | 1.017753557 | 0.891891892 | 0.999510189 | 0 | 2 | 1,1,1,1,1,1 | 100 | 1.000955 | 100 | 1.00112 | 100 | 1.0014( |
| .3 | GCA_0098! | 12696 | 66.31111111 | 0.989378364 | 0.72972973 | 0.976678245 | 0 | 1 | 1,1,1,1,0,1 | 39.3 | 0.505253 | 64.4 | 1.014558 | 55.8 | |
| .4 | GCA_0022( | 3732 | 70.76 | 0.953156834 | 0.27027027 | 0.999791043 | 0 | 1 | 0,0,0,0,0,0 | 0 | 0 | 0 | 0 | 0 | |
| .5 | GCA_0143! | 17287 | 94.88076923 | 0.991814101 | 0.702702703 | 0.996137664 | 0 | 1 | 1,1,1,1,0,0 | 99.5 | 1.000955 | 77.3 | 0.7514 | 99 | 1.0014( |
| .6 | GCA_0183 | 16810 | 90.82222222 | 0.981953964 | 0.72972973 | 0.975459013 | 3 | 2 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 45.6 | 0.4972 | 99.4 | 1.0014( |
| .7 | GCA_0003! | 11561 | 65.45769231 | 0.987019461 | 0.702702703 | 0.974890132 | 0 | 1 | 1,0,1,1,0,1 | 38.6 | 0.505253 | 0 | 0 | 57.2 | |
| .8 | GCA_0029 | 8314.8 | 60.34285714 | 0.984153796 | 0.567567568 | 0.958989142 | 5 | 1 | 1,0,1,1,0,1 | 38.2 | 0.505253 | 0 | 0 | 57.2 | |
| .9 | GCA_0107: | 6619.3 | 60.1375 | 0.976342955 | 0.432432432 | 0.901730924 | 0 | 1 | 1,0,1,1,0,0 | 38 | 0.505253 | 0 | 0 | 56.9 | |
| :0 | GCA_0065 | 14431.3 | 93.78636364 | 0.98152465 | 0.594594595 | 0.993653702 | 0 | 1 | 1,1,1,1,0,0 | 99.2 | 1.000955 | 46.5 | 0.466965 | 99.9 | 1.0014( |
| :1 | GCA_0005: | 18363 | 93.36785714 | 0.987718873 | 0.756756757 | 0.980422483 | 6 | 1 | 1,1,1,1,2,0 | 99.3 | 1.000955 | 49.2 | 1.075028 | 99.6 | 1.0014( |
| :2 | GCA_0018 | 5196 | 68.67857143 | 0.969767051 | 0.378378378 | 0.894937557 | 8 | 1 | 0,1,0,0,0,0 | 0 | 0 | 50 | 0.491601 | 0 | |
| :3 | GCA_0033: | 11572 | 65.49230769 | 0.987019461 | 0.702702703 | 0.974864171 | 0 | 1 | 1,0,1,1,0,1 | 39 | 0.505253 | 0 | 0 | 57.2 | |

Figure 1: fai_results

In addition, certain visuals are generated to help with visual validation of detected gene clusters in target genomes as truly being homologous or orthologous to the query gene cluster. Check out the page more info on fai for details on plots that fai can generate.

## zol results

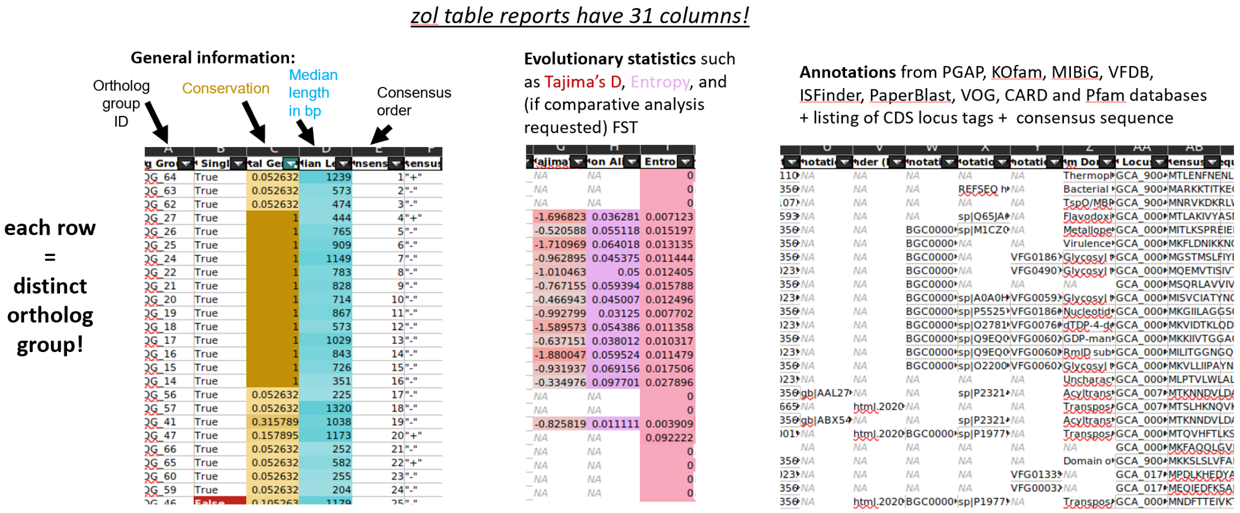
zol takes as input a set of related gene clusters in GenBank format and produces primarily one result file an XLSX spreadsheet which shows ortholog groups as rows and details on conservation, various evolutionary stats, and annotation info from multiple databases as the columns. Quantitative columns are automatically color-formatted. It is sorted by default in the consensus order genes occur in within gene clusters.

It can also be used to "dereplicate" gene clusters which can then be used for clinker analysis, for details on how to do that, check out the tutorial wiki page.

## abon, atpoc, and apos results

Similar to fai and zol's major results, the results from abon, atpoc, and apos also primarily produce XLSX spreadsheets. On the first tab of their resulting XLSX spreadsheet, is an overview of the focal sample's BGC, phage, or plasmid predictions:

Then on the second tab, the coverage of the focal sample's BGC-ome, phage-ome, or plasmid-ome across the genomes in the target genomes database is shown:



*zol table reports have 31 columns!*

**General information:**

Ortholog group ID

Conservation

Median length in bp

Consensus order

each row = distinct ortholog group!

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| g Gro | Singl | tal Ge | lian Le | nsens | ensus |
| G_64 | True | 0.052632 | 1239 | 1 | "+" |
| G_63 | True | 0.052632 | 573 | 2 | "-" |
| G_62 | True | 0.052632 | 474 | 3 | "-" |
| G_27 | True | 1 | 444 | 4 | "+" |
| G_26 | True | 1 | 765 | 5 | "-" |
| G_25 | True | 1 | 909 | 6 | "-" |
| G_24 | True | 1 | 1149 | 7 | "-" |
| G_22 | True | 1 | 783 | 8 | "-" |
| G_21 | True | 1 | 828 | 9 | "-" |
| G_20 | True | 1 | 714 | 10 | "-" |
| G_19 | True | 1 | 867 | 11 | "-" |
| G_18 | True | 1 | 573 | 12 | "-" |
| G_17 | True | 1 | 1029 | 13 | "-" |
| G_16 | True | 1 | 843 | 14 | "-" |
| G_15 | True | 1 | 726 | 15 | "-" |
| G_14 | True | 1 | 351 | 16 | "-" |
| G_56 | True | 0.052632 | 225 | 17 | "-" |
| G_57 | True | 0.052632 | 1320 | 18 | "-" |
| G_41 | True | 0.315789 | 1038 | 19 | "-" |
| G_47 | True | 0.157895 | 1173 | 20 | "+" |
| G_66 | True | 0.052632 | 252 | 21 | "-" |
| G_65 | True | 0.052632 | 582 | 22 | "+" |
| G_60 | True | 0.052632 | 255 | 23 | "-" |
| G_59 | True | 0.052632 | 204 | 24 | "-" |
| G_46 | False | 0.105263 | 1179 | 25 | "-" |

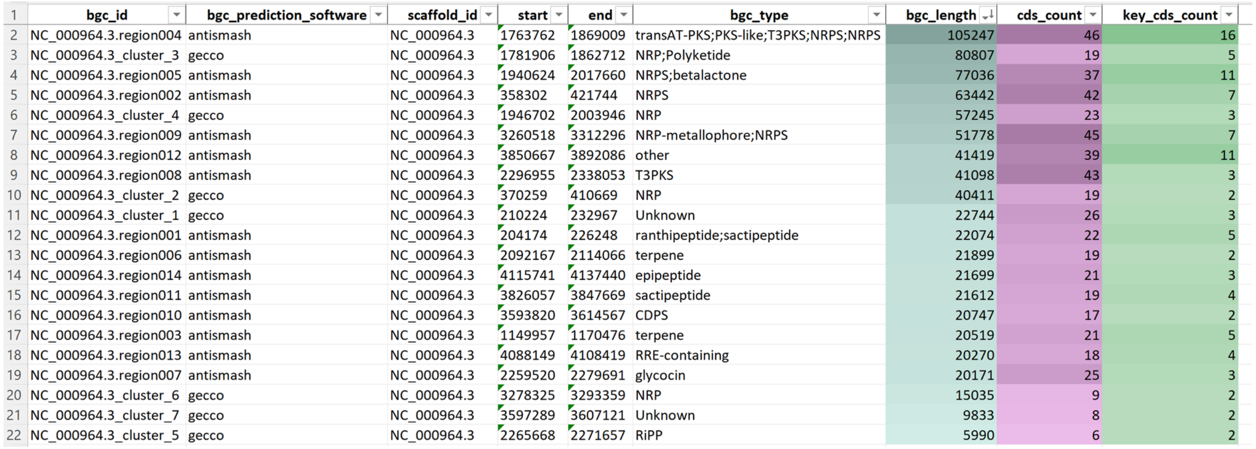
**Evolutionary statistics** such as Tajima's D, Entropy, and (if comparative analysis requested) FST

| G | H | I |
|---|---|---|
| ajima | on Ali | Entro |
| NA | NA | 0 |
| NA | NA | 0 |
| NA | NA | 0 |
| -1.696823 | 0.036281 | 0.007123 |
| -0.520588 | 0.055118 | 0.015197 |
| -1.710969 | 0.064018 | 0.013135 |
| -0.962895 | 0.045375 | 0.011444 |
| -1.010463 | 0.05 | 0.012405 |
| -0.767155 | 0.059394 | 0.015788 |
| -0.466943 | 0.045007 | 0.012496 |
| -0.992799 | 0.03125 | 0.007702 |
| -1.589573 | 0.054386 | 0.011358 |
| -0.637151 | 0.038012 | 0.010317 |
| -1.880047 | 0.059524 | 0.011479 |
| -0.931937 | 0.069156 | 0.017506 |
| -0.334976 | 0.097701 | 0.027896 |
| NA | NA | 0 |
| NA | NA | 0 |
| -0.825819 | 0.011111 | 0.003909 |
| NA | NA | 0.092222 |
| NA | NA | 0 |
| NA | NA | 0 |
| NA | NA | 0 |
| NA | NA | 0 |

**Annotations** from PGAP, KOfam, MIBiG, VFDB, ISFinder, PaperBlast, VOG, CARD and Pfam databases + listing of CDS locus tags + consensus sequence

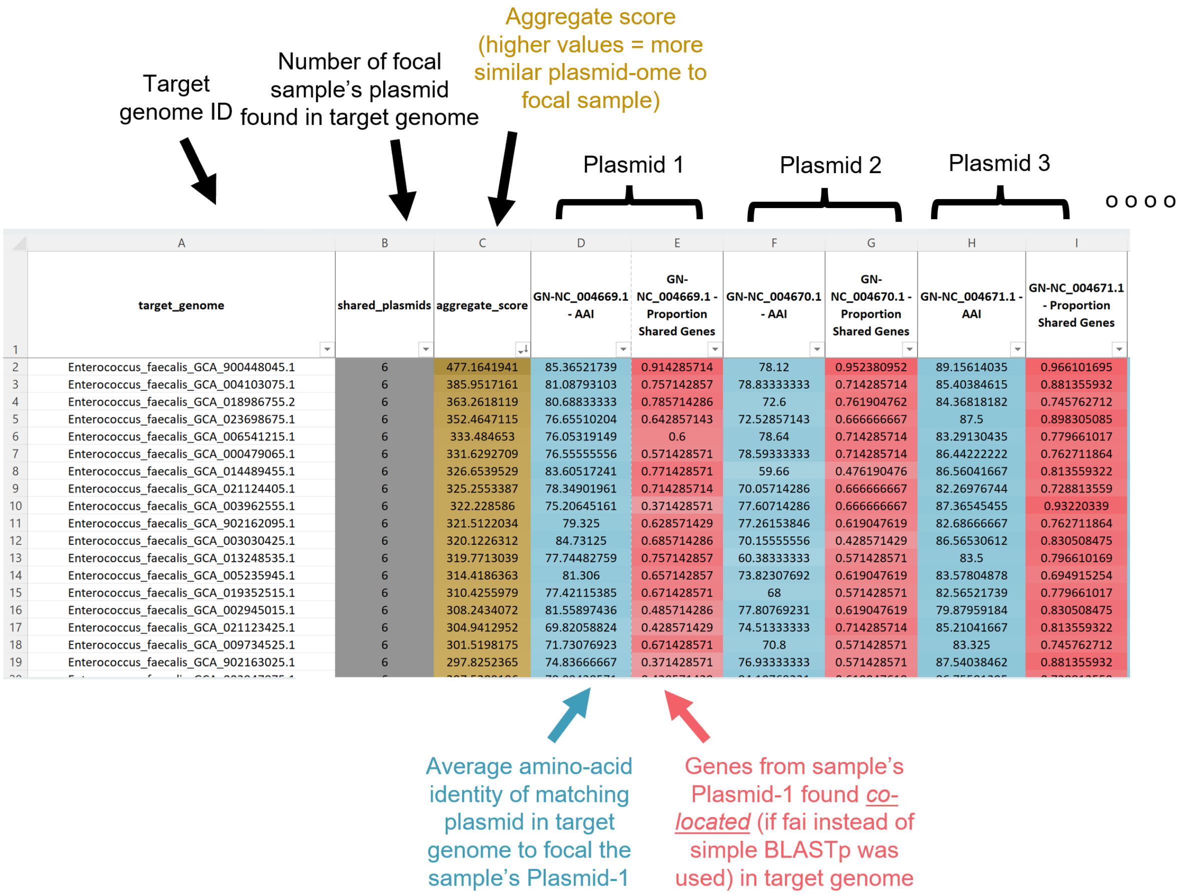
| | U | V | W | X | Y | Z | AA | AB |
|---|---|---|---|---|---|---|---|---|
| t | notati | nder ( | notati | otatio | notati | m Do | Locus | ensus |
| 110 | NA | NA | NA | NA | NA | Thermop | GCA_900 | MTLENFNENL |
| 356 | NA | NA | NA | REFSEQ h | NA | Bacterial | GCA_900 | MARKKTITKE |
| 107 | NA | NA | NA | NA | NA | TspO/MBP | GCA_900 | MNRVKDKRL |
| 593 | NA | NA | NA | sp\|Q65JA | NA | Flavodoxi | GCA_000 | MTLAKIVYASI |
| 356 | NA | NA | BGC0000 | sp\|M1CZO | NA | Metallope | GCA_000 | MITLKSPREIE |
| 356 | NA | NA | BGC0000 | NA | NA | Virulence | GCA_000 | MKFLDNIKKN |
| 356 | NA | NA | BGC0000 | NA | VFG0186 | Glycosyl t | GCA_000 | MGSTMSLFIY |
| 023 | NA | NA | BGC0000 | NA | VFG0490 | Glycosyl t | GCA_000 | MQEMVTISIV |
| 356 | NA | NA | BGC0000 | NA | NA | NA | GCA_000 | MSQRLAVVIV |
| 023 | NA | NA | BGC0000 | sp\|A0A0H | VFG0059 | Glycosyl t | GCA_000 | MISVCIATYN |
| 356 | NA | NA | BGC0000 | sp\|P5525 | VFG0186 | Nucleotid | GCA_000 | MKGIILAGGS |
| 023 | NA | NA | BGC0000 | sp\|O2781 | VFG0076 | dTDP-4-de | GCA_000 | MKVIDTKLQD |
| 356 | NA | NA | BGC0000 | sp\|Q9EQ0 | VFG0060 | GDP-man | GCA_000 | MKKIIVTGGA |
| 023 | NA | NA | BGC0000 | sp\|Q9EQ0 | VFG0060 | RmlD sub | GCA_000 | MILITGGNGQ |
| 356 | NA | NA | BGC0000 | sp\|O2200 | VFG0060 | Glycosyl t | GCA_000 | MKVLLIIPAYN |
| 023 | NA | NA | NA | NA | NA | Uncharac | GCA_000 | MLPTVLWLAL |
| 356 | gb\|AAL27 | NA | NA | sp\|P2321 | NA | Acyltrans | GCA_007 | MTKNNDVLD |
| 665 | NA | html.2020 | NA | NA | NA | Transpos | GCA_007 | MTSLHKNQV |
| 356 | gb\|ABX54 | NA | NA | sp\|P2321 | NA | Acyltrans | GCA_000 | MTKNNDVLD |
| 001 | NA | html.2020 | BGC0000 | sp\|P1977 | NA | Transpos | GCA_000 | MTQVHFTLKS |
| | NA | NA | NA | NA | NA | NA | GCA_000 | MKFAQQLGV |
| 356 | NA | NA | NA | NA | NA | Domain o | GCA_900 | MKKSLSLVFA |
| 023 | NA | NA | NA | NA | VFG0133 | NA | GCA_017 | MPDLKHEDYA |
| 356 | NA | NA | NA | NA | VFG0003 | NA | GCA_017 | MEQIEDFKSA |
| 356 | NA | html.2020 | BGC0000 | sp\|P1977 | NA | Transpos | GCA_000 | MNDFTTEIVK |

Figure 2: zol_results



| 1 | bgc_id | bgc_prediction_software | scaffold_id | start | end | bgc_type | bgc_length | cds_count | key_cds_count |
|---|---|---|---|---|---|---|---|---|---|
| 2 | NC_000964.3.region004 | antismash | NC_000964.3 | 1763762 | 1869009 | transAT-PKS;PKS-like;T3PKS;NRPS;NRPS | 105247 | 46 | 16 |
| 3 | NC_000964.3_cluster_3 | gecco | NC_000964.3 | 1781906 | 1862712 | NRP;Polyketide | 80807 | 19 | 5 |
| 4 | NC_000964.3.region005 | antismash | NC_000964.3 | 1940624 | 2017660 | NRPS;betalactone | 77036 | 37 | 11 |
| 5 | NC_000964.3.region002 | antismash | NC_000964.3 | 358302 | 421744 | NRPS | 63442 | 42 | 7 |
| 6 | NC_000964.3_cluster_4 | gecco | NC_000964.3 | 1946702 | 2003946 | NRP | 57245 | 23 | 3 |
| 7 | NC_000964.3.region009 | antismash | NC_000964.3 | 3260518 | 3312296 | NRP-metallophore;NRPS | 51778 | 45 | 7 |
| 8 | NC_000964.3.region012 | antismash | NC_000964.3 | 3850667 | 3892086 | other | 41419 | 39 | 11 |
| 9 | NC_000964.3.region008 | antismash | NC_000964.3 | 2296955 | 2338053 | T3PKS | 41098 | 43 | 3 |
| 10 | NC_000964.3_cluster_2 | gecco | NC_000964.3 | 370259 | 410669 | NRP | 40411 | 19 | 2 |
| 11 | NC_000964.3_cluster_1 | gecco | NC_000964.3 | 210224 | 232967 | Unknown | 22744 | 26 | 3 |
| 12 | NC_000964.3.region001 | antismash | NC_000964.3 | 204174 | 226248 | ranthipeptide;sactipeptide | 22074 | 22 | 5 |
| 13 | NC_000964.3.region006 | antismash | NC_000964.3 | 2092167 | 2114066 | terpene | 21899 | 19 | 2 |
| 14 | NC_000964.3.region014 | antismash | NC_000964.3 | 4115741 | 4137440 | epipeptide | 21699 | 21 | 3 |
| 15 | NC_000964.3.region011 | antismash | NC_000964.3 | 3826057 | 3847669 | sactipeptide | 21612 | 19 | 4 |
| 16 | NC_000964.3.region010 | antismash | NC_000964.3 | 3593820 | 3614567 | CDPS | 20747 | 17 | 2 |
| 17 | NC_000964.3.region003 | antismash | NC_000964.3 | 1149957 | 1170476 | terpene | 20519 | 21 | 5 |
| 18 | NC_000964.3.region013 | antismash | NC_000964.3 | 4088149 | 4108419 | RRE-containing | 20270 | 18 | 4 |
| 19 | NC_000964.3.region007 | antismash | NC_000964.3 | 2259520 | 2279691 | glycocin | 20171 | 25 | 3 |
| 20 | NC_000964.3_cluster_6 | gecco | NC_000964.3 | 3278325 | 3293359 | NRP | 15035 | 9 | 2 |
| 21 | NC_000964.3_cluster_7 | gecco | NC_000964.3 | 3597289 | 3607121 | Unknown | 9833 | 8 | 2 |
| 22 | NC_000964.3_cluster_5 | gecco | NC_000964.3 | 2265668 | 2271657 | RiPP | 5990 | 6 | 2 |

Figure 3: abon_first_page_results



Target genome ID

Number of focal sample's plasmid found in target genome

Aggregate score (higher values = more similar plasmid-ome to focal sample)

Plasmid 1

Plasmid 2

Plasmid 3

o o o o

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | target_genome | shared_plasmids | aggregate_score | GN-NC_004669.1 - AAI | GN-NC_004669.1 - Proportion Shared Genes | GN-NC_004670.1 - AAI | GN-NC_004670.1 - Proportion Shared Genes | GN-NC_004671.1 - AAI | GN-NC_004671.1 - Proportion Shared Genes |
| 2 | Enterococcus_faecalis_GCA_900448045.1 | 6 | 477.1641941 | 85.36521739 | 0.914285714 | 78.12 | 0.952380952 | 89.15614035 | 0.966101695 |
| 3 | Enterococcus_faecalis_GCA_004103075.1 | 6 | 385.9517161 | 81.08793103 | 0.757142857 | 78.83333333 | 0.714285714 | 85.40384615 | 0.881355932 |
| 4 | Enterococcus_faecalis_GCA_018986755.2 | 6 | 363.2618119 | 80.68833333 | 0.785714286 | 72.6 | 0.761904762 | 84.36818182 | 0.745762712 |
| 5 | Enterococcus_faecalis_GCA_023698675.1 | 6 | 352.4647115 | 76.65510204 | 0.642857143 | 72.52857143 | 0.666666667 | 87.5 | 0.898305085 |
| 6 | Enterococcus_faecalis_GCA_006541215.1 | 6 | 333.484653 | 76.05319149 | 0.6 | 78.64 | 0.714285714 | 83.29130435 | 0.779661017 |
| 7 | Enterococcus_faecalis_GCA_000479065.1 | 6 | 331.6292709 | 76.55555556 | 0.571428571 | 78.59333333 | 0.714285714 | 86.44222222 | 0.762711864 |
| 8 | Enterococcus_faecalis_GCA_014489455.1 | 6 | 326.6539529 | 83.60517241 | 0.771428571 | 59.66 | 0.476190476 | 86.56041667 | 0.813559322 |
| 9 | Enterococcus_faecalis_GCA_021124405.1 | 6 | 325.2553387 | 78.34901961 | 0.714285714 | 70.05714286 | 0.666666667 | 82.26976744 | 0.728813559 |
| 10 | Enterococcus_faecalis_GCA_003962555.1 | 6 | 322.228586 | 75.20645161 | 0.371428571 | 77.60714286 | 0.666666667 | 87.36545455 | 0.93220339 |
| 11 | Enterococcus_faecalis_GCA_902162095.1 | 6 | 321.5122034 | 79.325 | 0.628571429 | 77.26153846 | 0.619047619 | 82.68666667 | 0.762711864 |
| 12 | Enterococcus_faecalis_GCA_003030425.1 | 6 | 320.1226312 | 84.73125 | 0.685714286 | 70.15555556 | 0.428571429 | 86.56530612 | 0.830508475 |
| 13 | Enterococcus_faecalis_GCA_013248535.1 | 6 | 319.7713039 | 77.74482759 | 0.757142857 | 60.38333333 | 0.571428571 | 83.5 | 0.796610169 |
| 14 | Enterococcus_faecalis_GCA_005235945.1 | 6 | 314.4186363 | 81.306 | 0.657142857 | 73.82307692 | 0.619047619 | 83.57804878 | 0.694915254 |
| 15 | Enterococcus_faecalis_GCA_019352515.1 | 6 | 310.4255979 | 77.42115385 | 0.671428571 | 68 | 0.571428571 | 82.56521739 | 0.779661017 |
| 16 | Enterococcus_faecalis_GCA_002945015.1 | 6 | 308.2434072 | 81.55897436 | 0.485714286 | 77.80769231 | 0.619047619 | 79.87959184 | 0.830508475 |
| 17 | Enterococcus_faecalis_GCA_021123425.1 | 6 | 304.9412952 | 69.82058824 | 0.428571429 | 74.51333333 | 0.714285714 | 85.21041667 | 0.813559322 |
| 18 | Enterococcus_faecalis_GCA_009734525.1 | 6 | 301.5198175 | 71.73076923 | 0.671428571 | 70.8 | 0.571428571 | 83.325 | 0.745762712 |
| 19 | Enterococcus_faecalis_GCA_902163025.1 | 6 | 297.8252365 | 74.83666667 | 0.371428571 | 76.93333333 | 0.571428571 | 87.54038462 | 0.881355932 |

Average amino-acid identity of matching plasmid in target genome to focal the sample's Plasmid-1

Genes from sample's Plasmid-1 found *co-located* (if fai instead of simple BLASTp was used) in target genome

Figure 4: abon_second_page_results

# 1. more info on prepTG

prepTG creates a database directory of genomes to search for homologous instances of reference/query gene-clusters in using fai.

The input is simply a directory of either FASTA or GenBank formatted files - with CDS features for the latter - representing bacterial genomes or metagenomes.

For eukaryotic genomes full GenBank format with CDS features are expected; however, FASTA formatted assemblies may instead be provided if a “reference proteome” is provided.

Check out example commands for prepTG on the 4. basic usage examples wiki page.

## Gene-calling bacterial genomes using prodigal and pyrodigal

For bacterial genomes or bacterial metagenomes, users are able to use pyrodigal (default) or prodigal to perform *de novo* gene calling. More recently, we also have the availability of prodigal-gv as an option for gene-calling when phages are the gene clusters of interest.

## Gene-mapping in eukaryotic genomes using miniprot

For eukaryotic genomes, users are able to map a high-quality gene-calling prediction for some reference genome to the remainder of the genomes. This approach is generally recommended only for single-species investigations and has only been tested with microbial eukaryotic genomes of a modest size (e.g. fungii, not gigantic genomes such as those of plants).

## prepTG usage

```
usage: prepTG [-h] [-d DOWNLOAD_PREMADE] [-i INPUT_DIR] [-g GTDB_TAXON] -o OUTPUT_DIR
              [-l LOCUS_TAG_LENGTH] [-r] [-gcm GENE_CALLING_METHOD] [-m]
              [-rp REFERENCE_PROTEOME] [-cst] [-c CPUS] [-mm MAX_MEMORY] [-v]

    Program: prepTG
    Author: Rauf Salamzade
    Affiliation: Kalan Lab, UW Madison, Department of Medical Microbiology and Immunology

    Prepares a directory of target genomes for being searched for query gene clusters using fai.

    Premade databases of representative genomes are available for the following genera:

    Acinetobacter (n=1,643), Bacillales (n=3,150), Corynebacterium (n=726), Enterobacter (n=878),
    Enterococcus (n=937), Escherichia (n=2,436), Klebsiella (n=1,022), Listeria (n=353),
    Mycobacterium (n=744), Pseudomonas (n=2,666), Salmonella (n=308), Staphylococus (n=496),
    Streptomyces (n=1,555), Streptococcus (n=2,452), Cutibacterium (n=27), Neisseria (n=414),
    Lactobacillus (n=541), and Micromonospora (n=211).

    In addition, users can simply request all genomes belonging to a specific species/genus
    in GTDB R214 to be downloaded.

        ----------------------------------------------------------------------------------------
    > Example commands:

    1. Setup a prepTG database which includes some local genomes and all Cutibacterium granulosum
```

```
       genomes:

        prepTG -i User_Genomes_Directory/ -g "Cutibacterium granulosum" -o prepTG_Database/

    2. Setup local prepTG database by downloading a premade one of representative
       Cutibacterium genomes:

        prepTG -d Cutibacterium -o prepTG_Database/

    ----------------------------------------------------------------------------------------
    > Considerations
    If FASTA format is provided, assumption is that genomes are prokaryotic and
    pyrodigal/prodigal will be used to perform gene-calling. Eukaryotic genomes can
    be provided as FASTA format but the --reference_proteome file should be used in
    such case to map proteins from a reference proteome (from the same species ideally)
    on to the target genomes. This will prevent detection of new genes in gene-clusters
    detected by fai but synchronize gene-calling and allow for better similarity
    assessment between orthologous genes.


options:
  -h, --help            show this help message and exit
  -d DOWNLOAD_PREMADE, --download_premade DOWNLOAD_PREMADE
                        Download and setup pre-made databases of representative genomes
                        for specific taxon/genus. Provide name of the taxon,
                        e.g. "Escherichia"
  -i INPUT_DIR, --input_dir INPUT_DIR
                        Directory with target genomes (either featuring GenBanks or FASTAs).
  -g GTDB_TAXON, --gtdb_taxon GTDB_TAXON
                        Name of a GTDB-R214 valid genus or species, should be surrounded by
                        quotes (e.g. "Escherichia coli").
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Output directory, which can then be provided as input for the
                        "-tg" argument in fai.
  -l LOCUS_TAG_LENGTH, --locus_tag_length LOCUS_TAG_LENGTH
                        Length of locus tags to set. Default is 3, allows for <~18k genomes.
  -r, --rename_locus_tags
                        Whether to rename locus tags if provided for CDS features in GenBanks.
  -gcm GENE_CALLING_METHOD, --gene_calling_method GENE_CALLING_METHOD
                        Method to use for gene calling. Options are: pyrodigal, prodigal,
                        or prodigal-gv. [Default is pyrodigal].
  -m, --meta_mode       Flag to use meta mode instead of single for pyrodigal/prodigal.
  -rp REFERENCE_PROTEOME, --reference_proteome REFERENCE_PROTEOME
                        Provide path to a reference proteome to use for protein/gene-calling
                        in target genomes - which should be in FASTA format.
  -cst, --create_species_tree
                        Use skani to infer a neighbor-joing based species tree for the genomes.
  -c CPUS, --cpus CPUS  Total number of cpus/threads to use for running OrthoFinder2/prodigal.
                        [Default is 1].
  -mm MAX_MEMORY, --max_memory MAX_MEMORY
                        Uses resource module to set soft memory limit. Provide in Giga-bytes.
                        Generally memory shouldn't be a major concern unless working
                        with hundreds of large metagenomes. [currently
                        experimental; default is None].
```

```
-v, --version         Get version and exit.
```

# 2. more info on fai

fai allows for identification of homologous instances of a query or set of query gene-cluster(s). It has many options, including two different approaches for delineation of gene-cluster boundaries, requesting filtering of paralogous instances of gene-clusters, and piecing together gene-clusters which are fragmented across scaffolds in genomic assemblies.

The final set of homologous gene clusters from target genomes in GenBank format can be found in the fai results subdirectory: `/path/to/fai_Results/Homologous_GenBanks_Directory/`

## Accounting for paralogous gene-clusters: `-fp / --filter_paralogs`

While paralogy is often thought off with regards to individual genes, full gene-clusters can also be paralogous within individual genomes. We thus allow filtering of paralogous gene-clusters if more than two distinct reference proteins/homolog groups are shared - suggesting paralogy beyond fragmentation that might have split a gene in two.

## Working with poor quality assemblies or MAGs? Consider: `-dm / --draft_mode`

Similar to lsaBGC-Expansion.py, fai allows for detection of gene-clusters fragmented across multiple scaffolds by accounting for proximity to scaffold edges. If `--draft_mode` is requested, thresholds needed for discovery of gene-clusters can simply be met in aggregate, putting together other homologous gene-clusters. Note, each individual gene-cluster segment must still contain three distinct query homolog groups.

## Modes for Gene-Cluster Discovery

Similar to lsaBGC-Expansion and cblaster, fai looks for homologs in target genomes by "BLASTp"ing proteins to predicted proteomes using DIAMOND. Afterwards, the programs differ in how they identify candidate/valid homologous gene-clusters.

cblaster uses a maximum distance parameter (default of 20kb) to determine whether genes should be grouped together in one gene-cluster segment. The cblaster suite also offers an intuitive approach to select the optimal value for this parameter. Similarly, fai offers **"Gene-Clumper"** mode (the current default) which simply groups genes together if they are within 5 genes of each other. fai also offers an **"HMM"** based approach which can be used to identify stringent blocks of gene-sets and then merged together into larger blocks through the same parameter, `--max_genes_disconnect`, used by "Gene-Clumper" mode to group individual genes. Unlike the "HMM" based approach in lsaBGC-Expansion, the emission probability parameters are not automatically determined based on reflexive alignment of the query gene-cluster proteins to the background genome from which it was extracted to determine cutoffs to distinguish paralogous hits from orthologous hits. Similar to lsaBGC-Expansion though, HMM probability parameters are user-adjustable.

## Visuals to Assess Quality of Detection of Homologous Gene-Clusters and Guide User Parameter Adjustment

fai will by default produce a "Tiny-AAI" plot which depicts the average-amino acid identity of genes from homologous gene clusters in target genomes to the query gene cluster (x-axis) and the proportion of query genes/ortholog groups found.

fai can also produce a multi-page PDF with plots such as the following for showing the quality of homologous gene-clusters detected. This report can be requested via the `-gp` or `--generate_plots` argument.

These plots showcase syntenic order and similarity to reference genes (height is the CDS to reference protein ratio - ideal match should be at 1, indicating they are the same length) and colored on a scale from 0 (grey)

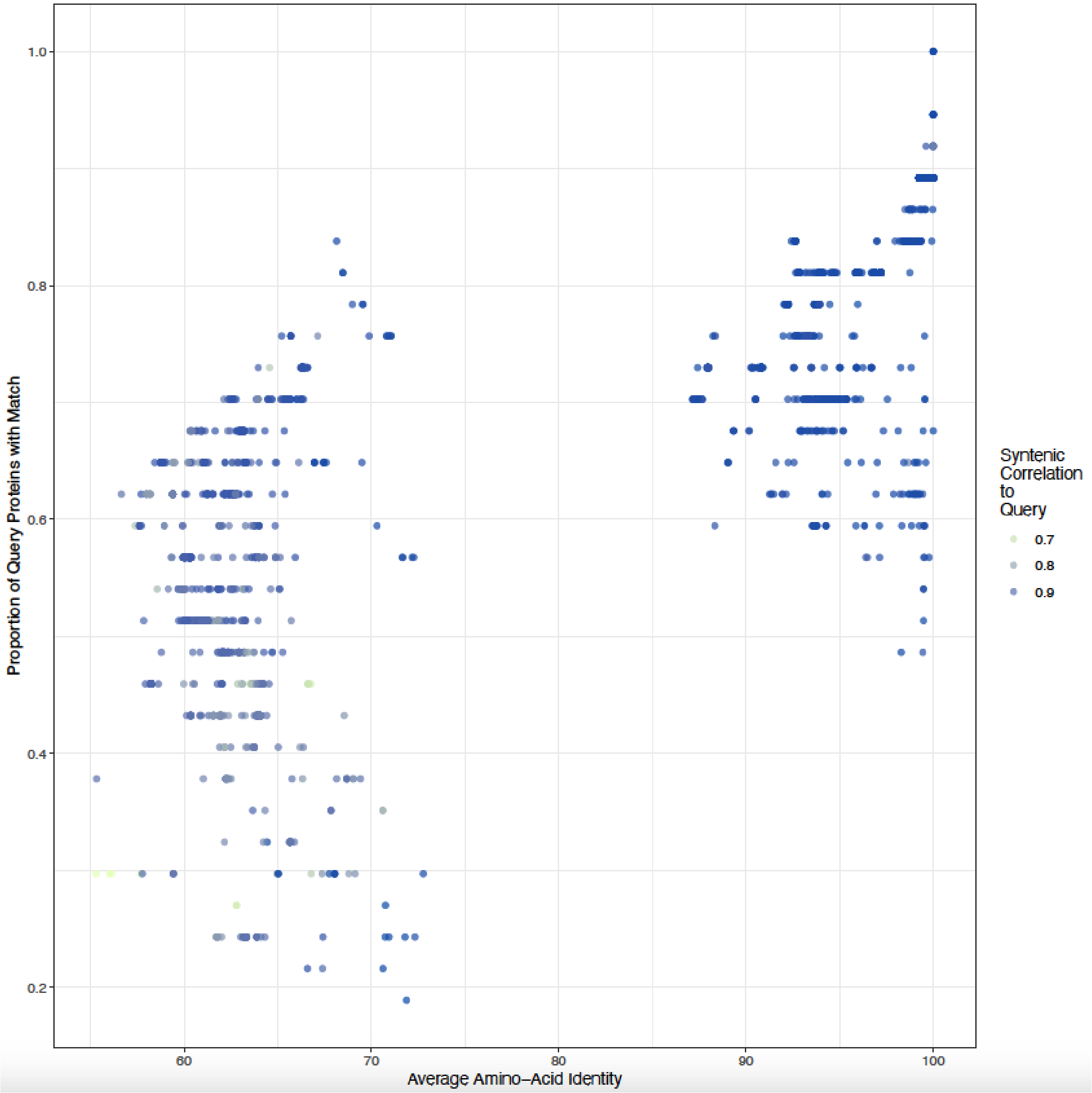


Figure 1: image

to 100 (red) corresponding to percent identity. Black borders indicate key query proteins - if provided by the user.

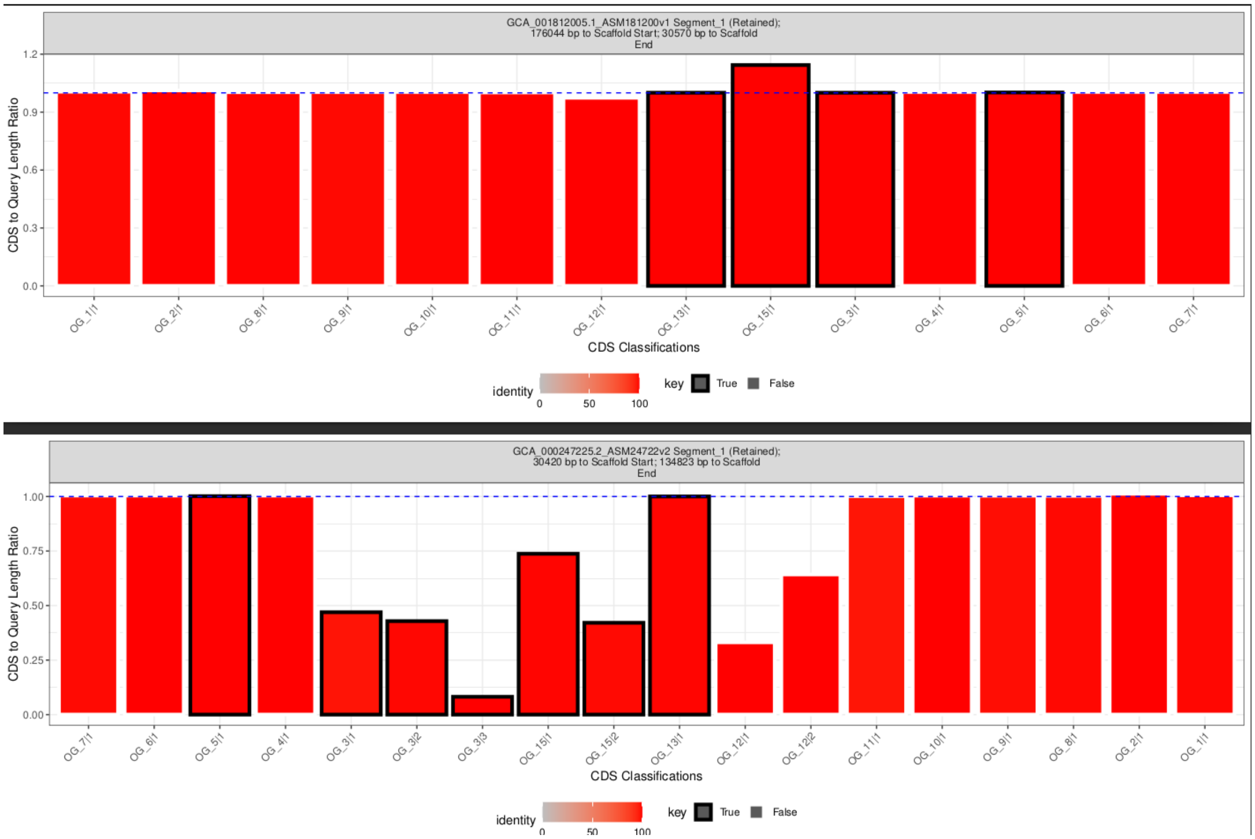


Figure 2: image

## Explanation of Report

| Column | Description | Notes |
|---|---|---|
| **sample** | The identifier of the target genome. | |
| **gene-cluster-id** | The identifier of a discrete neighborhood of genes identified as homologous to the query gene cluster. | Only in the “Gene Cluster Instance - Report” tab |
| **aggregate-bitscore** | The aggregate bitscore of hits to the query gene cluster genes. | Only the best hit for each query gene/ortholog-group is retained (based on bitscore). |
| **aai-to-query** | The average amino-acid identity of the proteins in the target genome to the query gene cluster genes. | Only the best hit for each query gene/ortholog-group is retained (based on bitscore). |
| **mean-sequence-to-query-ratio** | The average sequence-to-query ratio of the proteins. | Only the best hit for each query gene/ortholog-group is retained (based on bitscore). |

| Column | Description | Notes |
|---|---|---|
| **proportion-query-genes-found** | The proportion of query genes/ortholog-groups found in homologous gene clusters across the target genome ("Genome Wide - Report") or in a specific discrete neighborhood ("Gene Cluster Instance - Report") | |
| **avg-syntenic-correlation** | Pearson product-moment correlation coefficient for global syntenic similarity of a specific discrete neighborhood to the query gene cluster or the average of these values across all discrete neighborhoods which meet user defined filters. | |
| **number-background-genes** | The number of background genes in the delineated region within query gene hits which are not represented in the query. | |
| **number-gene-clusters** | The number of discrete gene neighborhoods which individually meet reporting criteria. | Only in the "Genome Wide - Report" tab |
| **copy-counts** | A string separated by commas listing the copy-count of individual query genes/ortholog-groups. | |

## Usage

```
usage: fai [-h] [-i QUERY_INPUTS [QUERY_INPUTS ...]] [-r REFERENCE_GENOME]
           [-rc REFERENCE_CONTIG] [-rs REFERENCE_START] [-re REFERENCE_END]
           [-pq PROTEIN_QUERIES] [-sq SINGLE_QUERY] -tg TARGET_GENOMES_DB
           -o OUTPUT_DIR [-st SPECIES_TREE] [-dm] [-fp] [-e EVALUE_CUTOFF]
           [-m MIN_PROP] [-kpq KEY_PROTEIN_QUERIES]
           [-kpe KEY_PROTEIN_EVALUE_CUTOFF] [-kpm KEY_PROTEIN_MIN_PROP]
           [-sct SYNTENIC_CORRELATION_THRESHOLD] [-gdm GC_DELINEATION_MODE]
           [-f FLANKING_CONTEXT] [-mgd MAX_GENES_DISCONNECT] [-gt GC_TRANSITION]
           [-bt BG_TRANSITION] [-ge GC_EMISSION] [-be BG_EMISSION] [-gp]
           [-ds DIAMOND_SENSITIVITY] [-phl PHYLOHEATMAP_LENGTH]
           [-phw PHYLOHEATMAP_WIDTH] [-c CPUS] [-cl] [-mm MAX_MEMORY] [-v]

    Program: fai
    Author: Rauf Salamzade
    Affiliation: Kalan Lab, UW Madison, Department of Medical Microbiology and Immunology
```

```
         .o88o.            o8o
         888 ‘"            ‘"’
        o888oo   .oooo.   oooo
         888    ‘P  )88b  ‘888
         888     .oP"888   888
         888    d8(  888   888
        o888o   ‘Y888""8o o888o

    MODES OF INPUT:
    **********************************************************************************

    Type 1: Directory of Homologous Gene-Cluster GenBanks
    (GenBanks must have CDS features with locus_tag or protein_id names)
    -----------------------------------------------------
    $ fai -i Known_GeneCluster.gbk -tg prepTG_Database/ -o fai_Results/

    Type 2: Reference Genome with Gene-Cluster/Locus Coordinates
    (proteins are collapsed for high-similarity using cd-hit)
    -----------------------------------------------------
    $ fai -r Reference.fasta -rc scaffold01 -rs 40201 -re 45043 \
          -tg prepTG_Database/ -o fai_Results/

    Type 3: Set of Query Proteins (not compatible with the syntenic similarity cutoff)
    (proteins are collapsed for high-similarity using cd-hit)
    Similar to input for cblaster
    -----------------------------------------------------
    $ fai -pq Gene-Cluster_Query_Proteins.faa -tg prepTG_Database/ -o fai_Results/

    Type 4: Single Query (provide the amino acid sequence directly)
    Similar to CORASON
    ----------------------------------------------------
    $ fai -sq Single_Query_Protein.fa -tg prepTG_Database/ -o fai_Results/


    The final set of homologous gene cluster instances within target genomes which meet
    the specified criteria can be found in the subdirectory named:
    Final_Results/Homologous_Gene_Cluster_GenBanks/


options:
  -h, --help            show this help message and exit
  -i QUERY_INPUTS [QUERY_INPUTS ...], --query_inputs QUERY_INPUTS [QUERY_INPUTS ...]
                        Paths to locus-specific GenBank(s) [could be multiple] to use as
                        queries for searching for homologous/orthologous instances in target
                        genomes. Files must end with ".gbk", ".gbff", or ".genbank".
  -r REFERENCE_GENOME, --reference_genome REFERENCE_GENOME
                        Path to reference genome in FASTA or GenBank format.
  -rc REFERENCE_CONTIG, --reference_contig REFERENCE_CONTIG
                        Scaffold name (up to first space) which features region
                        of interest.
  -rs REFERENCE_START, --reference_start REFERENCE_START
```

```
                        Start position of gene-cluster on scaffold.
  -re REFERENCE_END, --reference_end REFERENCE_END
                        End position of gene-cluster on scaffold.
  -pq PROTEIN_QUERIES, --protein_queries PROTEIN_QUERIES
                        Path to protein multi-FASTA file containing to use as queries.
  -sq SINGLE_QUERY, --single_query SINGLE_QUERY
                        Path to protein FASTA file containing a single protein to use as a query.
  -tg TARGET_GENOMES_DB, --target_genomes_db TARGET_GENOMES_DB
                        Result directory from running prepTG for target genomes of interest.
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Parent output/workspace directory.
  -st SPECIES_TREE, --species_tree SPECIES_TREE
                        Phylogeny in Newick format - with names matching target
                        genomes db [Optional]. Will be used for creating an extra visual.
  -dm, --draft_mode     Run in draft-mode to also report segments on scaffold edges which in
                        aggregate (with other such segments) they meet criteria
                        for reporting.
  -fp, --filter_paralogs
                        Filter paralogous instances of gene-cluster identified in a
                        single target genome.
  -e EVALUE_CUTOFF, --evalue_cutoff EVALUE_CUTOFF
                        E-value cutoff for DIAMOND blastp to consider a gene in a
                        target genome a hit to a query protein. [Default
                        is 1e-10].
  -m MIN_PROP, --min_prop MIN_PROP
                        The minimum proportion of distinct proteins/ortholog groups
                        needed to report discrete segments of the gene-cluster.
                        Note, that a minimum of 3 distinct query proteins/homolog
                        groups are needed per segment reported.
  -kpq KEY_PROTEIN_QUERIES, --key_protein_queries KEY_PROTEIN_QUERIES
                        Path to protein multi-FASTA file containing key query sequences
                        which some proportin of are required to be present in a gene cluster
                        at a specific E-value cutoff.
  -kpe KEY_PROTEIN_EVALUE_CUTOFF, --key_protein_evalue_cutoff KEY_PROTEIN_EVALUE_CUTOFF
                        E-value cutoff for finding key query sequences in putative
                        gene cluster homolog segments. [Default
                        is 1e-20]. Disregarded if less strict than the
                        general --evalue cutoff.
  -kpm KEY_PROTEIN_MIN_PROP, --key_protein_min_prop KEY_PROTEIN_MIN_PROP
                        The minimum proportion of distinct ortholog groups matching
                        key proteins needed to report a homologous gene-cluster. [Default is 0.0].
  -sct SYNTENIC_CORRELATION_THRESHOLD, --syntenic_correlation_threshold SYNTENIC_CORRELATION_THRESHOLD
                        The minimum syntenic correlation needed to at least one known
                        GCF instance to report segment. [Default is 0.6]
  -gdm GC_DELINEATION_MODE, --gc_delineation_mode GC_DELINEATION_MODE
                        Method/mode for delineation of gene-cluster boundaries. Options are
                        "Gene-Clumper" or "HMM". Default is Gene-Clumper.
  -f FLANKING_CONTEXT, --flanking_context FLANKING_CONTEXT
                        Number of bases to append to the beginning/end of the gene-cluster
                        segments identified. [Default is 1000].
  -mgd MAX_GENES_DISCONNECT, --max_genes_disconnect MAX_GENES_DISCONNECT
                        Maximum number of genes between gene-cluster segments detected by HMM to
                        merge segments together. Alternatively the number of genes separating
                        hits if Gene-Clumper mode is used. Allows for more inclusivity of novel
```

```
                      auxiliary genes. [Default is 5].
-gt GC_TRANSITION, --gc_transition GC_TRANSITION
                      Probability for gene-cluster to gene-cluster transition in HMM.
                      Should be between 0.0 and 1.0. [Default is 0.9].
-bt BG_TRANSITION, --bg_transition BG_TRANSITION
                      Probability for background to background transition in HMM.
                      Should be between 0.0 and 1.0. [Default is 0.9].
-ge GC_EMISSION, --gc_emission GC_EMISSION
                      Emission probability of gene being in gene-cluster state assuming a
                      orthologis found at the e-value cutoff. [Default is 0.95].
-be BG_EMISSION, --bg_emission BG_EMISSION
                      Emission probability of gene being in gene-cluster state assuming no
                      homolog is found at the e-value cutoff. [Default is 0.2].
-gp, --generate_plots
                      Generate plots for assessing gene-cluster segments identified.
-ds DIAMOND_SENSITIVITY, --diamond_sensitivity DIAMOND_SENSITIVITY
                      DIAMOND alignment sensitivity. Options include: fast, mid-sensitive,
                      sensitive, more-sensitive, very-sensitive, and ultra-sensitive.
                      [Default is very-sensitive].
-phl PHYLOHEATMAP_LENGTH, --phyloheatmap_length PHYLOHEATMAP_LENGTH
                      Specify the height/length of the phylo-heatmap plot. Default is 7.
-phw PHYLOHEATMAP_WIDTH, --phyloheatmap_width PHYLOHEATMAP_WIDTH
                      Specify the width of the phylo-heatmap plot. Default is 10.
-c CPUS, --cpus CPUS  The number of cpus to use. [Default is 1].
-cl, --clean_up       Clean up disk-heavy files/folders.
-mm MAX_MEMORY, --max_memory MAX_MEMORY
                      Uses resource module to set soft memory limit. Provide
                      in Giga-bytes. Generally memory shouldn't be a major concern unless
                      working with hundreds of large metagenomes. [currently
                      experimental; default is None].
-v, --version         Get version and exit.
```

# 3. more info on zol

## Explanation of Report

| Column | Description | Notes |
|---|---|---|
| **Ortholog Group (OG) ID** | The identifier of the ortholog/homolog group. | |
| **OG is Single Copy?** | Whether the ortholog/homolog group is single copy in the context of the gene-cluster. ***Evolutionary statistics should be evaluated carefully if False or multiple gene-clusters are from the same genome.*** | |
| **Proportion of Total Gene Clusters with OG** | The proportion of input gene-clusters/GenBanks which feature the homolog group. | |
| **OG Median Length (bp)** | The median length of the homolog group in basepairs. | |
| **OG Consensus Order** | The consensus order of the homolog group across all gene clusters. | |
| **OG Consensus Direction** | The consensus direction of the homolog group across all gene clusters. | |
| **Proportion of Focal Gene Clusters with OG** | | Only produced if comparative analysis is requested by user. |
| **Proportion of Comparator Gene Clusters with OG** | | Only produced if comparative analysis is requested by user. |
| **Fixation Index** | Fst estimate based on measuring pairwise differences in codon alignments and the statistic developed by Hudson, Slatkin, and Maddison 1992 | Only produced if comparative analysis is requested by user. |
| **Upstream Region Fixation Index** | Fst estimate based on upstream 100 bp nucleotide alignments. | |
| **Tajima's D** | Tajima's D calculated using implementation described in lsaBGC based on statistic developed by Tajima 1989. | Interpret with care and consideration of divergence of genomes from which gene clusters were extracted. Calculation of statistic modified to account for the presence of gaps in alignments. Filtering of codon alignments in zol is currently different than what is applied in lsaBGC. |
| **Proportion of Filtered Codon Alignment is Segregating Sites** | Proportion of sites in filtered codon alignment which correspond to segregating sites. | Note, segregating sites require two different non-gap alleles - gaps are not counted as a distinct allele. |

| Column | Description | Notes |
|---|---|---|
| **Entropy** | Average entropy over largely non-ambiguous sites (<10% ambiguity) in codon alignments. | |
| **Upstream Region Entropy** | Average entropy over largely non-ambiguous sites (<10% ambiguity) in nucleotide alignments of upstream regions. | |
| **Median Beta-RD-gc** | The median Beta-RD statistic for ortholog group relative to the full gene-cluster. | Calculation is similar to what was described in the lsaBGC study, but expected divergence for ortholog group sequence between pair of gene-clusters/samples is not based on genome-wide divergence but gene-cluster divergence. |
| **Max Beta-RD-gc** | The max Beta-RD statistic observed for the ortholog group between two gene-clusters. | |
| **Proportion of sites which are highly ambiguous in codon alignment** | The proportion of sites which are ambiguous (e.g. feature gaps) in greater than 10% of the sequences of a codon alignments (before trimming/filtering). | |
| **Proportion of sites which are highly ambiguous in trimmed codon alignment** | The proportion of sites which are ambiguous (e.g. feature gaps) in greater than 10% of the sequences of trimmed codon alignments (via trimal). | |
| **Median GC** | The median GC% of genes belonging to the ortholog group. | |
| **Median GC Skew** | The median GC skew (G-C)/(G+C) of genes belonging to the ortholog group. | |
| **GARD Partitions Based on Distinct Segments based on Recombination Breakpoints** | Number of recombination segments detected by HyPhy's GARD method: Kosakovsky Pond et al. 2006. Not run by default due to time requirements. | |
| **Number of Sites Identified as Under Positive or Negative Selection** | The number of sites inferred as under positive Prob[N1<N2] or negative selection Prob[N1>N2] based on FUBAR method: Not run by default due to time requirements. Uses HyPhy's FUBAR method: Murrell et al. 2013 | |
| **Average delta(Beta, Alpha) by FUBAR across sites** | The average difference of N2-N1 across sites in the codon alignment as calculated by FUBAR. | More negative values imply greater purifying selection whereas more positive values imply greater positive selection. |

| Column | Description | Notes |
|---|---|---|
| **Proportion of Sites Under Selection which are Positive** | Proportion of the number of sites identified as under either positive or negative selection by FUBAR analysis which are under positive selection. | |
| **Custom Annotation (E-value)** | Custom annotation based on user providing custom protein database. | |
| **KO Annotation (E-value)** | Best KEGG ortholog annotation(s) (the HMMER3 E-value associated with the best score) | |
| **PGAP Annotation (E-value)** | Best PGAP annotation(s) (the HMMER3 E-value associated with the best score) | |
| **PaperBLAST Annotation (E-value)** | Best PaperBLAST annotation(s) (the DIAMOND E-value associated with the best bitscore). For associated papers BLAST the consensus sequence or the ID here to on the PaperBLAST webpage. | |
| **CARD Annotation (E-value)** | Best CARD annotation(s) of antimicrobial resistance genes (the DIAMOND E-value associated with the best bitscore) | |
| **IS Finder (E-value)** | Best ISFinder annotation(s) of IS elements / transposons (the DIAMOND E-value associated with the best bitscore) | |
| **MIBiG Annotation (E-value)** | Best MIBiG annotation(s) for genes in characterized BGCs (the DIAMOND E-value associated with the best bitscore) | |
| **VOG Annotation (E-value)** | Best VOG annotation(s) for viral/phage ortholog groups (the HMMER3 E-value associated with the best score) | |
| **Pfam Domains** | Pfam domains with E-value < 1e-5 and meeting the "trusted" score thresholds. | |
| **CDS Locus Tags** | Locus tag identifiers of genes belonging to the ortholog group. | |
| **Consensus Sequence** | The consensus sequence for the ortholog group. | |

## Method of Annotation

Some of the 8 annotation databases are profile HMMs whereas others are DIAMOND databases:

profile HMMs: KO, PGAP, VOG, Pfam DIAMOND databases: PaperBLAST, CARD, IS Finder, and MIBiG

The consensus sequence of each ortholog group is used for annotations for computational efficiency and consolidation. For both profile HMMs (searched using hmmscan in HMMER3) and DIAMOND databases (searched via DIAMOND blastp) we require an E-value of 1e-5 to report the best scoring annotation(s) (based on score or bitscore).

## Determination of Ortholog Group Consensus Order and Direction

For details on how the ortholog group consensus order and direction are calculated, please reference the description on the lsaBGC wiki. We use a similar approach in zol.

## Usage

```
usage: zol [-h] [-i INPUT_DIR] -o OUTPUT_DIR [-sfp] [-it IDENTITY_THRESHOLD]
           [-ct COVERAGE_THRESHOLD] [-et EVALUE_THRESHOLD] [-fl] [-fd] [-r] [-d]
           [-ri] [-dt DEREP_IDENTITY] [-dc DEREP_COVERAGE] [-di DEREP_INFLATION]
           [-ibc] [-ces] [-aec] [-q] [-s] [-sg] [-cd CUSTOM_DATABASE] [-rgc]
           [-l LENGTH] [-w WIDTH] [-fgl] [-f FOCAL_GENBANKS]
           [-fc COMPARATOR_GENBANKS] [-oo] [-c CPUS] [-mm MAX_MEMORY] [-v]

        Program: zol
        Author: Rauf Salamzade
        Affiliation: Kalan Lab, UW Madison, Department of Medical Microbiology and Immunology

        ***************************************************************************************

                       oooooooooooo           ooooo
                      d’’’’’’’d888’           ‘888’
                            .888P    .ooooo.   888
                           d888’    d88’ ‘88b  888
                         .888P      888   888  888
                        d888’    .P 888   888  888       o
                      .8888888888P  ‘Y8bod8P’ o888ooooood8

        ***************************************************************************************

        zol is a lightweight software that can generate reports on conservation, annotation,
        and evolutionary statistics for defined orthologous/homologous loci (e.g. BGCs, phages,
        MGEs, or any genomic island / operon!).

        CONSIDERATIONS:
        ---------------
        * It is advised that multiple GenBanks from the same genome/sample be concatenated into
          a multi-record GenBank to account for fragmentation of gene-clusters and properly
          calculate copy count of ortholog groups.
        * Locus tags cannot contain commas, if they do however, you can use the --rename_lt flag
          to request new locus tags!
        * Ortholog group and homolog group are/were used inter-changeably in the code/comments. We
          recommend using the term ortholog group which is more commonly used in literature for
          the type of protein clustering we perform in zol. Since v1.28 - result files and logging
          messages should largely use "ortholog group" or "OG".
        * Dereplication uses ANI & AF estimates by skani, which the author recommends should be
```

```
          used on contigs (or gene-clusters in this case) greater than 10 kb for accurate
          calculations.


options:
  -h, --help            show this help message and exit
  -i INPUT_DIR, --input_dir INPUT_DIR
                        Directory with orthologous/homologous locus-specific GenBanks.
                        Files must end with ".gbk", ".gbff", or ".genbank".
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Parent output/workspace directory.
  -sfp, --select_fai_params_mode
                        Determine statistics informative for selecting parameters for running
                        fai to find more instances of the gene cluster.
  -it IDENTITY_THRESHOLD, --identity_threshold IDENTITY_THRESHOLD
                        Minimum identity coverage for an alignment between protein
                        pairs from two gene-clusters to consider in search for
                        orthologs. [Default is 30].
  -ct COVERAGE_THRESHOLD, --coverage_threshold COVERAGE_THRESHOLD
                        Minimum query coverage for an alignment between protein
                        pairs from two gene-clusters to consider in search for
                        orthologs. [Default is 50].
  -et EVALUE_THRESHOLD, --evalue_threshold EVALUE_THRESHOLD
                        Maximum E-value for an alignment between protein pairs from
                        two gene-clusters to consider in search for orthologs.
                        [Default is 0.001].
  -fl, --filter_low_quality
                        Filter gene-clusters which feature alot of missing
                        bases (>10 percent).
  -fd, --filter_draft_quality
                        Filter records of gene-clusters which feature CDS
                        features on the edge of contigs (those marked with
                        attribute near_contig_edge=True by fai) or which are
                        multi-record.
  -r, --rename_lt       Rename locus-tags for CDS features in GenBanks.
  -d, --dereplicate     Perform dereplication of input GenBanks using skani
                        and single-linkage clustering or MCL.
  -ri, --reinflate      Perform re-inflation with all gene-clusters of
                        ortho-groups identified via dereplicated analysis.
  -dt DEREP_IDENTITY, --derep_identity DEREP_IDENTITY
                        skani ANI threshold to use for dereplication. [Default is 99.0].
  -dc DEREP_COVERAGE, --derep_coverage DEREP_COVERAGE
                        skani aligned fraction threshold to use for
                        dereplication. [Default is 95.0].
  -di DEREP_INFLATION, --derep_inflation DEREP_INFLATION
                        Inflation parameter for MCL to use for dereplication of
                        gene-clusters. If not specified single-linkage clustering
                        will be used instead.
  -ibc, --impute_broad_conservation
                        Impute weighted conservation stats based on cluster size associated
                        with dereplicated representatives.
  -ces, --comprehensive_evo_stats
                        Allow computing of evolutionary statistics for non-single-copy genes.
  -aec, --allow_edge_cds
```

```
                      Allow CDS within gene-cluster GenBanks with the attribute
                      "near_scaffold_edge=True", which is set by fai for features
                      within 2kb of contig edges.
-q, --use_super5      Use MUSCLE super5 for alignments - faster
                      but less accurate.
-s, --selection_analysis
                      Run selection analysis using HyPhy's GARD
                      and FUBAR methods. These are turned off by default because
                      they are computationally intensive.
-sg, --skip_gard      Skip GARD detection of recombination breakpoints
                      prior to running FUBAR selection analysis. Less
                      accurate than running with GARD preliminary analysis,
                      but much faster. Default is False because these are
                      computationally intensive.
-cd CUSTOM_DATABASE, --custom_database CUSTOM_DATABASE
                      Path to FASTA file of protein sequences corresponding to a
                      custom annotation database.
-rgc, --refine_gene_calling
                      Perform gene-calling refinement using custom database.
                      All ortholog groups which don't match to a protein in the
                      custom database will be ignored.
-l LENGTH, --length LENGTH
                      Specify the height/length of the heatmap plot. Default is 7.
-w WIDTH, --width WIDTH
                      Specify the width of the heatmap plot. Default is 10.
-fgl, --full_genbank_labels
                      Use full GenBank labels instead of just the first 20 characters.
-f FOCAL_GENBANKS, --focal_genbanks FOCAL_GENBANKS
                      File with focal genbank(s) listed (one per line).
-fc COMPARATOR_GENBANKS, --comparator_genbanks COMPARATOR_GENBANKS
                      Optional file with comparator genbank(s) listed.
                      Default is to use remaining GenBanks as comparators to focal listing.
-oo, --only_orthogroups
                      Only compute ortholog groups and stop (runs up to step 2).
-c CPUS, --cpus CPUS  Number of cpus/threads to use.
-mm MAX_MEMORY, --max_memory MAX_MEMORY
                      Uses resource module to set soft memory limit. Provide in Giga-bytes.
                      Generally memory shouldn't be a major concern unless working
                      with hundreds of large metagenomes. [currently
                      experimental; default is None].
-v, --version         Get version and exit.
```

# 4. basic usage examples

## prepTG (preparing target genomes database)

prepTG formats and parses information in provided GenBank files or can run prodigal (for bacteria only!) for gene-calling if provided FASTA files and subsequently create GenBank files.

Create a target genomes database from user provided genomes (in FASTA or GenBank format) provided in a folder.

```
prepTG -i Folder_with_Genomes_to_Search/ -o prepTG_DB/
```

Create a target genomes database from user provided genomes (in FASTA or GenBank format) provided in a folder *and* include all genomes assigned as a certain bacterial genus or species in GTDB R214 (e.g. *Cutibacterium acnes*):

```
prepTG -i Folder_with_User_Provided_Genomes/ -g "Cutibacterium acnes" -o prepTG_DB/
```

:warning:***BE CAREFUL, WELL-SEQUENCED TAXA CAN RESULT IN LARGE PREPTG DATABASES AND LARGE FILES IN THE FAI RESULTS!!!***

Download a pre-made target genomes database based on *distinct representative genomes* for a variety of taxa:

```
prepTG -d Cutibacterium -o prepTG_DB/
```

For additional details on prepTG (e.g. how to download genomes from NCBI), please check out the 1. more info on prepTG wiki page.

## fai (finding homologous instances of query gene clusters)

### 1. Provide GenBank(s) of known instance(s) of gene cluster

```
fai -i Known_GeneCluster.gbk -tg prepTG_Database/ -o fai_Results/
```

Here the `Known_GeneCluster_GenBank.gbk` represents a GenBank corresponding to a reference of a single gene-cluster of interest. Multiple reference gene cluster GenBanks can be provided. If multiple GenBanks are provided, homolog groups are identified between them to simplify the DIAMOND search operation.

MIBiG users, rejoice, you can download a GenBank for any entry using the "Download Cluster GenBank file" link. This input format is made in mind for most users of BGC prediction software, such as antiSMASH or GECCO.

### 2. Provide gene-cluster coordinates along a FASTA reference genome

```
fai -r Reference.fasta -rc scaffold01 -rs 40201 -re 45043 -tg prepTG_Database/ -o fai_Results/
```

Provide the coordinates of a gene-cluster along a reference genome. This option is likely the most compatible with sources of gene-clusters from various websites such as ICEberg, IslandViewer, and PHASTER.

### 3. Provide proteins gene-cluster using set of proteins that should be co-clustered (similar to cblaster!)

```
fai -pq Gene-Cluster_Query_Proteins.faa -tg prepTG_Database/ -o fai_Results/
```

In this format a FASTA file with protein sequences belonging to the gene-cluster is used for searching in target genomes. This is the same format as what cblaster uses. Note, this input format does not allow for assessment of syntenic similarity between the query gene-cluster(s) and homologous instances identified in target genomes.

### 4. Provide a single query protein and use to extract surrounding +/-20kb of homologs in target genomes (inspired by CORASON; implementation still experimental)

Note, this option is still experimental. The concept of looking at variability in the context of a focal gene stems from CORASON but we don't use RBH and only an adjustable E-value threshold to identify homologs in target genomes. Unlike, the other 3 ways to run fai to identify gene clusters - where syntenic support can be used to better infer orthology - here we are more limited and can only infer homology. We might pair the -sq argument with another to provide a reference genome for the single query protein eventually.

```
fai -sq Single_Query_Protein.faa -tg prepTG_Database/ -o fai_Results/ -f 20000
```

For additional details on fai (e.g. how it relates to cblaster and lsaBGC-Expansion, plots it can create to assess homologous gene-clusters detected), please check out the 2. more info on fai wiki page.

## zol (summarize information across homologous instances of a gene cluster)

```
zol -i Genbanks_Directory/ -o zol_Results/
```

if running after fai, then the input directory would be the `Homologous_GenBanks_Directory/` subdirectory. So the typical run through the workflow would likely involve a command similar to the following:

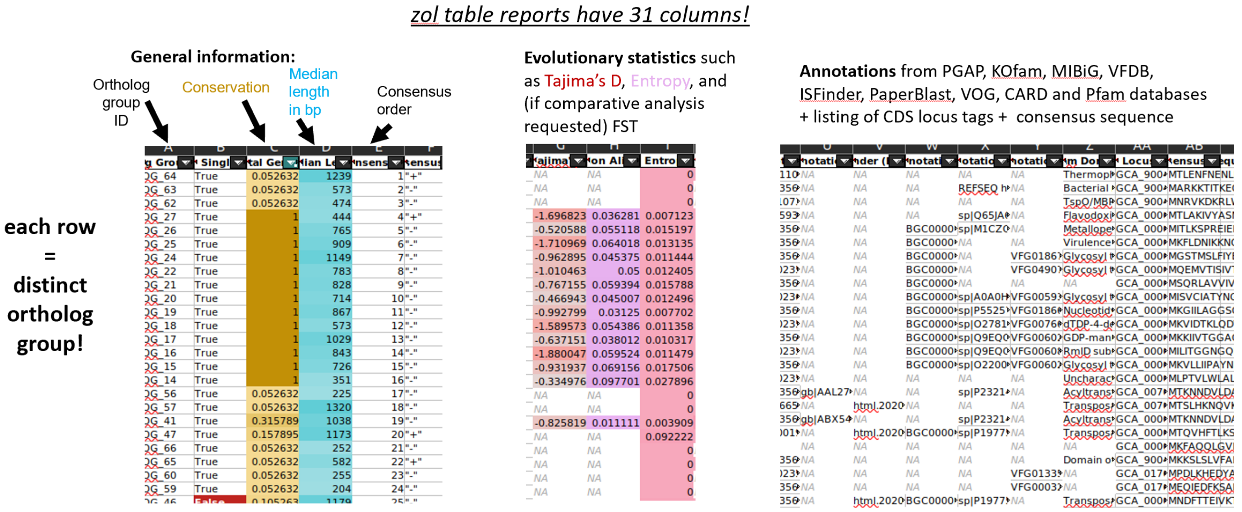
```
zol -i fai_Results/Final_Results/Homologous_Gene_Cluster_GenBanks/ -o zol_Results/
```

> By default, zol will scale to around 100 to 300 distinct gene clusters, if you have more and you suspect there is some redundancy, you can use dereplication via the `-d` option to collapse very similar gene-cluster instances down and use only representative gene clusters to determine ortholog groups before expanding back out to compute evolutionary stats!

zol produces an XLSX spreadsheet report (within the sub-directory `Final_Results/`) where rows correspond to each individual ortholog group/homolog-group and columns provide basic stats, consensus order, annotation information using multiple databases, and evolutionary/selection-inference statistics. Coloring is automatically applied on select quantitative field for users to more easily assess trends. ***I strongly recommend providing a custom-annotation database as a FASTA file of protein sequences with headers corresponding to unique identifiers via the `-cd` argument because this will allow you to more easily link the ortholog groups to known genes from a well studied instance of the gene cluster if that exists!***

Annotation databases include: KEGG, NCBI's PGAP, PaperBLAST, VOGs (phage related genes), MIBiG (genes from characterized BGCs), VFDB (virulence factors), CARD (antibiotic resistance), ISfinder (transposons/insertion-sequences).

For details on the stats/annotations zol infers, please refer to the zol wiki page.


zol table reports have 31 columns!
General information:
Ortholog group ID
Conservation
Median length in bp
Consensus order
Evolutionary statistics such as Tajima's D, Entropy, and (if comparative analysis requested) FST
Annotations from PGAP, KOfam, MIBiG, VFDB, ISFinder, PaperBlast, VOG, CARD and Pfam databases + listing of CDS locus tags + consensus sequence
each row = distinct ortholog group!

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| g Gro | Singl | tal Ge | lian Le | nsens | ensus |
| OG_64 | True | 0.052632 | 1239 | 1 | "+" |
| OG_63 | True | 0.052632 | 573 | 2 | "-" |
| OG_62 | True | 0.052632 | 474 | 3 | "-" |
| OG_27 | True | 1 | 444 | 4 | "+" |
| OG_26 | True | 1 | 765 | 5 | "-" |
| OG_25 | True | 1 | 909 | 6 | "-" |
| OG_24 | True | 1 | 1149 | 7 | "-" |
| OG_22 | True | 1 | 783 | 8 | "-" |
| OG_21 | True | 1 | 828 | 9 | "-" |
| OG_20 | True | 1 | 714 | 10 | "-" |
| OG_19 | True | 1 | 867 | 11 | "-" |
| OG_18 | True | 1 | 573 | 12 | "-" |
| OG_17 | True | 1 | 1029 | 13 | "-" |
| OG_16 | True | 1 | 843 | 14 | "-" |
| OG_15 | True | 1 | 726 | 15 | "-" |
| OG_14 | True | 1 | 351 | 16 | "-" |
| OG_56 | True | 0.052632 | 225 | 17 | "-" |
| OG_57 | True | 0.052632 | 1320 | 18 | "-" |
| OG_41 | True | 0.315789 | 1038 | 19 | "-" |
| OG_47 | True | 0.157895 | 1173 | 20 | "+" |
| OG_66 | True | 0.052632 | 252 | 21 | "-" |
| OG_65 | True | 0.052632 | 582 | 22 | "+" |
| OG_60 | True | 0.052632 | 255 | 23 | "-" |
| OG_59 | True | 0.052632 | 204 | 24 | "-" |
| OG_46 | False | 0.105263 | 1179 | 25 | "-" |

| G | H | I |
|---|---|---|
| ajima | on Ali | Entro |
| NA | NA | 0 |
| NA | NA | 0 |
| NA | NA | 0 |
| -1.696823 | 0.036281 | 0.007123 |
| -0.520588 | 0.055118 | 0.015197 |
| -1.710969 | 0.064018 | 0.013135 |
| -0.962895 | 0.045375 | 0.011444 |
| -1.010463 | 0.05 | 0.012405 |
| -0.767155 | 0.059394 | 0.015788 |
| -0.466943 | 0.045007 | 0.012496 |
| -0.992799 | 0.03125 | 0.007702 |
| -1.589573 | 0.054386 | 0.011358 |
| -0.637151 | 0.038012 | 0.010317 |
| -1.880047 | 0.059524 | 0.011479 |
| -0.931937 | 0.069156 | 0.017506 |
| -0.334976 | 0.097701 | 0.027896 |
| NA | NA | 0 |
| NA | NA | 0 |
| -0.825819 | 0.011111 | 0.003909 |
| NA | NA | 0.092222 |
| NA | NA | 0 |
| NA | NA | 0 |
| NA | NA | 0 |
| NA | NA | 0 |

|  | U | V | W | X | Y | Z | AA | AB |
|---|---|---|---|---|---|---|---|---|
| t | notati | nder ( | notati | otatio | notati | m Do | Locus | ensus |
| 110 | NA | NA | NA | NA | NA | Thermop | GCA_900 | MTLENFNENL |
| 356 | NA | NA | NA | REFSEQ h | NA | Bacterial | GCA_900 | MARKKTITKE |
| 107 | NA | NA | NA | NA | NA | TspO/MBP | GCA_900 | MNRVKDKRL |
| 593 | NA | NA | NA | sp\|Q65JA | NA | Flavodoxi | GCA_000 | MTLAKIVYASI |
| 356 | NA | NA | BGC0000 | sp\|M1CZ0 | NA | Metallope | GCA_000 | MITLKSPREIE |
| 356 | NA | NA | BGC0000 | NA | NA | Virulence | GCA_000 | MKFLDNIKKN |
| 356 | NA | NA | BGC0000 | NA | VFG0186 | Glycosyl t | GCA_000 | MGSTMSLFIY |
| 023 | NA | NA | BGC0000 | NA | VFG0490 | Glycosyl t | GCA_000 | MQEMVTISIV |
| 356 | NA | NA | BGC0000 | NA | NA | NA | GCA_000 | MSQRLAVVIV |
| 023 | NA | NA | BGC0000 | sp\|A0A0H | VFG0059 | Glycosyl t | GCA_000 | MISVCIATYN |
| 356 | NA | NA | BGC0000 | sp\|P5525 | VFG0186 | Nucleotid | GCA_000 | MKGIILAGGS |
| 023 | NA | NA | BGC0000 | sp\|O2781 | VFG0076 | dTDP-4-d | GCA_000 | MKVIDTKLQD |
| 356 | NA | NA | BGC0000 | sp\|Q9EQ0 | VFG0060 | GDP-man | GCA_000 | MKKIIVTGGA |
| 023 | NA | NA | BGC0000 | sp\|Q9EQ0 | VFG0060 | RmlD sub | GCA_000 | MILITGGNGQ |
| 356 | NA | NA | BGC0000 | sp\|O2200 | VFG0060 | Glycosyl t | GCA_000 | MKVLLIIPAYN |
| 023 | NA | NA | NA | NA | NA | Uncharac | GCA_000 | MLPTVLWLAL |
| 356 | gb\|AAL27 | NA | NA | sp\|P2321 | NA | Acyltrans | GCA_007 | MTKNNDVLDA |
| 665 | NA | html.2020 | NA | NA | NA | Transpos | GCA_007 | MTSLHKNQVK |
| 356 | gb\|ABX54 | NA | NA | sp\|P2321 | NA | Acyltrans | GCA_000 | MTKNNDVLDA |
| 001 | NA | html.2020 | BGC0000 | sp\|P1977 | NA | Transpos | GCA_000 | MTQVHFTLKS |
|  | NA | NA | NA | NA | NA | NA | GCA_000 | MKFAQQLGV |
| 356 | NA | NA | NA | NA | NA | Domain o | GCA_900 | MKKSLSLVFA |
| 023 | NA | NA | NA | NA | VFG0133 | NA | GCA_017 | MPDLKHEDYA |
| 356 | NA | NA | NA | NA | VFG0003 | NA | GCA_017 | MEQIEDFKSA |
| 356 | NA | html.2020 | BGC0000 | sp\|P1977 | NA | Transpos | GCA_000 | MNDFTTEIVK |

Figure 1: overview_of_zol_result_spreadshseet

**Use for dereplication of gene cluster GenBanks to ease visualization with clinker or CORASON**

Another application of zol is to use it for preliminary dereplication for visualization with clinker, CORASON, etc.

zol uses skani to perform dereplication with adjustable options (see `zol --help`).

*Note, skani estimates for ANI and AF become less reliable when working with contigs <10kb, so zol-based dereplication should only be used for gene clusters 10 kb or larger.*

```
# Run zol with dereplication requested
zol -i GenBanks_Directory/ -o zol_Results/ -d

# Reference dereplicated representative GenBanks/gene clusters as input for clinker analysis
clinker zol_Results/Dereplicated_GenBanks/*.gbk -p clinker_visualization.html
```

# selecting parameters for fai and zol

## Selecting parameter values for fai

If the user has previously identified a handful of diverse instances of a gene cluster, they can provide them to zol and request the mode `--select_fai_params_mode` and identify appropriate parameters and command recommendations for fai.

An example report produced looks something like:

```
=======================================================
Recommendations for running fai to find additional instances of gene cluster:
-------------------------------------------------------
Note, this functionality assumes that the known instances of the gene cluster
are representative of the gene cluster/taxonomic diversity you will be searching.
=======================================================
General statistics:
=======================================================
Maximum of maximum E-values observed for any OG 0.000868
Maximum of near-core OG E-values observed:      1.98e-05
Maximum distance between near-core OGs: 14
Median CDS count:       84.0
Median proportion of CDS which are near-core (conserved in 80 percent of gene-clusters):        0.39506
Best representative query gene-cluster instance to use: /home/salamzade/zol_development/showcase_exampl
=======================================================
Parameter recommendations - CPUs set to 4 by default
please provide the path to the prepTG database yourself!
=======================================================
Lenient / Sensitive Recommendations for Exploratory Analysis:
fai --cpus 4 --output_dir fai_Search_Results/ --draft_mode --evalue_cutoff 0.000868 --min_prop 0.1 --sy
-------------------------------------------------------
Strict / Specific Recommendations:
fai --cpus 4 --output_dir fai_Search_Results/ --draft_mode --filter_paralogs --evalue_cutoff 0.000868 -
```

### Prior distributions for fai parameter values for gene cluster families (BiG-SCAPE GCFs) and phage clusters (PhamClust)

Characterized BGCs from MIBiG v3.1 were downloaded and clustered int gene cluster families using BiG-SCAPE. Phage clusters from PhamClust were also gathered. Clusters of similar gene clusters (BGCs or phages) were processed through zol with the `--select_fai_params_mode` requested in batch. Results from the investigation which could be used to set fai parameter settings when looking at BGCs or phages without better prior information available:

## Selecting parameter values for zol

There are some parameters which control the granularity of ortholog group clustering by zol. This includes thresholds for percent identity and coverage for pairs of proteins to be considered as related prior to MCL clustering. The default values of these parameters might be too stringent or conversely too loose depending on the set of gene clusters being investigated.

For best results with zol, if fai was used to identify the gene clusters, we thus advice users to assess the spreadsheet fai produces to see what values for these thresholds might be appropriate!

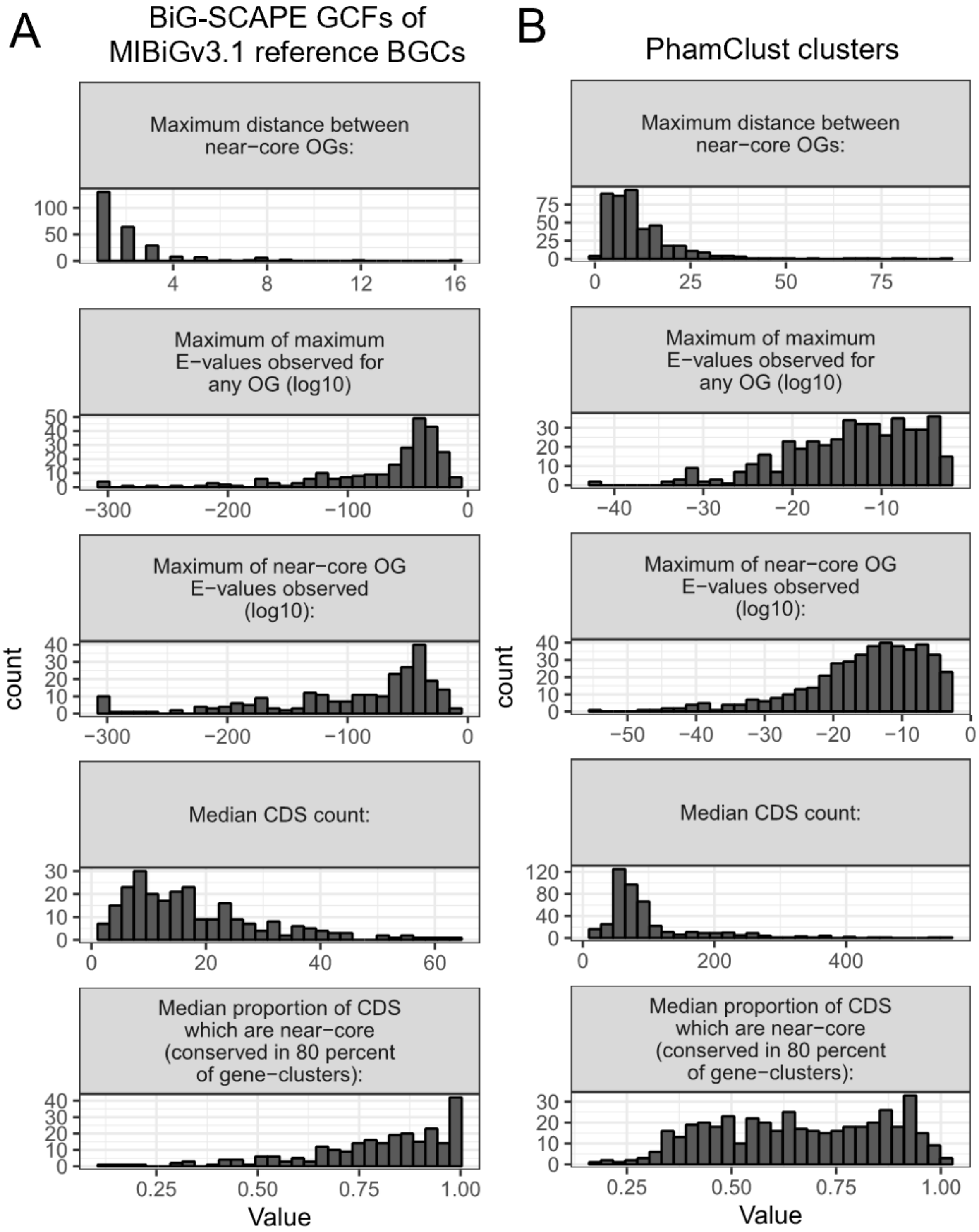


Figure 1: priors__for__fai__parameters__for__bgcs__and__phages

# 5. tutorial - a detailed walkthrough

## Contents

## Note on re-running and writing to the sample results directory

If re-running fai and zol and writing to a previously existing results directory, they will "overwrite" results - but this is only for steps which have not been successfully run already. Checkpoint files are kept in the subdirectory `Checkpoint_Files/`. If users which to completely overwrite results, they can simply delete the resulting directory or if they want to rerun specific steps, they can delete the corresponding checkpoint files.

## Overview: Investigation of the Enterococcal polysaccharide antigen (*epa*) in *E. faecalis* and *E. faecium*

This tutorial will simply walkthrough the test commands in the `run_tests.sh` script to test proper installation and showcase various features in the suite. The test data pertains to the enterococcal polysaccharide antigen encoding locus *epa* in *Enterococcus faecalis* and *Enterococcus faecium.* We perform a more thorough examination of this gene cluster in the zol manuscript.

## Step 1: Download and setup workspace

Lets begin by downloading the test dataset:

```
wget https://github.com/Kalan-Lab/zol/raw/main/test_case.tar.gz
```

Next, we can uncompress it and change directories into it:

```
tar -zxvf test_case.tar.gz
cd test_case/
```

## Step 2: Create a prepTG database of target genomes to search (or download a premade one) :file_folder:

Before we use fai to identify homologous or orthologous instances of gene clusters in a set of target genomes, we must format the target genomes using prepTG.

prepTG takes a directory of genomic assemblies as input, in either FASTA or GenBank format. If GenBank format is provided, the expectation is that CDS features are included (in other words it is a full GenBank). If FASTA files are provided the default operation is to use pyrodigal for bacterial gene calling, however, options exist to use prodigal or minimap instead. minimap was incorporated specifically for eukaryotic genomes provided in FASTA format, where users can also provide a predicted proteome file (another FASTA file with protein sequences) of a reference genome to map those to the remainder of the genomes. We took such an approach in the zol manuscript to map high-quality coding sequence predictions from a reference Aspergillus flavus genome to the remainder of Aspergillus flavus genomes available in NCBI but lacking coding sequence predictions. For prodigal and pyrodigal, usage metagenomics gene calling mode is also available as a configurable option.

Our target genomes, where we will be searching for the *epa* locus can be found in the subdirectory: `Target_Genomes/`. Because we are dealing with bacterial genomes - we can simply run prepTG with default settings as such:

```
prepTG -i Target_Genomes/ -o prepTG_Database/ -c 4 -cst
```

The `-c` option in this case controls the number of parallel threads to use.

> The `-cst` option (new in v1.3.7+) creates a species tree of the target genomes from skani-based ANI estimates (works best for within a species or less diverse genus) + neighbor-joining. This species tree can then be used for visualization purposes downstream in fai. The resulting file will be in the prepTG database folder and named `Species_Tree.nwk`.

### Downloading a premade database

This might take a while, but we have uploaded premade databases of representative genomes for select bacterial taxa (see the premade prepTG dbs wiki page for further info).

Starting in version 1.3.7 of the suite, users can simply issue the command:

```
prepTG -d Enterococcus -o prepTG_Database/
```

> Note, databases are stored on Zenodo. Option should generally work, but download speeds might be slow at times.

### Downloading genomes from NCBI's RefSeq or GenBank databases

Users should also be familiar with the ever so useful `ncbi-genome-download` program which is included automatically in the zol conda environment.

GitHub link: https://github.com/kblin/ncbi-genome-download

```
# Example for downloading all Enterococcus genomes in RefSeq in FASTA format
ncbi-genome-download -F fasta -s refseq -g "Enterococcus" --flat-output -o RefSeq_Enterococcus_Genomes/
```

## Step 3: Search for your query gene cluster in target genomes with fai :mag_right:

### Get your query gene cluster(s) from somewhere in FASTA or GenBank format

fai accepts query gene clusters in multiple formats. For a description of these please see the basic usage examples wiki page.

In the test dataset, we already have a known instance of the *epa* gene cluster from *E. faecalis* provided as a protein FASTA (`Epa_Proteins_from_MIBiG_GenBank.faa`) and in GenBank format (`Epa_MIBiG_GBK/Epa_MIBiG_GenBank.gbk`).

We could also manually assemble a protein FASTA file from literature or online database such as NCBI or KEGG. We could also download the GenBank for the gene cluster directly from MIBiG using wget: `wget https://mibig.secondarymetabolites.org/repository/BGC0000792/BGC0000792.gbk`. Alternatively, we could provide a coordinate for the *epa* along some reference genome, e.g. the *E. faecalis** V583 genome (also provided in the workspace, `Efaecalis_V583_Genome.fasta`). These options enables users to reference the coordinates of interesting they find in literature or cool webservers, such ICEberg or IslandFinder, more easily.

### Running fai

Here is a quick look at how using these options might look.

#### Using a query provided as a FASTA file of proteins

```
fai -pq Epa_Proteins_from_MIBiG_GenBank.faa -tg prepTG_Database/ -o fai_Results/
```

> A special case is using a single gene as a query via the `-sq` option - which gains inspiration from CORASON/EvoMining - more support for this option might be developed in the future.

#### Using a query provided as a GenBank file

```
fai -i Epa_MIBiG_GBK/Epa_MIBiG_GenBank.gbk -tg prepTG_Database/ -o fai_Results/
```

#### Using coordinates along a reference genome

```
fai -r Efaecalis_V583_Genome.fasta -rc NC_004668.1 -rs 2083902 -re 2115174 -tg prepTG_Database/ -o fai_
```

#### Where are the results? :collision:

The set of homologous gene cluster instances identified in target genomes will be located in the subdirectory: `fai_Results/Final_Results/Homologous_Gene_Cluster_GenBanks/`

### Key options in fai to consider:

Here is a quick overview of key options in fai a user should consider. Additional details around the algorithms fai uses to delineate gene can be found on the more info on fai wiki page.

- `-dm` or `--draft_mode` : specifying this option will enable draft-assembly mode searching, which allows for looser search criteria and assumptions that the gene cluster might be split up across multiple scaffolds/contigs. False positives are also likely to be incurred.
- `-fp` or `--filter_paralogs` : specifying this option will specify to filter secondary hits (which might be paralogous instances of the gene cluster) if multiple, overlapping hits for the gene cluster are found in individual target genomes.
- `-e` or `--evalue` : The e-value cutoff for whether to consider a gene in a target genome exhibits similarity to a protein from the query gene cluster using DIAMOND blastp. Default value is 1e-10.
- `-m` or `--min_prop` : The minimum proportion of genes from the query gene cluster needed to report a homologous instance of a query gene cluster. Note genes are actually collapsed into distinct alleles upfront, so this is the minimum proportion of distinct alleles. Default value is 0.5.
- `-mgd` or `--max_genes_disconnect` : The maximum number of CDS features separating two genes in a target genome which exhibit sufficient similarity to a protein from the query gene cluster for them to be considered as part of the same cluster instance. Default is 5.

- `-kpq` or `--key_protein_queries` : A FASTA protein file can be provided with individual proteins that are special and that users can specify separate e-value or conservation thresholds for gene cluster detection using (via the `-kpe` and `-kpm` options). E.g. are you looking for an NRPS-type BGC and there are three key NRPS genes, then you can specify them separately and make your search more stringent while loosening up criteria for detection of auxiliary gene cluster components.
- `-sct` or `--synteic_correlation_threshold` : If a GenBank or coordinates along a reference genome are provided, a syntenic similarity assessment between detected gene clusters and the query gene cluster will be performed based on global correlation gene order similarity. Values closer to 1 are more stringent whereas a value of 0 implies no syntenic filter should be applied. The default value is 0.6.
- `-gdm` or `--gc_delineation_mode` : The gene cluster delineation mode. There are basically two options, for most users we recommend the default setting.
- `-f` or `--flanking_context` : The flanking of a homologous instance of the gene cluster identified in a target genome to include in the resulting GenBank output, useful to explore conservation and annotation of surrounding contexts of the gene cluster downstream in zol. Default is 1000 bp.
- `-gp` or `--generate_plots` : Whether to generate a PDF with plots showcasing the similarity of detected homologous gene cluster instances from target genomes to the query gene cluster. Looks like the following, each bar is a gene along the target genome, the height corresponds to the ratio of the protein in the target genome to the best matching protein from the query gene cluster. The color corresponds to the percent identity (more red = higher identity).

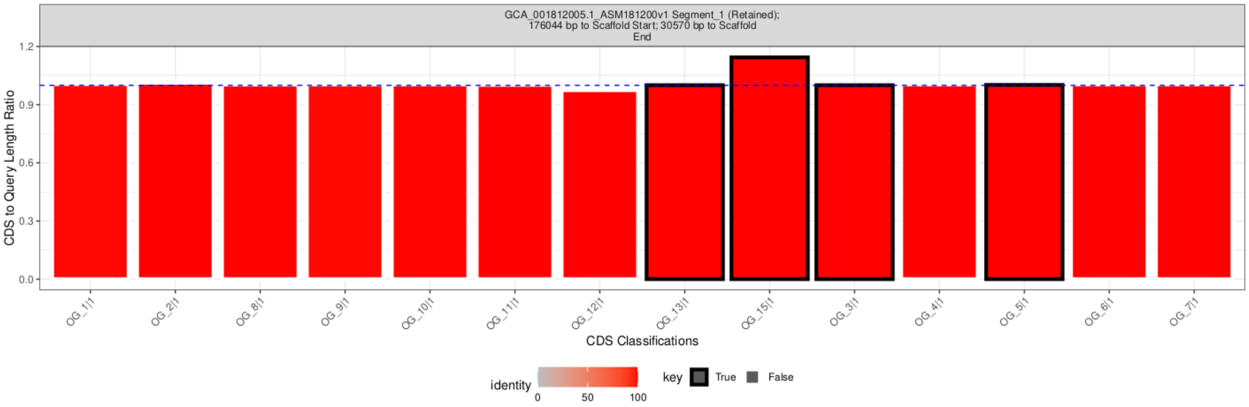


Figure 1: example_visual_from_fai

- `-st` or `--species_tree` : providing this option will allow using a species tree of the target genomes to construct a tree-heatmap figure showing whether the presence of the query gene cluster is widespread or clade specific. It will look something like the following, with darker values indicating a higher bitscore between the query and target proteins:

### Check out cblaster and CAGECAT as alternatives to fai for finding homologous instances of a query gene cluster

zol just takes a set of GenBanks as input and it is definitely possible to use cblaster by Gilchrist et al. 2021 instead of fai to gather gene-clusters in GenBank format; for instance:

```
# use cblaster to search for homologous co-clusters in NCBI genomes
cblaster search -qf queries.faa -s cblaster_results.json

# use cblaster to extract GenBannks of homologous gene-clusters detected
cblaster extract_clusters session.json -o example_directory/
```

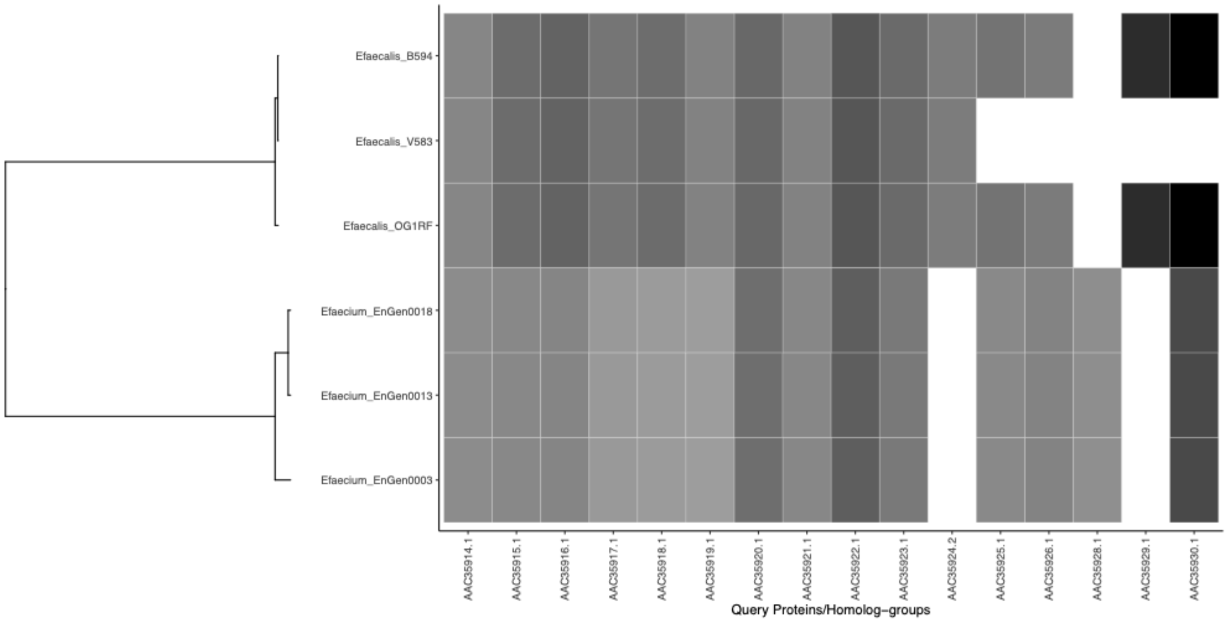


Figure 2: example_phylogenetic_view_by_fai

CAGECAT by van den Belt et al. 2023 is an awesome webserver for running cblaster and clinker.

cblaster features the fantastic ability to use NCBI's computational infrastructure to perform alignment thereby minimizing the need for local resources.

## Step 4: Manual selection of homologous/orthologous gene cluster instances identified by fai :eyes:

fai will automatically produce an XLSX spreadsheet which allows users to assess homologous hits to the query gene cluster at scale. Most columns feature automatic conditional formatting to ease user assessment of quantitative fields. It is inspired by the heatmap visuals from cblaster, but the spreadsheet format allows further flexibility for users to sort on various columns.

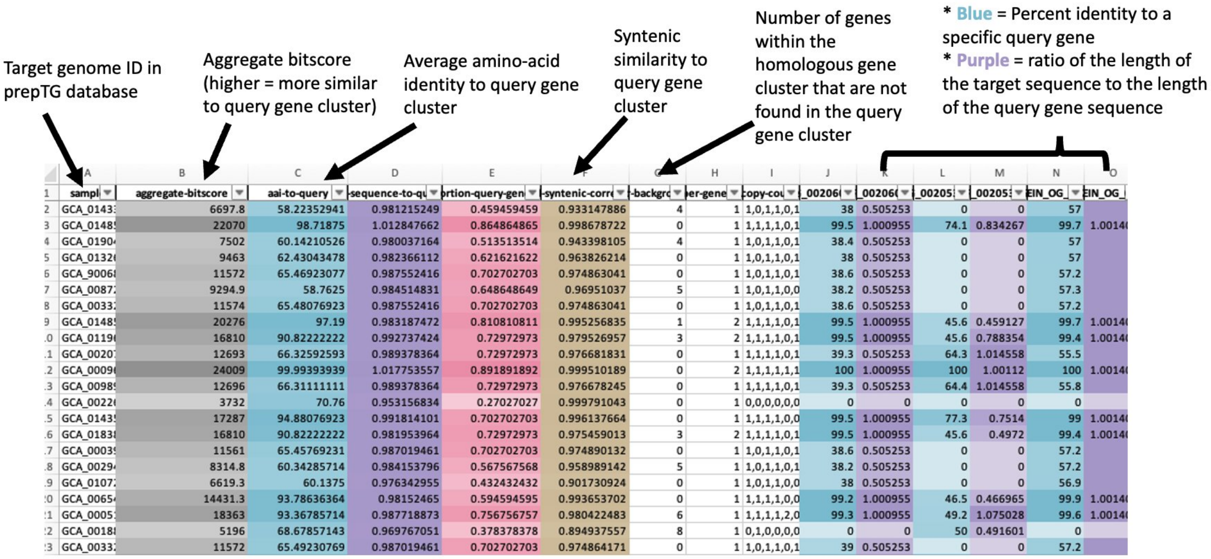
Users can more freely apply filters using this spreadsheet and select specific homologous gene cluster instances for follow-up analysis in zol. To further support such manual curation of fai's hits, we also include the script `selectSpecificGeneClusters.py` which can be provided with the fai results directory with a text file listing either (1) sample names (first column in the sheet "Genome Wide - Results") or (2) individual gene-cluster instances (second column in the sheet "Gene Cluster - Results").

```
selectSpecificGeneClusters.py -i fai_Results/ -s select_instances.txt -o select_instances_dir/  -t insta
```

## Step 5: Generate a tabular report with evolutionary trends, annotation info, and conservation stats using zol :page_with_curl: :chart_with_upwards_trend:

Finally, we can investigate the relationships between identified homologous instances of the query gene cluster using zol.

We can do this using the following basic command:


Target genome ID in prepTG database
Aggregate bitscore (higher = more similar to query gene cluster)
Average amino-acid identity to query gene cluster
Syntenic similarity to query gene cluster
Number of genes within the homologous gene cluster that are not found in the query gene cluster
* Blue = Percent identity to a specific query gene
* Purple = ratio of the length of the target sequence to the length of the query gene sequence

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | sampl | aggregate-bitscore | aai-to-query | -sequence-to-qu | ortion-query-gen | -syntenic-corre | -backgro | er-gene | copy-cou | _00206 | _00206 | _00205 | _00205 | EIN_OG_ | EIN_OG_ |
| 2 | GCA_0143 | 6697.8 | 58.22352941 | 0.981215249 | 0.459459459 | 0.933147886 | 4 | 1 | 1,0,1,1,0,1 | 38 | 0.505253 | 0 | 0 | 57 | |
| 3 | GCA_0148 | 22070 | 98.71875 | 1.012847662 | 0.864864865 | 0.998678722 | 0 | 1 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 74.1 | 0.834267 | 99.7 | 1.0014 |
| 4 | GCA_0190 | 7502 | 60.14210526 | 0.980037164 | 0.513513514 | 0.943398105 | 4 | 1 | 1,0,1,1,0,1 | 38.4 | 0.505253 | 0 | 0 | 57 | |
| 5 | GCA_0132 | 9463 | 62.43043478 | 0.982366112 | 0.621621622 | 0.963826214 | 0 | 1 | 1,0,1,1,0,1 | 38 | 0.505253 | 0 | 0 | 57 | |
| 6 | GCA_9006 | 11572 | 65.46923077 | 0.987552416 | 0.702702703 | 0.974863041 | 0 | 1 | 1,0,1,1,0,1 | 38.6 | 0.505253 | 0 | 0 | 57.2 | |
| 7 | GCA_0087 | 9294.9 | 58.7625 | 0.984514831 | 0.648648649 | 0.96951037 | 5 | 1 | 1,0,1,1,0,0 | 38.2 | 0.505253 | 0 | 0 | 57.3 | |
| 8 | GCA_0033 | 11574 | 65.48076923 | 0.987552416 | 0.702702703 | 0.974863041 | 0 | 1 | 1,0,1,1,0,1 | 38.6 | 0.505253 | 0 | 0 | 57.2 | |
| 9 | GCA_0148 | 20276 | 97.19 | 0.983187472 | 0.810810811 | 0.995256835 | 1 | 2 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 45.6 | 0.459127 | 99.7 | 1.0014 |
| 10 | GCA_0119 | 16810 | 90.82222222 | 0.992737424 | 0.72972973 | 0.979526957 | 3 | 2 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 45.6 | 0.788354 | 99.4 | 1.0014 |
| 11 | GCA_0020 | 12693 | 66.32592593 | 0.989378364 | 0.72972973 | 0.976681831 | 0 | 1 | 1,1,1,1,0,1 | 39.3 | 0.505253 | 64.3 | 1.014558 | 55.5 | |
| 12 | GCA_0009 | 24009 | 99.99393939 | 1.017753557 | 0.891891892 | 0.999510189 | 0 | 2 | 1,1,1,1,1,1 | 100 | 1.000955 | 100 | 1.00112 | 100 | 1.0014 |
| 13 | GCA_0098 | 12696 | 66.31111111 | 0.989378364 | 0.72972973 | 0.976678245 | 0 | 1 | 1,1,1,1,0,1 | 39.3 | 0.505253 | 64.4 | 1.014558 | 55.8 | |
| 14 | GCA_0022 | 3732 | 70.76 | 0.953156834 | 0.27027027 | 0.999791043 | 0 | 1 | 0,0,0,0,0,0 | 0 | 0 | 0 | 0 | 0 | |
| 15 | GCA_0143 | 17287 | 94.88076923 | 0.991814101 | 0.702702703 | 0.996137664 | 0 | 1 | 1,1,1,1,0,0 | 99.5 | 1.000955 | 77.3 | 0.7514 | 99 | 1.0014 |
| 16 | GCA_0183 | 16810 | 90.82222222 | 0.981953964 | 0.72972973 | 0.975459013 | 3 | 2 | 1,1,1,1,0,1 | 99.5 | 1.000955 | 45.6 | 0.4972 | 99.4 | 1.0014 |
| 17 | GCA_0003 | 11561 | 65.45769231 | 0.987019461 | 0.702702703 | 0.974890132 | 0 | 1 | 1,0,1,1,0,1 | 38.6 | 0.505253 | 0 | 0 | 57.2 | |
| 18 | GCA_0029 | 8314.8 | 60.34285714 | 0.984153796 | 0.567567568 | 0.958989142 | 5 | 1 | 1,0,1,1,0,1 | 38.2 | 0.505253 | 0 | 0 | 57.2 | |
| 19 | GCA_0107 | 6619.3 | 60.1375 | 0.976342955 | 0.432432432 | 0.901730924 | 0 | 1 | 1,0,1,1,0,0 | 38 | 0.505253 | 0 | 0 | 56.9 | |
| 20 | GCA_0065 | 14431.3 | 93.78636364 | 0.98152465 | 0.594594595 | 0.993653702 | 0 | 1 | 1,1,1,1,0,0 | 99.2 | 1.000955 | 46.5 | 0.466965 | 99.9 | 1.0014 |
| 21 | GCA_0005 | 18363 | 93.36785714 | 0.987718873 | 0.756756757 | 0.980422483 | 6 | 1 | 1,1,1,1,2,0 | 99.3 | 1.000955 | 49.2 | 1.075028 | 99.6 | 1.0014 |
| 22 | GCA_0018 | 5196 | 68.67857143 | 0.969767051 | 0.378378378 | 0.894937557 | 8 | 1 | 0,1,0,0,0,0 | 0 | 0 | 50 | 0.491601 | 0 | |
| 23 | GCA_0033 | 11572 | 65.49230769 | 0.987019461 | 0.702702703 | 0.974864171 | 0 | 1 | 1,0,1,1,0,1 | 39 | 0.505253 | 0 | 0 | 57.2 | |

Figure 3: overview_of_fai_result_spreadsheet

```
zol -i fai_Results/Final_Results/Homologous_Gene_Cluster_GenBanks/ -o zol_Results/ -c 10
```

Once more `-c` is just specifying the number of threads to use.

I highly recommend using the option `-cd` or `--custom_database` to provide proteins from the query gene cluster used for fai as an extra database for annotation. This will map these proteins to the ortholog groups so you have a point of reference to the original query gene cluster.

```
zol -i fai_Results/Final_Results/Homologous_Gene_Cluster_GenBanks/ -cd Epa_Proteins_from_MIBiG_GenBank.
```

In the final Excel spreadsheet generated, one of the columns will now feature annotations from the custom database (protein FASTA file) provided.

**Where are the results? :collision:**

The final results, once zol is done, can be found in the subdirectory at `zol_Results/Final_Results/` with the major results file being the XLSX spreadsheet `Consolidated_Report.xlsx`.

### Key options in zol to consider:

Here are a list and brief description of key options to consider in zol:

- `it` or `identity_threshold`, `ct` or `coverage_threshold`, and `et` or `evalue_threshold` : These are key thresholds used for InParnoid-type ortholog grouping in zol. The default values might be appropriate for some gene clusters being studied but looser or more stringent criteria might be beneficial to your analysis for other gene clusters.
- `-fl` or `--filter_low_quality` : Filter out gene clusters which feature alot of missing bases (>10%). I almost always issue this, but it is not turned on by default.
- `-fd` or `--filter_draft_quality` : Filter out gene clusters which were found near scaffold edges. I also usually specify this, especially when interested in gene conservation. This is because it becomes tricky to assess whether a gene is missing simply because the gene cluster is fragmented due to assembly issues or because it is actually missing. If fai is used in "draft mode", users could provide multi-record GenBanks and choose to skip this argument, I have not tested the use of multi-record GenBanks much.

- `-r` or `--rename_lt` : If CDS features in input GenBanks don't have `locus_tag` identifiers, just generate them from scratch.
- `-d` or `--dereplicate` : Dereplicate gene clusters to remove highly similar versions. Can be used to reduce the complexity of the analysis. **It is recommended to consider using dereplication if you are dealing with thousands of genomes to speed things up and keep the amount of harddisk space needed (albeit temporarily) down!!!**.
- `-ri` or `--reinflate` : This flag tells zol to reinflate the ortholog groups determined using a dereplicated/represntative set of gene clusters with proteins from the full set of gene clusters provided as input. Note, this reinflation currently works for grouping proteins which are very similar to an already (ortholog) grouped protein from the representative gene clusters. **dereplication must be specified alongside this option**. This approach is inspired by Roary.
- `-cd` or `--custom_database` : See above section, basically it is usually nice to have proteins from the known gene cluster instance to reference instead of just arbitrary ortholog group identifiers, you can do this by providing a protein FASTA file for custom annotation of ortholog groups.

### Pair up zol with the visual capabilities of clinker

A key feature of zol is the option to dereplicate gene clusters - which can allow zol to pair nicely with interactive gene cluster visualization software, in particular clinker. This is because users can apply zol to first select representative gene clusters that are sufficiently distinct from each other and then visualize only this set using clinker (which also just takes GenBank files of gene clusters as input) to help with computational scalability and the responsiveness of the awesome HTML reports/figures.

Assuming dereplication is requested, representative gene cluster GenBank files will be found in the directory: `zol_Results/Dereplicated_GenBanks/`. Users can provide the smaller set of gene cluster instances in this format as input to clinker.

## Step 6: Comparative analysis of gene clusters with zol :apple: :tangerine:

Users can perform comparative analyses between sets of gene clusters in zol. This is done by specifying a focal set of gene clusters by name in a text file and, optionally, a complementary set of gene clusters. These gene cluster sets could be instances belonging to a certain taxonomic group or associated with a certain environment.

We can demonstrate running a comparative analysis using the testing dataset. Specifically, let's say we want to compare instances of *epa* from *E. faecalis* to instances of *epa* from *E. faecium*. And our `fai_Results/Final_Results/Homologous_Gene_Cluster_GenBanks/` has the following files as viewed using `ls`:

```
Efaecalis_B594_fai-gene-cluster-1.gbk
Efaecalis_V583_fai-gene-cluster-1.gbk
Efaecium_EnGen0013_fai-gene-cluster-1.gbk
Efaecalis_OG1RF_fai-gene-cluster-1.gbk
Efaecium_EnGen0003_fai-gene-cluster-1.gbk
Efaecium_EnGen0018_fai-gene-cluster-1.gbk
```

We can define the set of gene clusters belonging to *E. faecalis* by name, one per line, in a file to provide to zol's `-f` argument as such:

```
Efaecalis_B594_fai-gene-cluster-1.gbk
Efaecalis_OG1RF_fai-gene-cluster-1.gbk
Efaecalis_V583_fai-gene-cluster-1.gbk
```

Then, we can simply run zol in comparative mode using the following command. Because all other gene clusters are from *E. faecium*, we don't need to formally specify the comparing set via the `-fc` option. By default all gene clusters not defined in the focal gene cluster set will be used as the comparing set when only `-f`.

```
zol -i fai_Results/Final_Results/Homologous_Gene_Cluster_GenBanks/ \
    -f E_faecalis_GeneCluster_Listing.txt \
    -o zol_Results_with_Comparaitve_Analysis/ -c 10
```

# Citations for dependencies, databases, and related software

***Please consider citing the following accordingly!***

- **pyrodigal**, **prodigal**, and **miniprot** for gene-calling/mapping.
  - Pyrodigal: Python bindings and interface to Prodigal, an efficient method for gene prediction in prokaryotes
  - Prodigal: prokaryotic gene recognition and translation initiation site identification
  - Protein-to-genome alignment with miniprot
- **MUSCLE5** for performing multiple sequence alignments and **PAL2NAL** for converting to codon alignments.
  - Muscle5: High-accuracy alignment ensembles enable unbiased assessments of sequence homology and phylogeny
  - PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments
- **DIAMOND** for alignments in determining ortholog groups and **FastTree2** for subsequent phylogeny construction.
  - Fast and sensitive protein alignment using DIAMOND
  - FastTree 2 ??? Approximately Maximum-Likelihood Trees for Large Alignments
- **CD-HIT** for query protein clustering in fai and 're-inflation' approach in zol.
  - CD-HIT: accelerated for clustering the next-generation sequencing data
- **HyPhy** and **FASTME** for selection analyses.
  - HyPhy: hypothesis testing using phylogenies
  - GARD: a genetic algorithm for recombination detection
  - FUBAR: A Fast, Unconstrained Bayesian AppRoximation for Inferring Selection
  - FastME 2.0: A Comprehensive, Accurate, and Fast Distance-Based Phylogeny Inference Program
- **skani** for dereplication of gene-clusters/GenBanks.
  - Fast and robust metagenomic sequence comparison through sparse chaining with skani
- **antiSMASH, GECCO, DeepBGC, geNomad, MOB-suite, PhiSpy, VIBRANT**, or **ICEfinder** if you used to identify a BGC, phage, plasmids, or ICEs.
  - antiSMASH 7.0: new and improved predictions for detection, regulation, chemical structures and visualisation
  - Accurate de novo identification of biosynthetic gene clusters with GECCO
  - A deep learning genome-mining strategy for biosynthetic gene cluster prediction
  - ICEberg 2.0: an updated database of bacterial integrative and conjugative elements
  - Identification of mobile genetic elements with geNomad
  - PhiSpy: a novel algorithm for finding prophages in bacterial genomes that combines similarity- and composition-based strategies
  - MOB-suite: software tools for clustering, reconstruction and typing of plasmids from draft assemblies
- **PFAM, KEGG, NCBI's PGAP, MIBiG, VOG, PaperBlast, VFDB, CARD,** and **ISFinder** databases used for annotation.
  - Pfam: The protein families database in 2021
  - KofamKOALA: KEGG Ortholog assignment based on profile HMM and adaptive score threshold
  - RefSeq: expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation

  - MIBiG 3.0: a community-driven effort to annotate experimentally validated biosynthetic gene clusters
  - PaperBLAST: Text Mining Papers for Information about Homologs
  - VFDB 2022: a general classification scheme for bacterial virulence factors
  - CARD 2020: antibiotic resistome surveillance with the comprehensive antibiotic resistance database
  - ISfinder: the reference centre for bacterial insertion sequences

- **DIAMOND** and **PyHMMER** for performing functional annotations against consensus ortholog group sequences in zol.
  - Fast and sensitive protein alignment using DIAMOND
  - PyHMMER: a Python library binding to HMMER for efficient sequence analysis

- **lsaBGC, BiG-SCAPE/CORASON, cblaster/CAGECAT, BiG-SLICE, vConTACT v2.0**, or **IslandCompare** studies if you used them to identify homologous gene clusters.
  - Evolutionary investigations of the biosynthetic diversity in the skin microbiome using lsaBGC
  - A computational framework to explore large-scale biosynthetic diversity
  - cblaster: a remote search tool for rapid identification and visualization of homologous gene clusters
  - CAGECAT: The CompArative GEne Cluster Analysis Toolbox for rapid search and visualisation of homologous gene clusters
  - BiG-SLiCE: A highly scalable tool maps the diversity of 1.2 million biosynthetic gene clusters
  - Taxonomic assignment of uncultivated prokaryotic virus genomes is enabled by gene-sharing networks
  - Enabling genomic island prediction and comparison in multiple genomes to investigate bacterial evolution and outbreaks

# 7. premade prepTG databases

We provide premade databases for 18 bacterial taxa (mostly at the genus level). These databases are not all inclusive - but fai & zol certainly have the capabilities to handle searches on 5,000+ genomes, as we showed in the manuscript. Comprehensive databases can further be set up using the `-g` argument in prepTG, which takes the name of a genus or species from GTDB R214.

Rather these premade databases only contain distinct representative genomes selected using our tool skDER with the greedy clustering approach to sufficiently sample the known pangenome space of the taxa. This is to keep the size of the databases relatively low to aid with download speeds (not super fast currently as is).

The databases are stored on Zenodo (ESKAPE genera, BGC-rich taxa, and other commonly studied genera) and also feature GToTree based phylogenies which can be used as input to the `-st` argument in fai to generate phylogenetic-heatmaps showcasing the presence of query gene clusters.

```
- Acinetobacter - 1,643 rep genomes (17.8% of 9,221 total genomes considered)
- Bacillales - 3,150 rep genomes (35.9% of 8,766 total genomes considered)
- Corynebacterium - 726 rep genomes (43.0% of 1,688 total genomes considered)
- Cutibacterium - 27 rep genomes (5.4% of 502 total genomes considered)
- Enterobacter - 878 rep genomes (19.9% of 4,408 total genomes considered)
- Enterococcus - 937 rep genomes (14.6% of 6,426 total genomes considered)
- Escherichia - 2,436 rep genomes (7.1% of 34,358 total genomes considered)
- Klebsiella - 1,022 rep genomes (5.6% of 18,145 total genomes considered)
- Lactobacillus - 541 rep genomes (30.9% of 1,747 total genomes considered)
- Listeria - 353 rep genomes (6.9% of 5,062 total genomes considered)
- Micromonospora - 211 rep genomes (73.3% of 288 total genomes considered)
- Mycobacterium - 744 rep genomes (6.9% of 10,657 total genomes considered)
- Neisseria - 414 rep genomes (12.8% of 3,235 total genomes considered)
- Pseudomonas - 2,666 rep genomes (18.9% of 14,066 total genomes considered)
- Salmonella - 308 rep genomes (2.2% of 14,109 total genomes considered)
- Staphylococcus - 496 rep genomes (2.5% of 19,627 total genomes considered)
- Streptococcus - 2,452 rep genomes (13.3% of 18,492 total genomes considered)
- Streptomyces - 1,555 rep genomes (57.7% of 2,697 total genomes considered)
```

# 8. overview of prior updates

## Updates

### version 1.3.20

- Update extraction of gene cluster GenBank files from full GenBank files in fai to be much more efficient in fai. Time difference mostly noticeable for metagenomic application where full GenBank files can be quite large.

### version 1.3.19

### Major Updates:

- Slight update to core ortholog group determining algorithm in zol to reduce memory consumption and aid scalability without a dereplicaiton/re-inflation approach.
- Slight update to processing of miniprot protein mappings to account for overlap in exon coordinates (best scoring exon is selected in such cases) in prepTG.
- New mode in zol where users can provide known instances of a gene cluster and determine appropriate parameters for searching for additional instances using fai.

### Minor Updates:

- New script to extract proteins from GenBank files into FASTA format, extractProteinsFromGenBank.py.

### version 1.3.18

- Add option to prepTG to easily/automatically create databases for any bacterial genus/species in GTDB.

### version 1.3.17

- Update fai catching of cases when no homologous BGC instance is found among target genomes.
- Round metrics in fai's report.
- Temporarily remove abon, atpoc, apos from Docker wrapper as these are not yet working - will need to update the bash script for simplifying docker usage at some point later.

### version 1.3.12-1.3.16

- Introduce apos (assess plasmid-ome similarity) and atpoc (assess temperate phage-ome conservation) to assess conservation of a focal sample's plasmids and phages across some set of target/database genomes (e.g. all other genomes in the same species/genus as the sample)
- Add prodigal-gv option in prepTG and fai.
- Add simple BLASTp search option in place of fai for abon.
- Make minor corrections for newly introduced programs.

### version 1.3.11

- Introduce abon!
- Update links to newer versions of precompiled prepTG databases for select bacterial taxa.
- Update wiki documentation.

### version 1.3.10

- Introduce clean up option in fai.
- Reorganize fai's results directory.
- Generate Tiny AAI figure and an XLSX spreadsheet in fai to allow for manual curation of homologous gene clusters detected.

### version 1.3.8

- Update script for downloading annotation databases to account for changes in naming structure in the tar.gz directory with PGAP HMMs.

### version 1.3.7

- Add option to prepTG to download premade databases for certain bacterial taxa/genera hosted on Zenodo.
- Add option to prepTG to construct a species tree based on skani ANI + neighbor-joining on Zenodo.
- Add option to provide species tree in fai and generate a phylo-heatmap of gene cluster searching results.
- Loosen restrictions around the need for a core ortholog group in zol analysis.

### version 1.3.6

- Fix conditional statement in determination of 'consensus directionality' in zol - should be flipped.

### version 1.3.5

- Fix mis-spelling of "Oomolog Group" to "Ortholog Group" in consolidated zol report.

### version 1.3.4

- Fix mismapping of parameter names and arguments in file for provenane for fai (introduced in 1.3.3 after incorporation of single query mode).
- Add consideration point for dereplication in zol help and README to only be used when working with gene-clusters >10kb.

### version 1.3.3

- Correct and clairfy usage of "key protein" filters in fai.
- Introduce single query mode in fai, whereby users can use a single gene as a query to look at differences in surrounding context CORASON style.
- Add miniprot (v0.7) dependency to conda yaml file (and planning to bioconda).

### version 1.3.2

- Allow for failures of specific databases (i.e., if hosting server goes down) in `setup_annotation_dbs.py`.

### version 1.3.1

- Update for release.

### version 1.3.0

- Add better support for query GenBanks without locus tags for CDS features in fai & clearer message to simply use the `-r/--rename_lt` flag to automatically rename locus tags if this is the case for input GenBanks for zol.
- Switch to pyhmmer for faster annotation in zol.

### version 1.2.10

- Update CITATION.cff

### version 1.2.9

- Minor changes to code documentation and updates to citation references README.
- Added reporting on steps to console for prepTG.
- Slight updates to plotting function in fai to allow more robust parsing of GenBanks.

### version 1.2.8

- Update README to add Bioconda installation guide.
- Add more comprehensive comments to python modules with the bulk of the code.
- Add traceback statement to all functions to generate detailed reports of what might be causing issues if they arise.
- Switch to consistently using the term ortholog groups (instead of ortholog groups) in the code/messages/results/comments.
- Updated to more flexible inputting of query GenBanks in fai.
- Corrected processing of cases where GenBanks with CDS features are provided as ready to go in prepTG.

### version 1.2.6 & 1.2.7

- Additional changes to allow for better incorporation into bioconda.

### version 1.2.5

- Additional safety for when statistics are unavailable to incorporate into the consolidated report.

### version 1.2.4

- Docker set up should now work.
- fixed bug introduced in 1.2.3 related to new names for arguments in prepTG in prepTG
- note, will update bioconda recipe after release to get size of release tar.gz.

### version 1.2.3

- updated argument names to prepTG.
- updated the way version information was being reported in programs to make more compatible with bioconda.
- added initial attempt at Dockerfile for creating Docker image and auxiliary scripts to ease usage.
- will likely make another update or two in the near future to get Docker and bioconda options working.

### version 1.2.2

- added initial attempt at bioconda recipe - no changes to core programs.
- introduced ZOL (all captials) - wrapper of the 3 main programs - for use as entrypoint in Docker image.

### Version 1.2.1

- add line in beginning of fai to request "fork" method for multiprocessing to work on macOS with python >=v3.8.
- clean up unused functions and simplify yaml file for specifying conda environment.

### Minor Update - 05/05/2023

- update parsing of PGAP HMMs directory after extracting with tar.

### Version 1.2/1.02

- prepTG sample to GenBank relations now specified locally so creation of database is not locked into one location.
- Individual pickle files produced by prepTG per genome/metagenome for lower memory use with fai.
- New "Gene-Clumper" mode for gene-cluster discovery in fai, which is now the default.
- Fixed bug pertaining to overlap between merged gene-clusters based on `--max_gene_disconnect` parameter when using "HMM" mode.
- Improved filtering and retention of GenBanks in zol.
- Fixed bug in re-inflation method in zol.

### Version 1.1/1.01

- Remove unused individual proteome files in prepTG database directory.
- Store only gene-location information for scaffolds with hits by query proteins in fai to keep memory use low.
- Introduce parallelization to HMM step of fai and use global variables to access common data without duplicating in memory.
- Improve parsing of different input formats for fai and generate new PDF at end mapping individual protein names to non-redundantified protein queries.
- Declare "< 3 segrating sites found" as reason for inability to calculate Tajima's D instead of just "NA", which could also arise from not enough sequences or the sequence length threshold being met.

## 9.1 more info on abon

## abon - Assess Bgc-Ome Novelty

**abon** takes as input antiSMASH and/or GECCO results directories for a single sample together with a prepTG (target genomes) database to determine how unique the sample's BGC-ome is to the genomes in the database. Its development is inspired by studies which have shown that BGCs are often co-regulated (see: Beyond the Biosynthetic Gene Cluster Paradigm: Genome-Wide Coexpression Networks Connect Clustered and Unclustered Transcription Factors to Secondary Metabolic Pathways) and that secondary/specialized metabolites can be the product of additional genes across the genome (e.g. as described in these two nice studies Kim and Lee 2012 & Mohite et al. 2022) and potentially multiple BGCs.

Importantly, abon will parse out "key" biosynthetic CDS features to enable more stringent requirement of their presence while allowing for more leniency in the presence of auxiliary BGC genes. For antiSMASH BGCs, these are CDS features marked with `rule-based-clusters`. For GECCO BGCs, these are CDS with domains bearing the most "weight" in the CRF detection of BGCs (see: https://github.com/Kalan-Lab/lsaBGC/pull/11 for more info).

The specific cutoffs used in fai for gene cluster detection in target genomes can be adapted as needed. Alternatively, a simple BLASTp search can be performed instead to determine all homologs of proteins for each BGC from the focal sample in target genomes regardless of whether they are similarly co-located or not.

> Note, to assess how individual BGCs relate to cataloged/known BGCs or gene cluster families (GCFs), we recommend the awesome BiG-FAM webserver

### Novelty of the *B. subtilis* st.168 BGC-ome relative to other Bacilllales

The following is a mini-tutorial on using abon to investigate the novelty of the BGC-ome of *Bacillus subtilis* st. 168 to representative Bacillales genomes we made available in a precompiled prepTG database.

First, lets download the query genome of interest.

```
# Download genome from NCBI
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/045/GCF_000009045.1_ASM904v1/GCF_000009045.1_A

# Uncompress it & rename it
gunzip GCF_000009045.1_ASM904v1_genomic.fna.gz
mv GCF_000009045.1_ASM904v1_genomic.fna Bsubtilis_st168.fasta
```

Next, we can run antiSMASH and GECCO to call BGCs

```
# in some conda environment or setting with antiSMASH available
antismash --output-dir Bsubtilis_st168_antiSMASH_Results/ \
          --genefinding-tool prodigal Bsubtilis_st168.fasta

# in some conda environment or setting with GECCO available
gecco run --genome Bsubtilis_st168.fasta -o Bsubtilis_st168_GECCO_Results/
```

Next, we can setup the precompiled database of Bacillales representative genome using prepTG:

```
# in zol's conda environment or via the Docker wrapper:
prepTG -d Bacillales -o Bacillales_Reps_prepTG_Database/
```

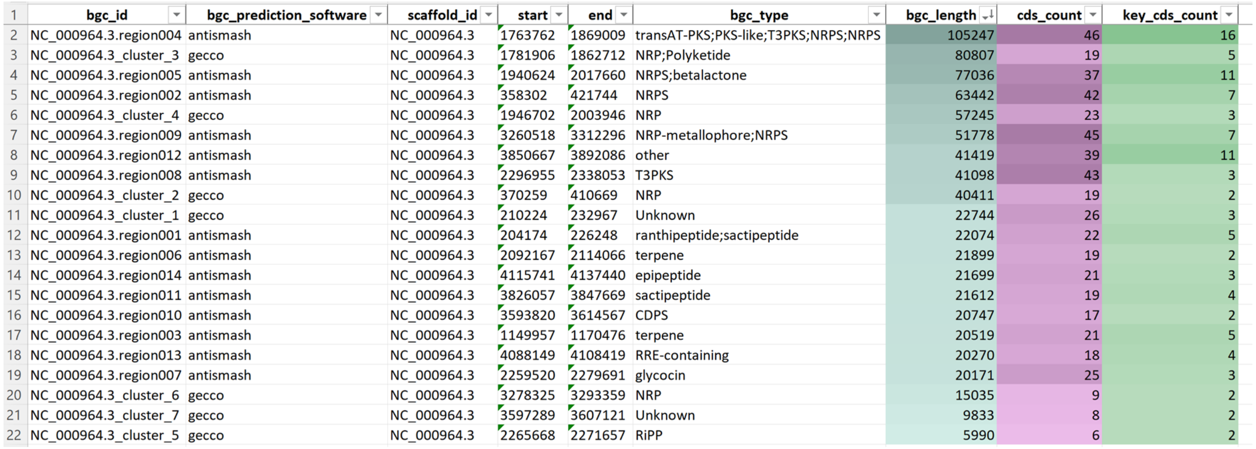
Now we are ready to run abon!

```
abon -tg Bacillales_Reps_prepTG_Database/ -a Bsubtilis_st168_antiSMASH_Results/ \
     -g Bsubtilis_st168_GECCO_Results/ -o abon_Results/ -c 20
```

Note, this can take a while as it will involve running fai X times (where X is the number of BGCs in the focal sample of interest).

### The result!

Similar to fai and zol's major results, abon also primarily produces an XLSX spreadsheet. On the first tab of abon's results XLSX spreadsheet, is an overview of the focal sample's antiSMASH and/or GECCO biosynthetic gene clusters:



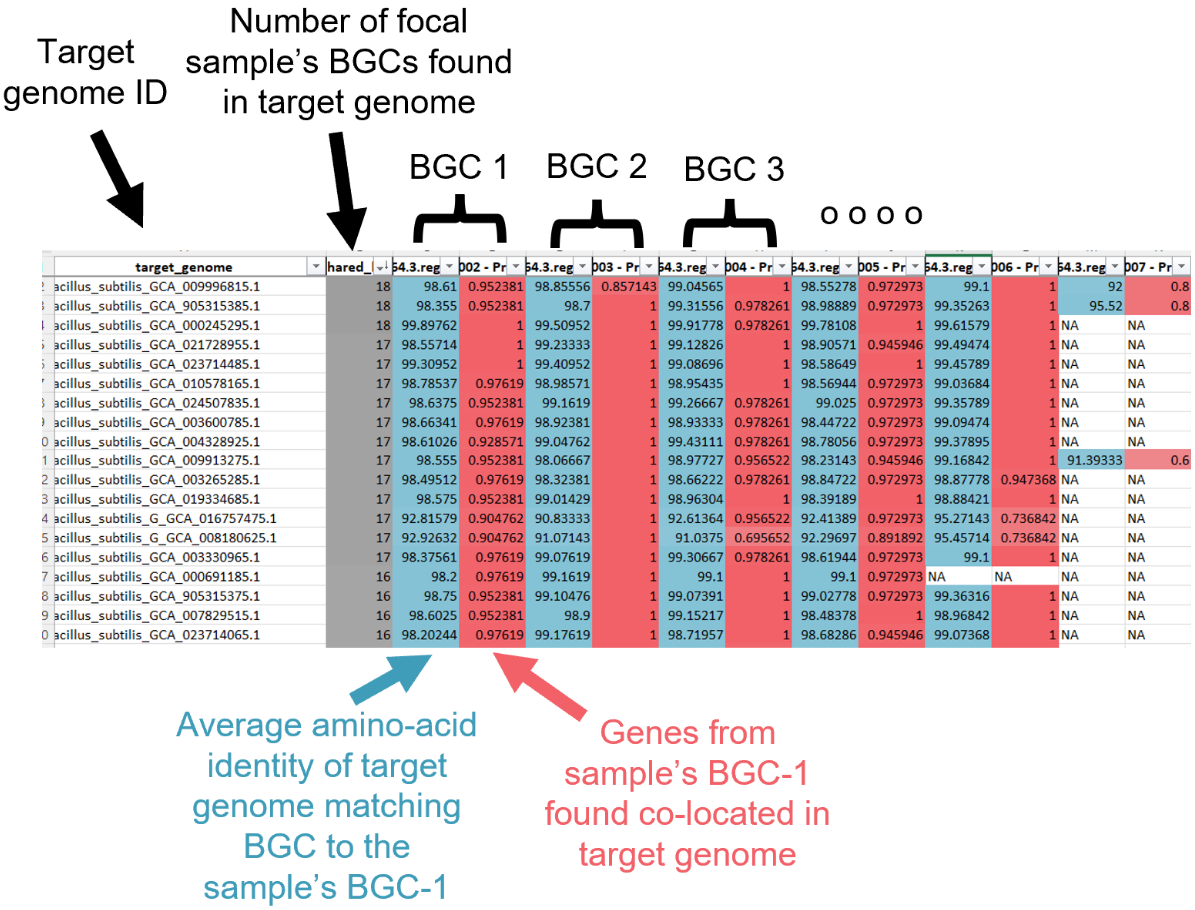
| 1 | bgc_id | bgc_prediction_software | scaffold_id | start | end | bgc_type | bgc_length | cds_count | key_cds_count |
|---|---|---|---|---|---|---|---|---|---|
| 2 | NC_000964.3.region004 | antismash | NC_000964.3 | 1763762 | 1869009 | transAT-PKS;PKS-like;T3PKS;NRPS;NRPS | 105247 | 46 | 16 |
| 3 | NC_000964.3_cluster_3 | gecco | NC_000964.3 | 1781906 | 1862712 | NRP;Polyketide | 80807 | 19 | 5 |
| 4 | NC_000964.3.region005 | antismash | NC_000964.3 | 1940624 | 2017660 | NRPS;betalactone | 77036 | 37 | 11 |
| 5 | NC_000964.3.region002 | antismash | NC_000964.3 | 358302 | 421744 | NRPS | 63442 | 42 | 7 |
| 6 | NC_000964.3_cluster_4 | gecco | NC_000964.3 | 1946702 | 2003946 | NRP | 57245 | 23 | 3 |
| 7 | NC_000964.3.region009 | antismash | NC_000964.3 | 3260518 | 3312296 | NRP-metallophore;NRPS | 51778 | 45 | 7 |
| 8 | NC_000964.3.region012 | antismash | NC_000964.3 | 3850667 | 3892086 | other | 41419 | 39 | 11 |
| 9 | NC_000964.3.region008 | antismash | NC_000964.3 | 2296955 | 2338053 | T3PKS | 41098 | 43 | 3 |
| 10 | NC_000964.3_cluster_2 | gecco | NC_000964.3 | 370259 | 410669 | NRP | 40411 | 19 | 2 |
| 11 | NC_000964.3_cluster_1 | gecco | NC_000964.3 | 210224 | 232967 | Unknown | 22744 | 26 | 3 |
| 12 | NC_000964.3.region001 | antismash | NC_000964.3 | 204174 | 226248 | ranthipeptide;sactipeptide | 22074 | 22 | 5 |
| 13 | NC_000964.3.region006 | antismash | NC_000964.3 | 2092167 | 2114066 | terpene | 21899 | 19 | 2 |
| 14 | NC_000964.3.region014 | antismash | NC_000964.3 | 4115741 | 4137440 | epipeptide | 21699 | 21 | 3 |
| 15 | NC_000964.3.region011 | antismash | NC_000964.3 | 3826057 | 3847669 | sactipeptide | 21612 | 19 | 4 |
| 16 | NC_000964.3.region010 | antismash | NC_000964.3 | 3593820 | 3614567 | CDPS | 20747 | 17 | 2 |
| 17 | NC_000964.3.region003 | antismash | NC_000964.3 | 1149957 | 1170476 | terpene | 20519 | 21 | 5 |
| 18 | NC_000964.3.region013 | antismash | NC_000964.3 | 4088149 | 4108419 | RRE-containing | 20270 | 18 | 4 |
| 19 | NC_000964.3.region007 | antismash | NC_000964.3 | 2259520 | 2279691 | glycocin | 20171 | 25 | 3 |
| 20 | NC_000964.3_cluster_6 | gecco | NC_000964.3 | 3278325 | 3293359 | NRP | 15035 | 9 | 2 |
| 21 | NC_000964.3_cluster_7 | gecco | NC_000964.3 | 3597289 | 3607121 | Unknown | 9833 | 8 | 2 |
| 22 | NC_000964.3_cluster_5 | gecco | NC_000964.3 | 2265668 | 2271657 | RiPP | 5990 | 6 | 2 |

Figure 1: image

Then on the second tab, the coverage of the focal sample's BGC-ome across the genomes in the target genomes database is shown:

## Important notes:

- Checking for BGC-Ome novelty is an exhaustive process and in the above example we used a database of representative genomes (dereplicated at 99% average nucleotide identity). Therefore we see that the B. subtilis st 168 BGC-Ome doesn't match any representative genome exactly; however, using a database of all Bacillus genomes present in GTDB release 214 (R214), we see that several Bacillus subtilis genomes are regarded as having all the BGCs predicted by antiSMASH & GECCO in strain 168. We provide comprehensive precompiled prepTG databases on Zenodo for the genera Bacillus, Streptomyces, and Micromonospora (featuring nearly all genomes belonging to these genera in GTDB R214) at: https://zenodo.org/records/10050207. To use these you would just download and uncompress, e.g. `wget https://zenodo.org/records/10050207/files/Micromonospora_prepTG_Database.tar.gz?download=1; gunzip -zxvf Micromonospora_prepTG_Database.tar.gz`.
- Default parameters for fai-based detection of BGCs are: 50% of BGC genes and 75% of key BGC genes (see above) need to be identified in whole or fragmented along scaffold edges via DIAMOND BLASTp at an E-value threshold of 1e-10. A syntenic similarity of 0.6 is also required. Note, there is a possibility that some BGCs might be highly paralogous and abon might not be able to resolve this super well - e.g. if your sample has two paralogous BGCs it might say they are both present in a target genome when only one is.



Target genome ID

Number of focal sample's BGCs found in target genome

BGC 1

BGC 2

BGC 3

o o o o

|  | target_genome | hared_ | 54.3.reg | 002 - Pr | 54.3.reg | 003 - Pr | 54.3.reg | 004 - Pr | 54.3.reg | 005 - Pr | 54.3.reg | 006 - Pr | 54.3.reg | 007 - Pr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | acillus_subtilis_GCA_009996815.1 | 18 | 98.61 | 0.952381 | 98.85556 | 0.857143 | 99.04565 | 1 | 98.55278 | 0.972973 | 99.1 | 1 | 92 | 0.8 |
| 3 | acillus_subtilis_GCA_905315385.1 | 18 | 98.355 | 0.952381 | 98.7 | 1 | 99.31556 | 0.978261 | 98.98889 | 0.972973 | 99.35263 | 1 | 95.52 | 0.8 |
| 4 | acillus_subtilis_GCA_000245295.1 | 18 | 99.89762 | 1 | 99.50952 | 1 | 99.91778 | 0.978261 | 99.78108 | 1 | 99.61579 | 1 | NA | NA |
| 5 | acillus_subtilis_GCA_021728955.1 | 17 | 98.55714 | 1 | 99.23333 | 1 | 99.12826 | 1 | 98.90571 | 0.945946 | 99.49474 | 1 | NA | NA |
| 6 | acillus_subtilis_GCA_023714485.1 | 17 | 99.30952 | 1 | 99.40952 | 1 | 99.08696 | 1 | 98.58649 | 1 | 99.45789 | 1 | NA | NA |
| 7 | acillus_subtilis_GCA_010578165.1 | 17 | 98.78537 | 0.97619 | 98.98571 | 1 | 98.95435 | 1 | 98.56944 | 0.972973 | 99.03684 | 1 | NA | NA |
| 8 | acillus_subtilis_GCA_024507835.1 | 17 | 98.6375 | 0.952381 | 99.1619 | 1 | 99.26667 | 0.978261 | 99.025 | 0.972973 | 99.35789 | 1 | NA | NA |
| 9 | acillus_subtilis_GCA_003600785.1 | 17 | 98.66341 | 0.97619 | 98.92381 | 1 | 98.93333 | 0.978261 | 98.44722 | 0.972973 | 99.09474 | 1 | NA | NA |
| 0 | acillus_subtilis_GCA_004328925.1 | 17 | 98.61026 | 0.928571 | 99.04762 | 1 | 99.43111 | 0.978261 | 98.78056 | 0.972973 | 99.37895 | 1 | NA | NA |
| 1 | acillus_subtilis_GCA_009913275.1 | 17 | 98.555 | 0.952381 | 98.06667 | 1 | 98.97727 | 0.956522 | 98.23143 | 0.945946 | 99.16842 | 1 | 91.39333 | 0.6 |
| 2 | acillus_subtilis_GCA_003265285.1 | 17 | 98.49512 | 0.97619 | 98.32381 | 1 | 98.66222 | 0.978261 | 98.84722 | 0.972973 | 98.87778 | 0.947368 | NA | NA |
| 3 | acillus_subtilis_GCA_019334685.1 | 17 | 98.575 | 0.952381 | 99.01429 | 1 | 98.96304 | 1 | 98.39189 | 1 | 98.88421 | 1 | NA | NA |
| 4 | acillus_subtilis_G_GCA_016757475.1 | 17 | 92.81579 | 0.904762 | 90.83333 | 1 | 92.61364 | 0.956522 | 92.41389 | 0.972973 | 95.27143 | 0.736842 | NA | NA |
| 5 | acillus_subtilis_G_GCA_008180625.1 | 17 | 92.92632 | 0.904762 | 91.07143 | 1 | 91.0375 | 0.695652 | 92.29697 | 0.891892 | 95.45714 | 0.736842 | NA | NA |
| 6 | acillus_subtilis_GCA_003330965.1 | 17 | 98.37561 | 0.97619 | 99.07619 | 1 | 99.30667 | 0.978261 | 98.61944 | 0.972973 | 99.1 | 1 | NA | NA |
| 7 | acillus_subtilis_GCA_000691185.1 | 16 | 98.2 | 0.97619 | 99.1619 | 1 | 99.1 | 1 | 99.1 | 0.972973 | NA | NA | NA | NA |
| 8 | acillus_subtilis_GCA_905315375.1 | 16 | 98.75 | 0.952381 | 99.10476 | 1 | 99.07391 | 1 | 99.02778 | 0.972973 | 99.36316 | 1 | NA | NA |
| 9 | acillus_subtilis_GCA_007829515.1 | 16 | 98.6025 | 0.952381 | 98.9 | 1 | 99.15217 | 1 | 98.48378 | 1 | 98.96842 | 1 | NA | NA |
| 0 | acillus_subtilis_GCA_023714065.1 | 16 | 98.20244 | 0.97619 | 99.17619 | 1 | 98.71957 | 1 | 98.68286 | 0.945946 | 99.07368 | 1 | NA | NA |

Average amino-acid identity of target genome matching BGC to the sample's BGC-1

Genes from sample's BGC-1 found co-located in target genome

Figure 2: image

## Usage

```
usage: abon [-h] [-a ANTISMASH_RESULTS] [-g GECCO_RESULTS] -tg TARGET_GENOMES_DB
            [-fo FAI_OPTIONS] [-s] [-si SIMPLE_BLASTP_IDENTITY_CUTOFF]
            [-sc SIMPLE_BLASTP_COVERAGE_CUTOFF] [-se SIMPLE_BLASTP_EVALUE_CUTOFF]
            [-sk SIMPLE_BLASTP_KEY_PROTEINS_PROPORTION_CUTOFF]
            [-sm SIMPLE_BLASTP_SENSITIVITY_MODE] -o OUTDIR [-c CPUS]

    Program: abon
    Author: Rauf Salamzade
    Affiliation: Kalan Lab, UW Madison, Department of Medical Microbiology and Immunology

    abon - Assess Bgc-Ome Novelty

    abon wraps fai to assess the novelty of a sample’s BGC-ome relative to a set of
    target genomes. Alternatively, it can run a simple DIAMOND BLASTp analysis to
    just assess the presence of BGC genes individually - without the requirement
    they are co-located like in the focal sample’s BGCs.


options:
  -h, --help            show this help message and exit
  -a ANTISMASH_RESULTS, --antismash_results ANTISMASH_RESULTS
                        Path to antiSMASH BGC prediction results directory for a
                        single sample/genome.
  -g GECCO_RESULTS, --gecco_results GECCO_RESULTS
                        Path to GECCO BGC prediction results directory for a single
                        sample/genome.
  -tg TARGET_GENOMES_DB, --target_genomes_db TARGET_GENOMES_DB
                        prepTG database directory for target genomes of interest.
  -fo FAI_OPTIONS, --fai_options FAI_OPTIONS
                        Provide fai options to run. Should be surrounded by quotes.
                        [Default is "-kpm 0.75 -kpe 1e-10 -e 1e-10 -m 0.5 -dm -sct 0.6"]
  -s, --use_simple_blastp
                        Use a simple DIAMOND BLASTp search with no requirement
                        for co-localization of hits.
  -si SIMPLE_BLASTP_IDENTITY_CUTOFF, --simple_blastp_identity_cutoff
                                      SIMPLE_BLASTP_IDENTITY_CUTOFF
                        If simple BLASTp mode requested : cutoff for identity
                        between query proteins and matches in target genomes
                        [Default is 40.0].
  -sc SIMPLE_BLASTP_COVERAGE_CUTOFF, --simple_blastp_coverage_cutoff
                                      SIMPLE_BLASTP_COVERAGE_CUTOFF
                        If simple BLASTp mode requested : cutoff for coverage
                        between query proteins and matches in target genomes
                        [Default is 70.0].
  -se SIMPLE_BLASTP_EVALUE_CUTOFF, --simple_blastp_evalue_cutoff
                                    SIMPLE_BLASTP_EVALUE_CUTOFF
                        If simple BLASTp mode requested : cutoff for E-value
                        between query proteins and matches in target genomes
                        [Default is 1e-10].
  -sk SIMPLE_BLASTP_KEY_PROTEINS_PROPORTION_CUTOFF,
                                  --simple_blastp_key_proteins_proportion_cutoff
                                    SIMPLE_BLASTP_KEY_PROTEINS_PROPORTION_CUTOFF
                        If simple BLASTp mode requested : cutoff for proportion
                        of key proteins needed to consider a BGC as present in a
                        target genome [Default is 0.75].
  -sm SIMPLE_BLASTP_SENSITIVITY_MODE, --simple_blastp_sensitivity_mode
                                        SIMPLE_BLASTP_SENSITIVITY_MODE
                        Sensitivity mode for DIAMOND BLASTp. [Default is
                        "very-sensititve"].
  -o OUTDIR, --outdir OUTDIR
                        Output directory.
  -c CPUS, --cpus CPUS  The number of CPUs to use.
```

## 9.2 more info on atpoc

# atpoc - Assess Temperate Phage-Ome Conservation

**atpoc** takes as input VIBRANT, PhiSpy, and/or geNomad results directories (with prophage predictions) for a single sample together with a prepTG (target genomes) database to determine how conserved the sample's Phage-ome is across the genomes in the database. This could be insightful as to when a temperate phage might have integrated into a species genome and whether certain prophages are unique to certain strains.

The specific cutoffs used in fai for gene cluster detection in target genomes can be adapted as needed. Alternatively, a simple BLASTp search can be performed instead to determine all homologs of proteins for each BGC from the focal sample in target genomes regardless of whether they are similarly co-located or not. Default parameters for fai-based detection of phages are: 50% of phage genes need to be identified in whole or fragmented along scaffold edges via DIAMOND BLASTp at an E-value threshold of 1e-10. A syntenic similarity of 0.4 is also required. Note, there is a possibility that some phages might be highly paralogous and atpoc might not be able to resolve this super well - e.g. if your sample has two paralogous phages it might say they are both present in a target genome when only one is.

> If **fai** is used for searching (the default), check out the individual fai results (in the subdirectory `fai_or_blast_Results/`) for each phage to see details on the conservation of individual genes. Further, follow up analysis can be performed using **zol** per phage to summarize the conservation of distinct ortholog groups, evolutionary stats, and functional info.

By default, prodigal-gv will be used for gene calling but you can use pyrodigal (with models for gene calling in bacteria) via the `--use_pyrodigal`. This might be more appropriate if gene calling for the target genomes was performed with default pyrodigal/prodigal instead of prodigal-gv via **prepTG**. ‘ > We also recommend checking out PHANOTATE and Pharokka for detailed annotation of phages or obtaining better gene calls and performing more manual fai & zol analysis.

### Conservation of *Streptococcus pyogenes* M1_GAS temperate phages across the *Streptococcus* genus

The following is a mini-tutorial on using atpoc to investigate the novelty of the Phage-ome of *Streptococcus pyogenes* st. M1_GAS to representative Streptococcus genomes we made available in a precompiled prepTG database. The focal *Streptococcus pyogenes* genome is the same one used as an example by PhiSpy.

First, lets download the query genome of interest from PhiSpy's git repo and also format it to FASTA format (for VIBRANT/geNomad):

```
# Download genome from NCBI
wget https://raw.githubusercontent.com/linsalrob/PhiSpy/master/tests/Streptococcus_pyogenes_M1_GAS.gb

# reformat to fasta (using script available in zol)
genbankToFasta.py Streptococcus_pyogenes_M1_GAS.gb > Streptococcus_pyogenes_M1_GAS.fna
```

Next, we can run PhiSpy, VIBRANT, and geNomade to identify phages in the focal genome:

```
# in some conda environment or setting with PhiSpy available
PhiSpy.py Streptococcus_pyogenes_M1_GAS.gb -o PhiSpy_Results/

# in some conda environment or setting with VIBRANT available
```

```
VIBRANT_run.py -i Streptococcus_pyogenes_M1_GAS.fna -folder VIBRANT_Results/

# in some conda environment or setting with geNomad available
genomad end-to-end Streptococcus_pyogenes_M1_GAS.fna geNomad_Results/ /path/to/genomad_dbs/
```

Next, we can setup the precompiled database of *Streptococcus* representative genome using prepTG:

```
# in zol's conda environment or via the Docker wrapper:
prepTG -d Streptococcus -o Streptococcus_Reps_prepTG_Database/
```

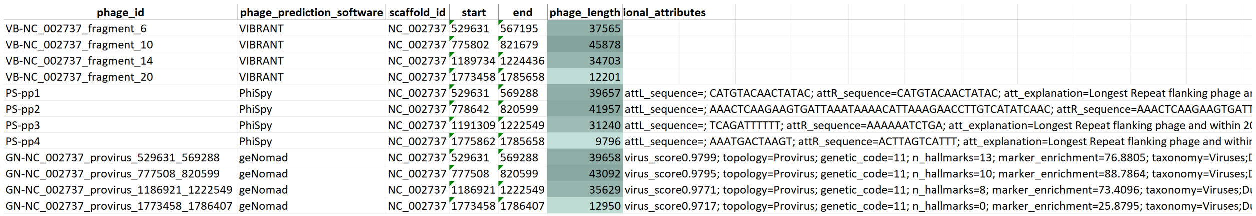
Now we are ready to run atpoc!

```
atpoc -i Streptococcus_pyogenes_M1_GAS.fna -tg Streptococcus_Reps_prepTG_Database/ \
      -ps PhiSpy_Results/ -vi VIBRANT_Results/ -gn geNomad_Results/ \
      -o atpoc_Results/ -c 20
```

Note, this can take a while as it will involve running fai X times (where X is the number of phage predictions across all methods in the focal sample of interest).

### The result!

Similar to fai and zol's major results, atpoc also primarily produces an XLSX spreadsheet. On the first tab of atpoc's resulting XLSX spreadsheet, is an overview of the focal sample's prophage predictions from the different software:



| phage_id | phage_prediction_software | scaffold_id | start | end | phage_length | ional_attributes |
|---|---|---|---|---|---|---|
| VB-NC_002737_fragment_6 | VIBRANT | NC_002737 | 529631 | 567195 | 37565 | |
| VB-NC_002737_fragment_10 | VIBRANT | NC_002737 | 775802 | 821679 | 45878 | |
| VB-NC_002737_fragment_14 | VIBRANT | NC_002737 | 1189734 | 1224436 | 34703 | |
| VB-NC_002737_fragment_20 | VIBRANT | NC_002737 | 1773458 | 1785658 | 12201 | |
| PS-pp1 | PhiSpy | NC_002737 | 529631 | 569288 | 39657 | attL_sequence=; CATGTACAACTATAC; attR_sequence=CATGTACAACTATAC; att_explanation=Longest Repeat flanking phage a |
| PS-pp2 | PhiSpy | NC_002737 | 778642 | 820599 | 41957 | attL_sequence=; AAACTCAAGAAGTGATTAAATAAAACATTAAAGAACCTTGTCATATCAAC; attR_sequence=AAACTCAAGAAGTGATT |
| PS-pp3 | PhiSpy | NC_002737 | 1191309 | 1222549 | 31240 | attL_sequence=; TCAGATTTTTT; attR_sequence=AAAAAATCTGA; att_explanation=Longest Repeat flanking phage and within 2 |
| PS-pp4 | PhiSpy | NC_002737 | 1775862 | 1785658 | 9796 | attL_sequence=; AAATGACTAAGT; attR_sequence=ACTTAGTCATTT; att_explanation=Longest Repeat flanking phage and withi |
| GN-NC_002737_provirus_529631_569288 | geNomad | NC_002737 | 529631 | 569288 | 39658 | virus_score0.9799; topology=Provirus; genetic_code=11; n_hallmarks=13; marker_enrichment=76.8805; taxonomy=Viruses; |
| GN-NC_002737_provirus_777508_820599 | geNomad | NC_002737 | 777508 | 820599 | 43092 | virus_score0.9795; topology=Provirus; genetic_code=11; n_hallmarks=10; marker_enrichment=88.7864; taxonomy=Viruses; |
| GN-NC_002737_provirus_1186921_1222549 | geNomad | NC_002737 | 1186921 | 1222549 | 35629 | virus_score0.9771; topology=Provirus; genetic_code=11; n_hallmarks=8; marker_enrichment=73.4096; taxonomy=Viruses;D |
| GN-NC_002737_provirus_1773458_1786407 | geNomad | NC_002737 | 1773458 | 1786407 | 12950 | virus_score0.9717; topology=Provirus; genetic_code=11; n_hallmarks=0; marker_enrichment=25.8795; taxonomy=Viruses;D |

Figure 1: image

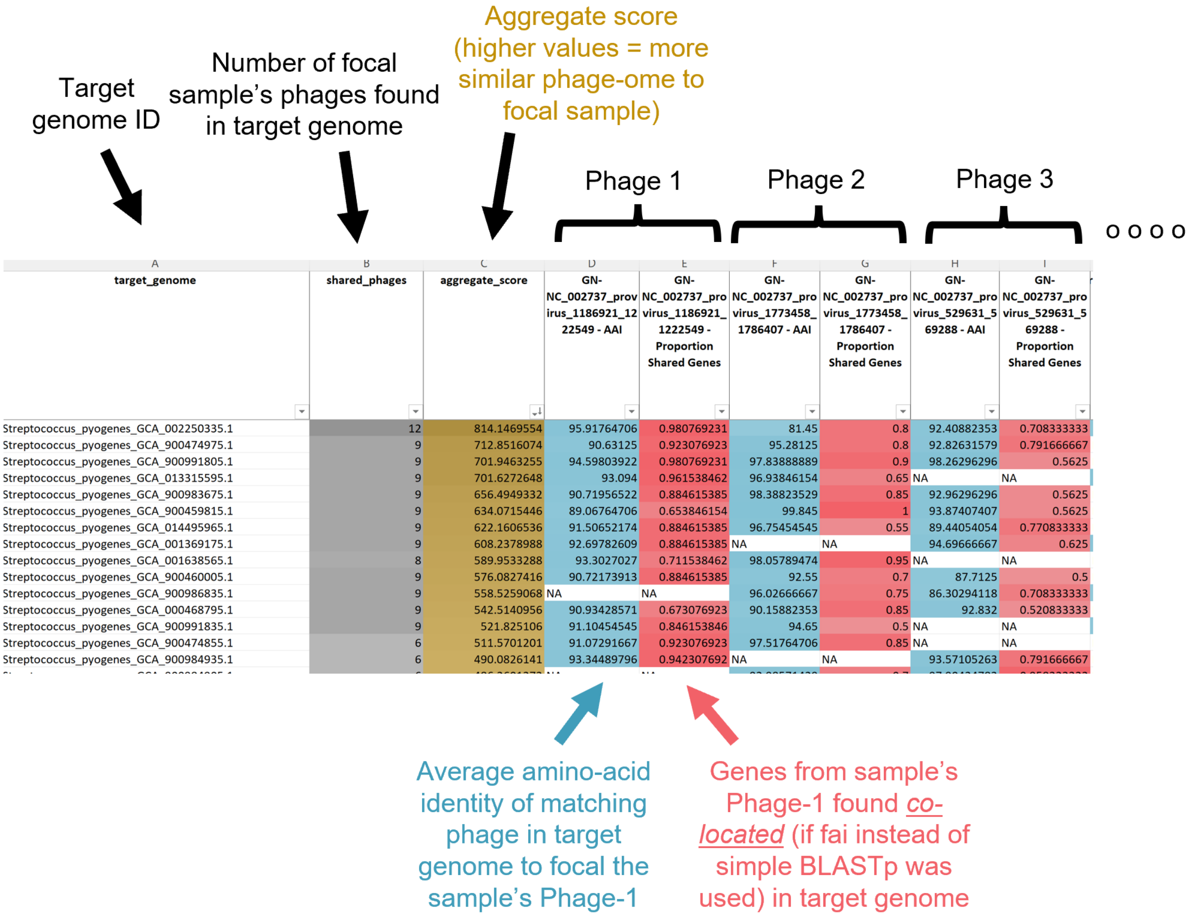
Then on the second tab, the coverage of the focal sample's phage-ome across the genomes in the target genomes database is shown:

## Usage

```
usage: atpoc [-h] -i SAMPLE_GENOME [-vi VIBRANT_RESULTS] [-ps PHISPY_RESULTS]
             [-gn GENOMAD_RESULTS] -tg TARGET_GENOMES_DB [-up] [-fo FAI_OPTIONS]
             [-s] [-si SIMPLE_BLASTP_IDENTITY_CUTOFF] [-sc SIMPLE_BLASTP_COVERAGE_CUTOFF]
             [-se SIMPLE_BLASTP_EVALUE_CUTOFF] [-sm SIMPLE_BLASTP_SENSITIVITY_MODE]
             -o OUTDIR [-c CPUS]

    Program: atpoc
    Author: Rauf Salamzade
    Affiliation: Kalan Lab, UW Madison, Department of Medical Microbiology and Immunology

    atpoc - Assess Temperate Phage-Ome Conservation
```



Target genome ID

Number of focal sample's phages found in target genome

Aggregate score (higher values = more similar phage-ome to focal sample)

| A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|
| target_genome | shared_phages | aggregate_score | GN-NC_002737_provirus_1186921_1222549 - AAI | GN-NC_002737_provirus_1186921_1222549 - Proportion Shared Genes | GN-NC_002737_provirus_1773458_1786407 - AAI | GN-NC_002737_provirus_1773458_1786407 - Proportion Shared Genes | GN-NC_002737_provirus_529631_569288 - AAI | GN-NC_002737_provirus_529631_569288 - Proportion Shared Genes |
| Streptococcus_pyogenes_GCA_002250335.1 | 12 | 814.1469554 | 95.91764706 | 0.980769231 | 81.45 | 0.8 | 92.40882353 | 0.708333333 |
| Streptococcus_pyogenes_GCA_900474975.1 | 9 | 712.8516074 | 90.63125 | 0.923076923 | 95.28125 | 0.8 | 92.82631579 | 0.791666667 |
| Streptococcus_pyogenes_GCA_900991805.1 | 9 | 701.9463255 | 94.59803922 | 0.980769231 | 97.83888889 | 0.9 | 98.26296296 | 0.5625 |
| Streptococcus_pyogenes_GCA_013315595.1 | 9 | 701.6272648 | 93.094 | 0.961538462 | 96.93846154 | 0.65 | NA | NA |
| Streptococcus_pyogenes_GCA_900983675.1 | 9 | 656.4949332 | 90.71956522 | 0.884615385 | 98.38823529 | 0.85 | 92.96296296 | 0.5625 |
| Streptococcus_pyogenes_GCA_900459815.1 | 9 | 634.0715446 | 89.06764706 | 0.653846154 | 99.845 | 1 | 93.87407407 | 0.5625 |
| Streptococcus_pyogenes_GCA_014495965.1 | 9 | 622.1606536 | 91.50652174 | 0.884615385 | 96.75454545 | 0.55 | 89.44054054 | 0.770833333 |
| Streptococcus_pyogenes_GCA_001369175.1 | 9 | 608.2378988 | 92.69782609 | 0.884615385 | NA | NA | 94.69666667 | 0.625 |
| Streptococcus_pyogenes_GCA_001638565.1 | 8 | 589.9533288 | 93.3027027 | 0.711538462 | 98.05789474 | 0.95 | NA | NA |
| Streptococcus_pyogenes_GCA_900460005.1 | 9 | 576.0827416 | 90.72173913 | 0.884615385 | 92.55 | 0.7 | 87.7125 | 0.5 |
| Streptococcus_pyogenes_GCA_900986835.1 | 9 | 558.5259068 | NA | NA | 96.02666667 | 0.75 | 86.30294118 | 0.708333333 |
| Streptococcus_pyogenes_GCA_000468795.1 | 9 | 542.5140956 | 90.93428571 | 0.673076923 | 90.15882353 | 0.85 | 92.832 | 0.520833333 |
| Streptococcus_pyogenes_GCA_900991835.1 | 9 | 521.825106 | 91.10454545 | 0.846153846 | 94.65 | 0.5 | NA | NA |
| Streptococcus_pyogenes_GCA_900474855.1 | 6 | 511.5701201 | 91.07291667 | 0.923076923 | 97.51764706 | 0.85 | NA | NA |
| Streptococcus_pyogenes_GCA_900984935.1 | 6 | 490.0826141 | 93.34489796 | 0.942307692 | NA | NA | 93.57105263 | 0.791666667 |

Average amino-acid identity of matching phage in target genome to focal the sample's Phage-1

Genes from sample's Phage-1 found *co-located* (if fai instead of simple BLASTp was used) in target genome

Figure 2: image

```
    atpoc wraps fai to assess the conservation of a sample’s integrated/temperate
    phage-ome relative to a set of target genomes (e.g. genomes belonging to the
    same genus). Alternatively, it can run a simple DIAMOND BLASTp analysis to
    just assess the presence of prophage genes individually - without the
    requirement they are co-located like in the focal sample.

options:
  -h, --help            show this help message and exit
  -i SAMPLE_GENOME, --sample_genome SAMPLE_GENOME
                        Path to sample genome in GenBank or FASTA format.
  -vi VIBRANT_RESULTS, --vibrant_results VIBRANT_RESULTS
                        Path to VIBRANT results directory for a single sample/genome.
  -ps PHISPY_RESULTS, --phispy_results PHISPY_RESULTS
                        Path to PhiSpy results directory for a single sample/genome.
  -gn GENOMAD_RESULTS, --genomad_results GENOMAD_RESULTS
                        Path to GeNomad results directory for a single sample/genome.
  -tg TARGET_GENOMES_DB, --target_genomes_db TARGET_GENOMES_DB
                        prepTG database directory for target genomes of interest.
  -up, --use_pyrodigal  Use default pyrodigal instead of prodigal-gv to call genes in
                        phage regions to use as queries in fai/simple-blast. This
                        is perhaps preferable if target genomes db was created with
                        default pyrodigal/prodigal.
  -fo FAI_OPTIONS, --fai_options FAI_OPTIONS
                        Provide fai options to run. Should be surrounded by quotes.
                        [Default is "-e 1e-10 -m 0.5 -dm -sct 0.4"]
  -s, --use_simple_blastp
                        Use a simple DIAMOND BLASTp search with no requirement for
                        co-localization of hits.
  -si SIMPLE_BLASTP_IDENTITY_CUTOFF, --simple_blastp_identity_cutoff
                                       SIMPLE_BLASTP_IDENTITY_CUTOFF
                        If simple BLASTp mode requested : cutoff for identity
                        between query proteins and matches in target genomes
                        [Default is 40.0].
  -sc SIMPLE_BLASTP_COVERAGE_CUTOFF, --simple_blastp_coverage_cutoff
                                       SIMPLE_BLASTP_COVERAGE_CUTOFF
                        If simple BLASTp mode requested : cutoff for coverage
                        between query proteins and matches in target genomes
                        [Default is 70.0].
  -se SIMPLE_BLASTP_EVALUE_CUTOFF, --simple_blastp_evalue_cutoff
                                     SIMPLE_BLASTP_EVALUE_CUTOFF
                        If simple BLASTp mode requested : cutoff for E-value
                        between query proteins and matches in target genomes
                        [Default is 1e-10].
  -sm SIMPLE_BLASTP_SENSITIVITY_MODE, --simple_blastp_sensitivity_mode
                                        SIMPLE_BLASTP_SENSITIVITY_MODE
                        Sensitivity mode for DIAMOND BLASTp. [Default is
                        "very-sensititve"].
  -o OUTDIR, --outdir OUTDIR
                        Output directory.
  -c CPUS, --cpus CPUS  The number of CPUs to use.
```

# 9.3 more info on apos

## apos - Assess Temperate Plasmid-Ome Conservation

**apos** takes as input MOB-suite and/or geNomad results directories (with plasmid predictions) for a single sample together with a prepTG (target genomes) database to determine how conserved the sample's plasmid-ome is across the genomes in the database. This could give insight into the conservation of specific plasmids in the focal sample's genome across its species or genus.

The specific cutoffs used in fai for gene cluster detection in target genomes can be adapted as needed. Alternatively, a simple BLASTp search can be performed instead to determine all homologs of proteins for each BGC from the focal sample in target genomes regardless of whether they are similarly co-located or not. Default parameters for fai-based detection of plasmids are: 50% of plasmid genes need to be identified in whole or fragmented along scaffold edges via DIAMOND BLASTp at an E-value threshold of 1e-10. Because plasmids are highly dynamic, the syntenic similarity requirement is turned off.

**Because plasmids are highly dynamic - we recommend using the simple BLASTp search mode instead of the default of fai. This is because fai will require genes to be co-located and plasmid parts can be exchanged with other plasmids and the chromosome. Simple BLASTp searching can be requested with the `-s` argument.**

> If **fai** is used for searching (the default), check out the individual fai results (in the subdirectory `fai_or_blast_Results/`) for each plasmid to see details on the conservation of individual genes. Further, follow up analysis can be performed using **zol** per plasmid to summarize the conservation of distinct ortholog groups, evolutionary stats, and functional info.

### Conservation of *Enterococcus faecalis* V583 plasmids across the *Enterococcus* genus

The following is a mini-tutorial on using apos to investigate the novelty of the plasmid-ome of *Enterococcus faecalis* st. V583 to representative *Enterococcus* genomes we made available in a precompiled prepTG database.

First, lets download the query genome of interest:

```
# Download genome from NCBI
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/007/785/GCF_000007785.1_ASM778v1/GCF_000007785.1_A

# Uncompress it & rename it
gunzip GCF_000007785.1_ASM778v1_genomic.fna.gz
mv GCF_000007785.1_ASM778v1_genomic.fna Enterococcus_faecalis_V583.fna
```

Next, we can run MOB-suite and geNomade to identify plasmids in the focal genome:

```
# in some conda environment or setting with MOB-suite available
mob_recon  --infile Enterococcus_faecalis_V583.fna --outdir MOBsuite_Results/

# in some conda environment or setting with geNomad available
genomad end-to-end Enterococcus_faecalis_V583.fna geNomad_Results/ /path/to/genomad_dbs/
```

Next, we can setup the precompiled database of *Enterococcus* representative genome using prepTG:

```
# in zol's conda environment or via the Docker wrapper:
prepTG -d Enterococcus -o Enterococcus_Reps_prepTG_Database/
```

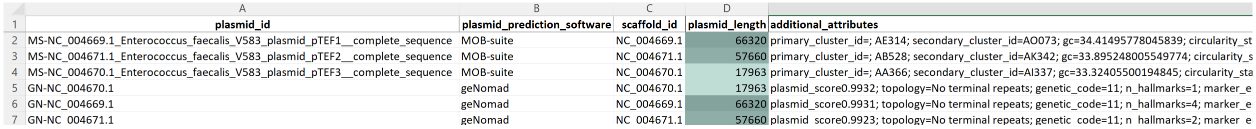
Now we are ready to run apos!

```
# Note, as per our recommendation above, we run apos with the simple blast search method via the -s arg
apos -i Enterococcus_faecalis_V583.fna -tg Enterococcus_Reps_prepTG_Database/ -ns MOBsuite_Results/ -gn
```

Note, this can take a while as it will involve running fai X times (where X is the number of plasmid predictions across all methods in the focal sample of interest).

### The result!

Similar to fai and zol's major results, apos also primarily produces an XLSX spreadsheet. On the first tab of apos's resulting XLSX spreadsheet, is an overview of the focal sample's plasmid predictions from the different software:



| | A | B | C | D | |
|---|---|---|---|---|---|
| 1 | plasmid_id | plasmid_prediction_software | scaffold_id | plasmid_length | additional_attributes |
| 2 | MS-NC_004669.1_Enterococcus_faecalis_V583_plasmid_pTEF1__complete_sequence | MOB-suite | NC_004669.1 | 66320 | primary_cluster_id=; AE314; secondary_cluster_id=AO073; gc=34.41495778045839; circularity_st |
| 3 | MS-NC_004671.1_Enterococcus_faecalis_V583_plasmid_pTEF2__complete_sequence | MOB-suite | NC_004671.1 | 57660 | primary_cluster_id=; AB528; secondary_cluster_id=AK342; gc=33.895248005549774; circularity_s |
| 4 | MS-NC_004670.1_Enterococcus_faecalis_V583_plasmid_pTEF3__complete_sequence | MOB-suite | NC_004670.1 | 17963 | primary_cluster_id=; AA366; secondary_cluster_id=AI337; gc=33.32405500194845; circularity_sta |
| 5 | GN-NC_004670.1 | geNomad | NC_004670.1 | 17963 | plasmid_score0.9932; topology=No terminal repeats; genetic_code=11; n_hallmarks=1; marker_e |
| 6 | GN-NC_004669.1 | geNomad | NC_004669.1 | 66320 | plasmid_score0.9931; topology=No terminal repeats; genetic_code=11; n_hallmarks=4; marker_e |
| 7 | GN-NC_004671.1 | geNomad | NC_004671.1 | 57660 | plasmid_score0.9923; topology=No terminal repeats; genetic_code=11; n_hallmarks=2; marker_e |

Figure 1: image

Then on the second tab, the coverage of the focal sample's plasmid-ome across the genomes in the target genomes database is shown:

## Usage

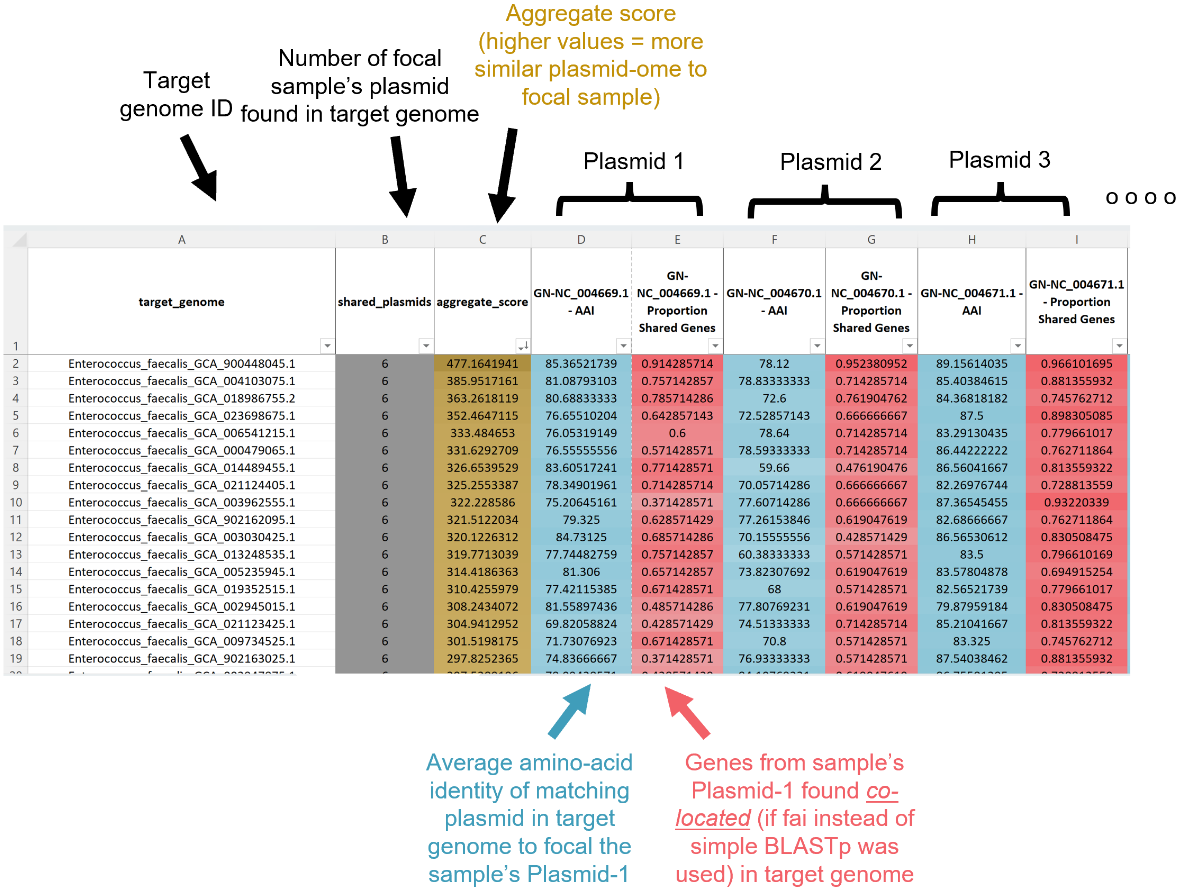
```
usage: apos [-h] -i SAMPLE_GENOME [-ms MOBSUITE_RESULTS] [-gn GENOMAD_RESULTS]
            -tg TARGET_GENOMES_DB [-up] [-fo FAI_OPTIONS] [-s]
            [-si SIMPLE_BLASTP_IDENTITY_CUTOFF] [-sc SIMPLE_BLASTP_COVERAGE_CUTOFF]
            [-se SIMPLE_BLASTP_EVALUE_CUTOFF] [-sm SIMPLE_BLASTP_SENSITIVITY_MODE]
            -o OUTDIR [-c CPUS]

        Program: apos
        Author: Rauf Salamzade
        Affiliation: Kalan Lab, UW Madison, Department of Medical Microbiology and Immunology

        apos - Assess Plasmid-Ome Similarity

        apos wraps fai to assess the conservation of a sample's plasmid-ome
        relative to a set of target genomes (e.g. genomes belonging to the same genus).
        Alternatively, it can run a simple DIAMOND BLASTp analysis to just assess
        the presence of plasmid genes individually - without the requirement they
        are co-located in one scaffold like in the focal sample.


options:
  -h, --help            show this help message and exit
  -i SAMPLE_GENOME, --sample_genome SAMPLE_GENOME
                        Path to sample genome in GenBank or FASTA format.
```



Aggregate score (higher values = more similar plasmid-ome to focal sample)

Target genome ID

Number of focal sample's plasmid found in target genome

Plasmid 1

Plasmid 2

Plasmid 3

o o o o

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | target_genome | shared_plasmids | aggregate_score | GN-NC_004669.1 - AAI | GN-NC_004669.1 - Proportion Shared Genes | GN-NC_004670.1 - AAI | GN-NC_004670.1 - Proportion Shared Genes | GN-NC_004671.1 - AAI | GN-NC_004671.1 - Proportion Shared Genes |
| 2 | Enterococcus_faecalis_GCA_900448045.1 | 6 | 477.1641941 | 85.36521739 | 0.914285714 | 78.12 | 0.952380952 | 89.15614035 | 0.966101695 |
| 3 | Enterococcus_faecalis_GCA_004103075.1 | 6 | 385.9517161 | 81.08793103 | 0.757142857 | 78.83333333 | 0.714285714 | 85.40384615 | 0.881355932 |
| 4 | Enterococcus_faecalis_GCA_018986755.2 | 6 | 363.2618119 | 80.68833333 | 0.785714286 | 72.6 | 0.761904762 | 84.36818182 | 0.745762712 |
| 5 | Enterococcus_faecalis_GCA_023698675.1 | 6 | 352.4647115 | 76.65510204 | 0.642857143 | 72.52857143 | 0.666666667 | 87.5 | 0.898305085 |
| 6 | Enterococcus_faecalis_GCA_006541215.1 | 6 | 333.484653 | 76.05319149 | 0.6 | 78.64 | 0.714285714 | 83.29130435 | 0.779661017 |
| 7 | Enterococcus_faecalis_GCA_000479065.1 | 6 | 331.6292709 | 76.55555556 | 0.571428571 | 78.59333333 | 0.714285714 | 86.44222222 | 0.762711864 |
| 8 | Enterococcus_faecalis_GCA_014489455.1 | 6 | 326.6539529 | 83.60517241 | 0.771428571 | 59.66 | 0.476190476 | 86.56041667 | 0.813559322 |
| 9 | Enterococcus_faecalis_GCA_021124405.1 | 6 | 325.2553387 | 78.34901961 | 0.714285714 | 70.05714286 | 0.666666667 | 82.26976744 | 0.728813559 |
| 10 | Enterococcus_faecalis_GCA_003962555.1 | 6 | 322.228586 | 75.20645161 | 0.371428571 | 77.60714286 | 0.666666667 | 87.36545455 | 0.93220339 |
| 11 | Enterococcus_faecalis_GCA_902162095.1 | 6 | 321.5122034 | 79.325 | 0.628571429 | 77.26153846 | 0.619047619 | 82.68666667 | 0.762711864 |
| 12 | Enterococcus_faecalis_GCA_003030425.1 | 6 | 320.1226312 | 84.73125 | 0.685714286 | 70.15555556 | 0.428571429 | 86.56530612 | 0.830508475 |
| 13 | Enterococcus_faecalis_GCA_013248535.1 | 6 | 319.7713039 | 77.74482759 | 0.757142857 | 60.38333333 | 0.571428571 | 83.5 | 0.796610169 |
| 14 | Enterococcus_faecalis_GCA_005235945.1 | 6 | 314.4186363 | 81.306 | 0.657142857 | 73.82307692 | 0.619047619 | 83.57804878 | 0.694915254 |
| 15 | Enterococcus_faecalis_GCA_019352515.1 | 6 | 310.4255979 | 77.42115385 | 0.671428571 | 68 | 0.571428571 | 82.56521739 | 0.779661017 |
| 16 | Enterococcus_faecalis_GCA_002945015.1 | 6 | 308.2434072 | 81.55897436 | 0.485714286 | 77.80769231 | 0.619047619 | 79.87959184 | 0.830508475 |
| 17 | Enterococcus_faecalis_GCA_021123425.1 | 6 | 304.9412952 | 69.82058824 | 0.428571429 | 74.51333333 | 0.714285714 | 85.21041667 | 0.813559322 |
| 18 | Enterococcus_faecalis_GCA_009734525.1 | 6 | 301.5198175 | 71.73076923 | 0.671428571 | 70.8 | 0.571428571 | 83.325 | 0.745762712 |
| 19 | Enterococcus_faecalis_GCA_902163025.1 | 6 | 297.8252365 | 74.83666667 | 0.371428571 | 76.93333333 | 0.571428571 | 87.54038462 | 0.881355932 |

Average amino-acid identity of matching plasmid in target genome to focal the sample's Plasmid-1

Genes from sample's Plasmid-1 found ***co-located*** (if fai instead of simple BLASTp was used) in target genome

Figure 2: image

```
-ms MOBSUITE_RESULTS, --mobsuite_results MOBSUITE_RESULTS
                      Path to MOB-suite (mob_recon) results directory for a single sample/genome.
-gn GENOMAD_RESULTS, --genomad_results GENOMAD_RESULTS
                      Path to GeNomad results directory for a single sample/genome.
-tg TARGET_GENOMES_DB, --target_genomes_db TARGET_GENOMES_DB
                      prepTG database directory for target genomes of interest.
-fo FAI_OPTIONS, --fai_options FAI_OPTIONS
                      Provide fai options to run. Should be surrounded by quotes. [Default is "-e 1e-
-s, --use_simple_blastp
                      Use a simple DIAMOND BLASTp search with no requirement for
                      co-localization of hits.
-si SIMPLE_BLASTP_IDENTITY_CUTOFF, --simple_blastp_identity_cutoff
                                     SIMPLE_BLASTP_IDENTITY_CUTOFF
                      If simple BLASTp mode requested : cutoff for identity
                      between query proteins and matches in target genomes
                      [Default is 40.0].
-sc SIMPLE_BLASTP_COVERAGE_CUTOFF, --simple_blastp_coverage_cutoff
                                     SIMPLE_BLASTP_COVERAGE_CUTOFF
                      If simple BLASTp mode requested : cutoff for coverage
                      between query proteins and matches in target genomes
                      [Default is 70.0].
-se SIMPLE_BLASTP_EVALUE_CUTOFF, --simple_blastp_evalue_cutoff
                                   SIMPLE_BLASTP_EVALUE_CUTOFF
                      If simple BLASTp mode requested : cutoff for E-value
                      between query proteins and matches in target genomes
                      [Default is 1e-10].
-sm SIMPLE_BLASTP_SENSITIVITY_MODE, --simple_blastp_sensitivity_mode
                                      SIMPLE_BLASTP_SENSITIVITY_MODE
                      Sensitivity mode for DIAMOND BLASTp. [Default is
                      "very-sensititve"].
-o OUTDIR, --outdir OUTDIR
                      Output directory.
-c CPUS, --cpus CPUS  The number of CPUs to use.
```