Hands-On Metagenomics

Adriana Rego, Nicola Gambardella

ciimar

BIOPORTUGAL S.A.
Químico, Farmacêutica

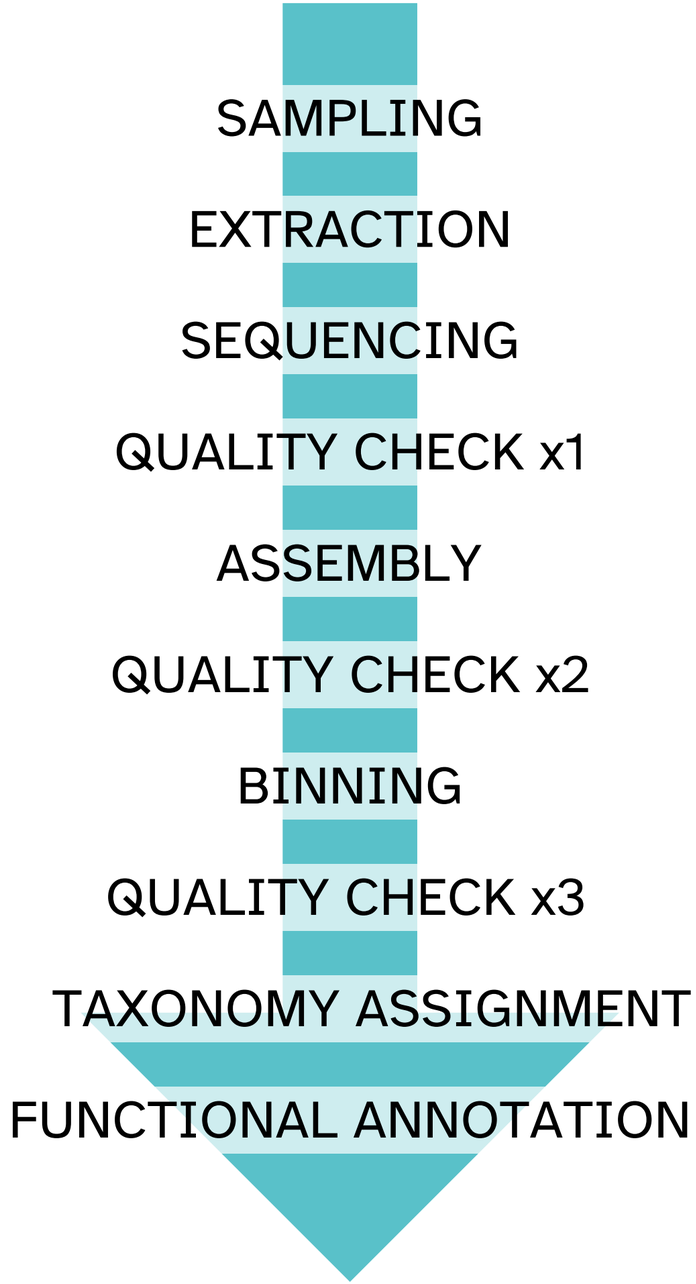
SPECO

Metagenomics

Metagenomic studies the genetic material recovered directly from environmental or clinical samples through sequencing techniques.

This approach is a **culture-independent** technique, thus providing insights into the majority of microorganisms that cannot be easily cultured.

It enables the study of microbial communities in diverse environments, contributing to our understanding of microbial **diversity** and **ecosystem** functions.



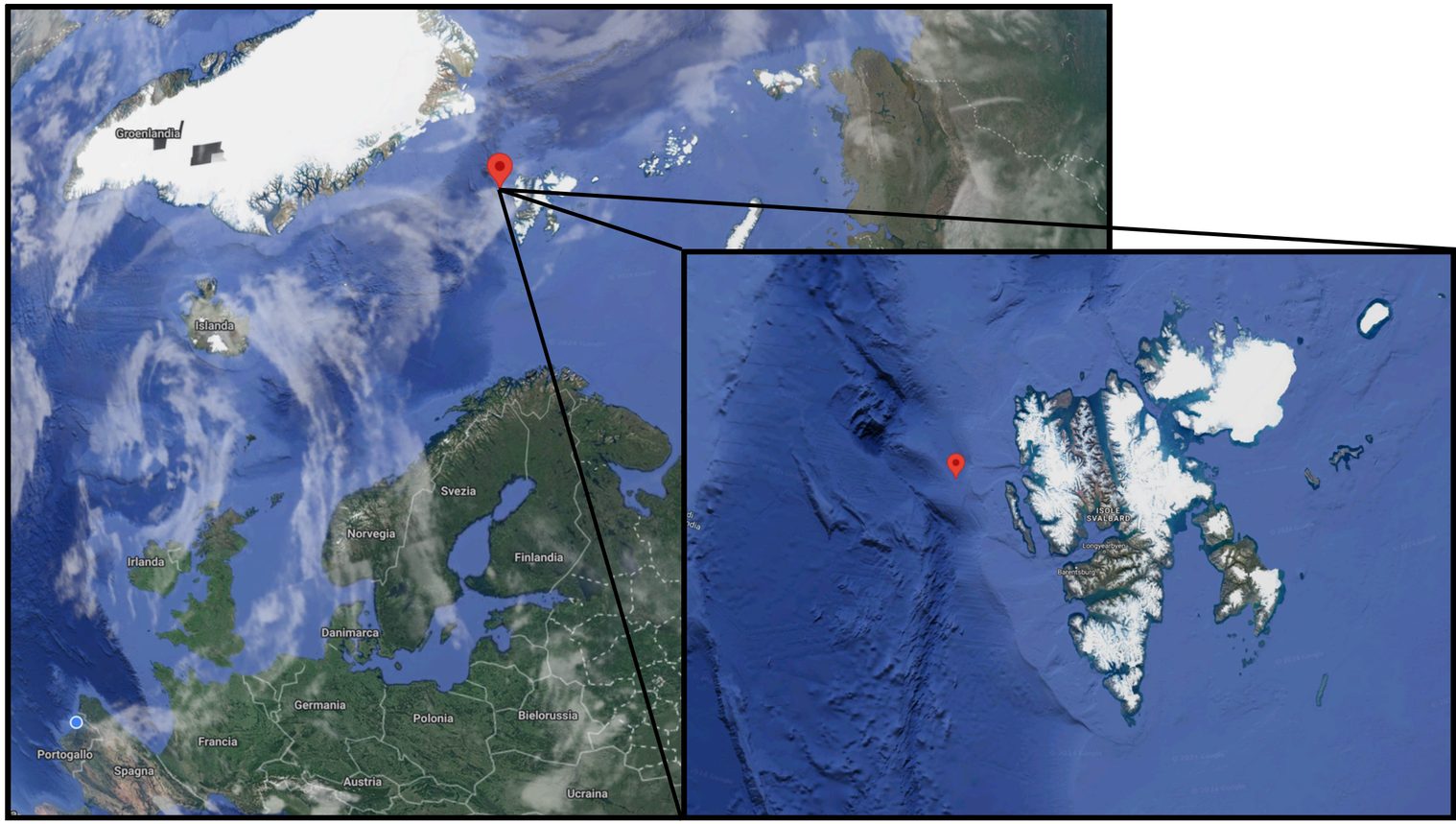
Today's example

Our dataset is composed of **3 samples** from 2019 collected in the **Arctic ocean**.

Samples were collected at **3 different depth**: 5, 25, and 1100 m.

These depths can be classified as “surface”, “DCM”, and “bottom”.

They represent 3 important point of the water column.

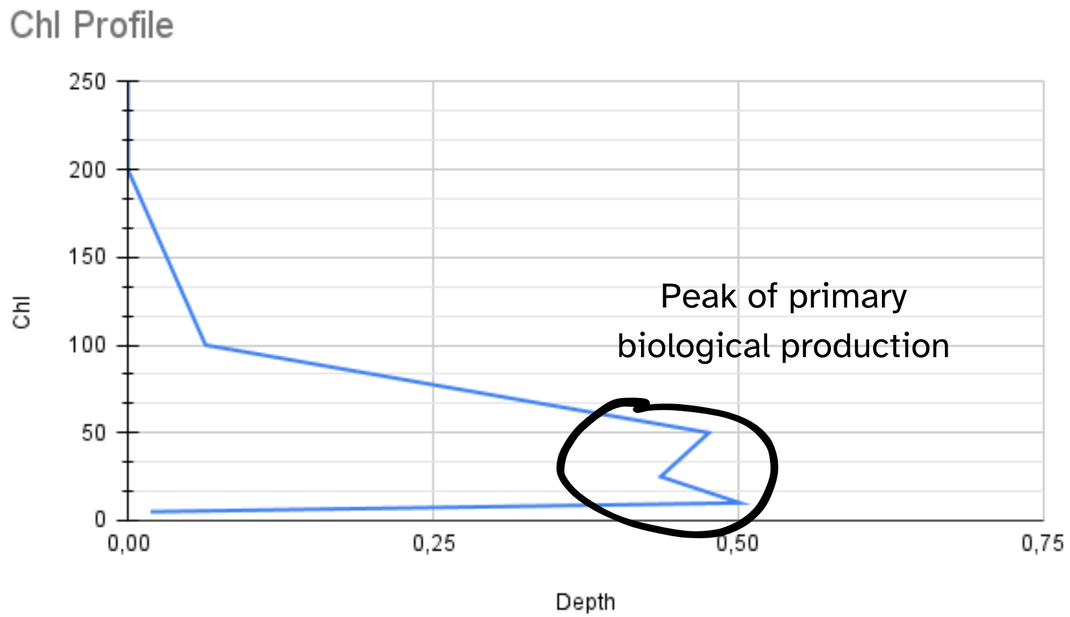


Today's example

DCM

The deep chlorophyll maximum (DCM) is the region below the surface of water with the maximum concentration of chlorophyll.

It holds much of the world's primary productivity, it plays a significant role in nutrient cycling, the flow of energy, and biogeochemical cycles.



Cornec, M., Claustre, H., Mignot, A., Guidi, L., Lacour, L., Poteau, A., et al. (2021). Deep chlorophyll maxima in the global ocean: Occurrences, drivers and characteristics. Global Biogeochemical Cycles, 35, e2020GB006759.

Today's example

Our second example is a dataset composed of 3 samples collected in 2015 during the N-ICE 2015 expedition in Arctic Ocean . Samples were collected at 3 different depth: 5, 20 and 50 m. Samples were collected in the deep Nansen Basin (NB) and in the shallower Yermak Plateau (YP).

NB - is an abyssal plain with water-depths of around 3 km in the Arctic Ocean

YP - local hotspot for vertical mixing and cooling of Atlantic water.

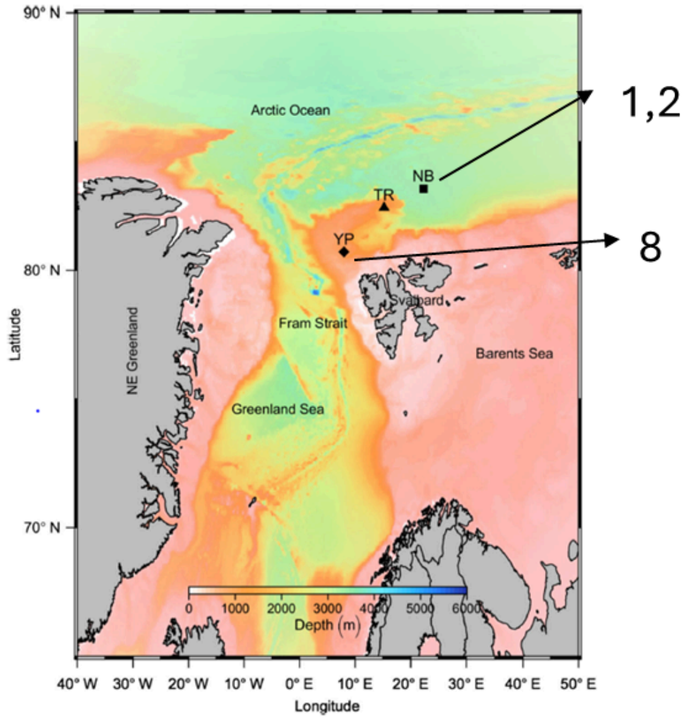


Figure 1 – Location of the samples in study.

MICROBIAL GENOMICS

RESEARCH ARTICLE

Rego *et al.*, *Microbial Genomics* 2021;7:000731
DOI 10.1099/mgen.0.000731

MICROBIOLOGY SOCIETY

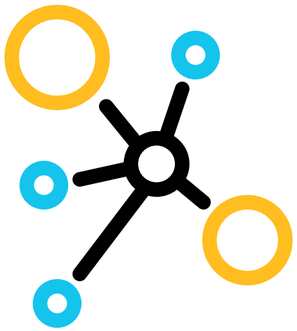
OPEN DATA OPEN ACCESS

Secondary metabolite biosynthetic diversity in Arctic Ocean metagenomes

Adriana Rego[1,2], Antonio Fernandez-Guerra[3], Pedro Duarte[4], Philipp Assmy[4], Pedro N. Leão[1,*] and Catarina Magalhães[1,5,*]

Today's example

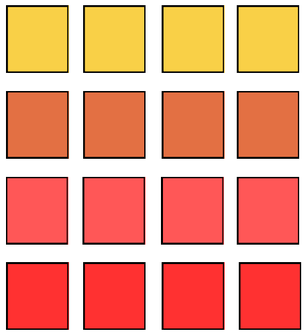
Our questions are:



Who is there?

Community structure

This will be answered by assigning the taxonomy to the obtained MAGs.



What are they doing?

Functional potential

This will be answered by annotating the predicted genes from the obtained MAGs.

Quality Check

Quality control is an essential first step in sequencing data analysis

The QC step measures a set of statistics to assess if its content matches the experiment expectations and if the data is suitable for downstream analysis.

FastQC is a well-established and famous program which provide a simple way to do QC on raw sequence data coming from high throughput sequencing pipelines.

It provides a modular set of analyses which gives a quick impression of whether the data has any problems.

- General Quality
- Per Tile Sequence Quality
- Per Sequence Quality Scores
- Per Base Sequence Content
- Per Sequence GC content
- Per Base N Content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented Sequences
- Adapter Content

General Quality

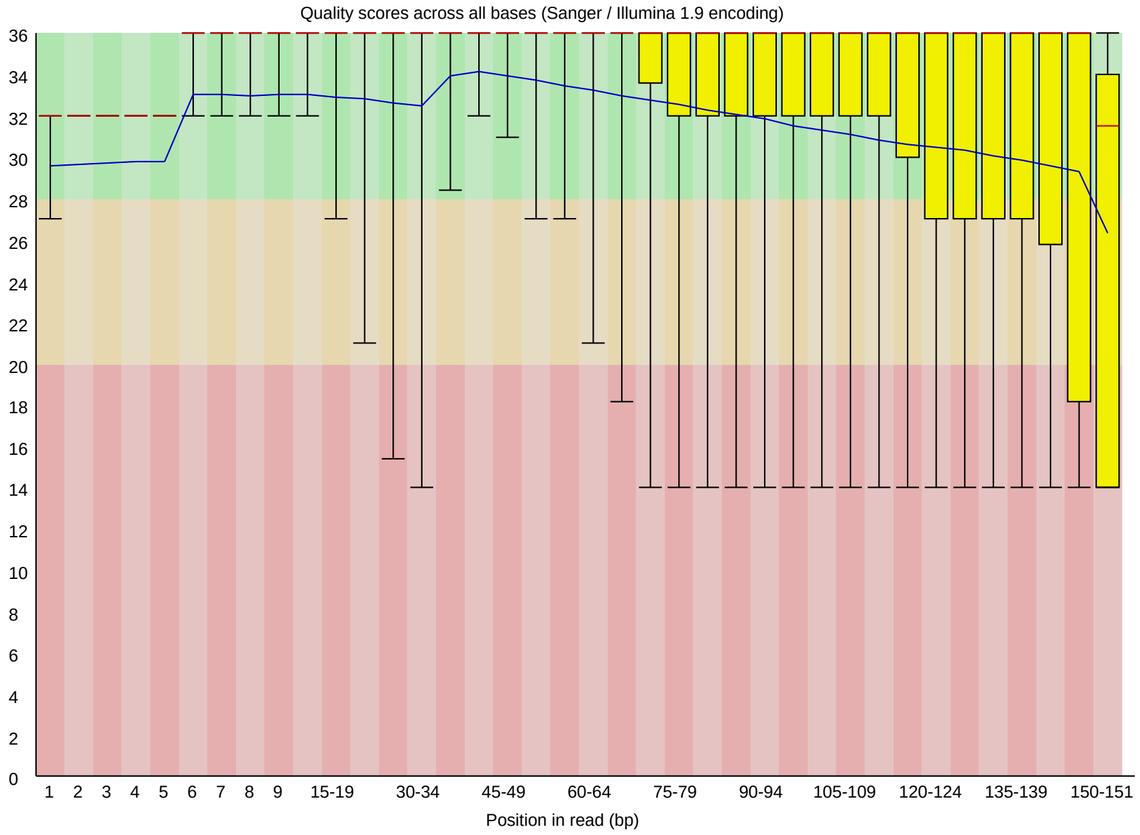
For each position a BoxWhisker type plot is drawn. The elements of the plot are as follows:

- The central red line is the median value
- The yellow box is the inter-quartile range (25-75%)
- The upper and lower whiskers are the 10% and 90% points
- The blue line is the mean quality

The y-axis on the graph shows the quality scores. The higher the score the better the base call.

The background of the graph divides the y axis into very good quality calls (green), calls of reasonable quality (orange), and calls of poor quality (red).

The quality of calls on most platforms will degrade as the run progresses, so it is common to see base calls falling into the orange area towards the end of a read.



Per Tile Quality

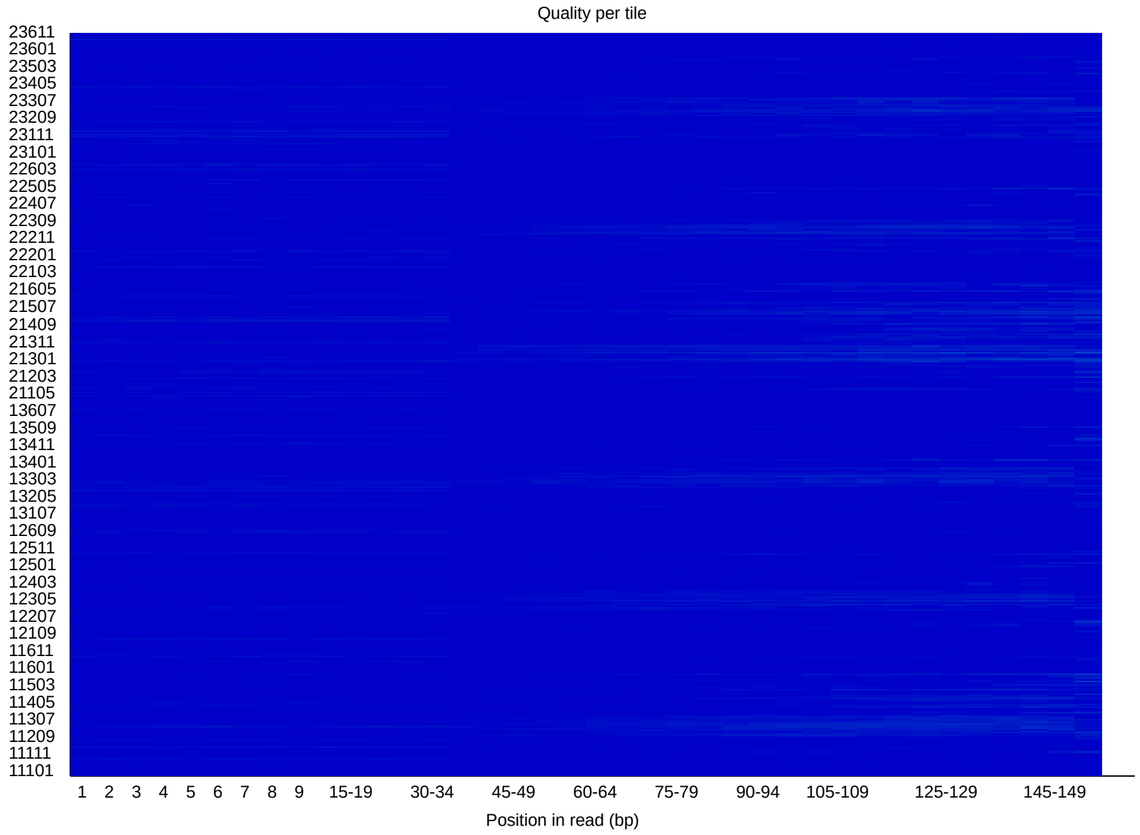
This graph will only appear in your analysis results if you're using an Illumina library which retains its original sequence identifiers.

Encoded in these is the flowcell tile from which each read came.

The graph allows you to look at the quality scores from each tile across all of your bases to see if there was a loss in quality associated with only one part of the flowcell.

The plot shows the deviation from the average quality for each tile. The colours are on a cold to hot scale, with cold colours being positions where the quality was at or above the average for that base in the run, and hotter colours indicate that a tile had worse qualities than other tiles for that base.

A good plot should be blue all over.

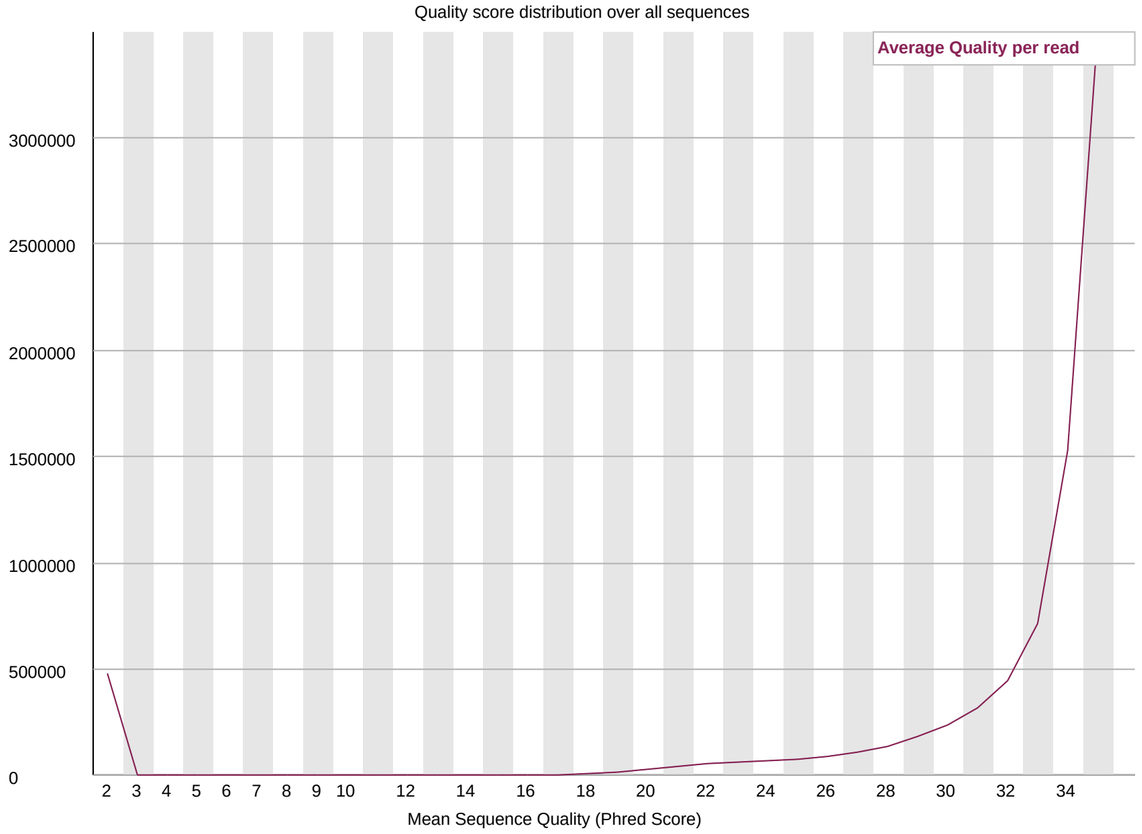


Quality per sequence

The per sequence quality score report allows you to see if a subset of your sequences have universally low quality values.

It is often the case that a subset of sequences will have universally poor quality, often because they are poorly imaged (on the edge of the field of view etc), however these should represent only a small percentage of the total sequences.

If a significant proportion of the sequences in a run have overall low quality then this could indicate some kind of systematic problem - possibly with just part of the run (for example one end of a flowcell).



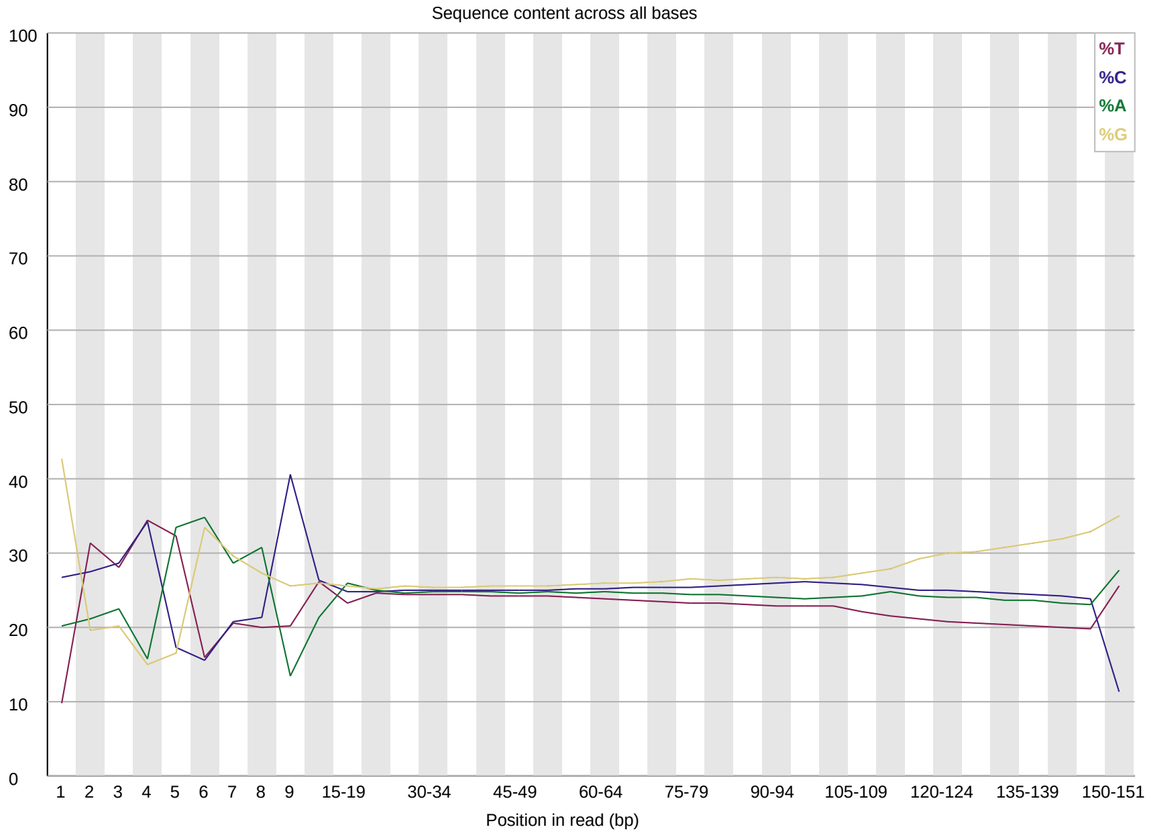
Sequence content

Per Base Sequence Content plots out the proportion of each base position in a file for which each of the four normal DNA bases has been called.

In a random library you would expect that there would be little to no difference between the different bases of a sequence run, so the lines in this plot should run parallel with each other.

The relative amount of each base should reflect the overall amount of these bases in your genome, but in any case they should not be hugely imbalanced from each other.

Some types of library will always produce biased sequence composition, normally at the start of the read (e.g. by priming using random hexamers, RNA-Seq libraries, fragmented using transposases).



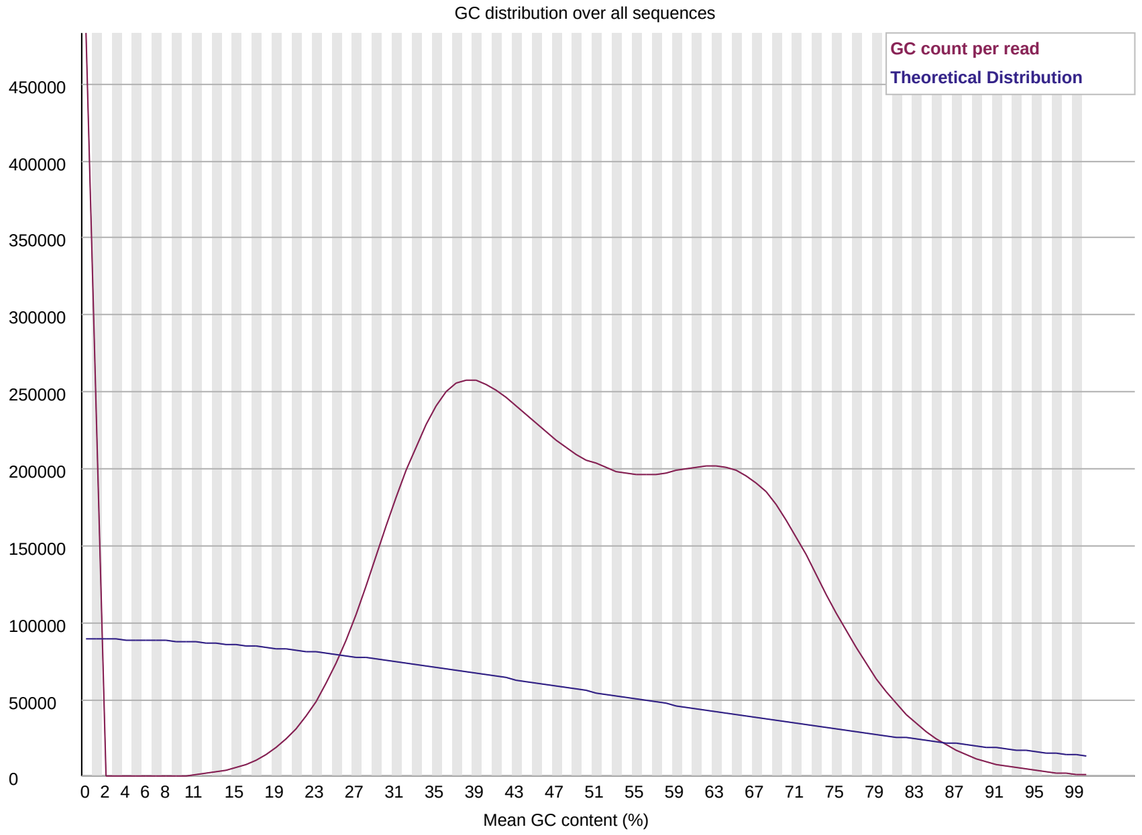
GC content

This module measures the GC content across the whole length of each sequence in a file and compares it to a modelled normal distribution of GC content.

In a normal random library you would expect to see a roughly normal distribution of GC content where the central peak corresponds to the overall GC content of the underlying genome.

Since we don't know the the GC content of the genome the modal GC content is calculated from the observed data and used to build a reference distribution.

An unusually shaped distribution could indicate a contaminated library or some other kinds of biased subset.



N content

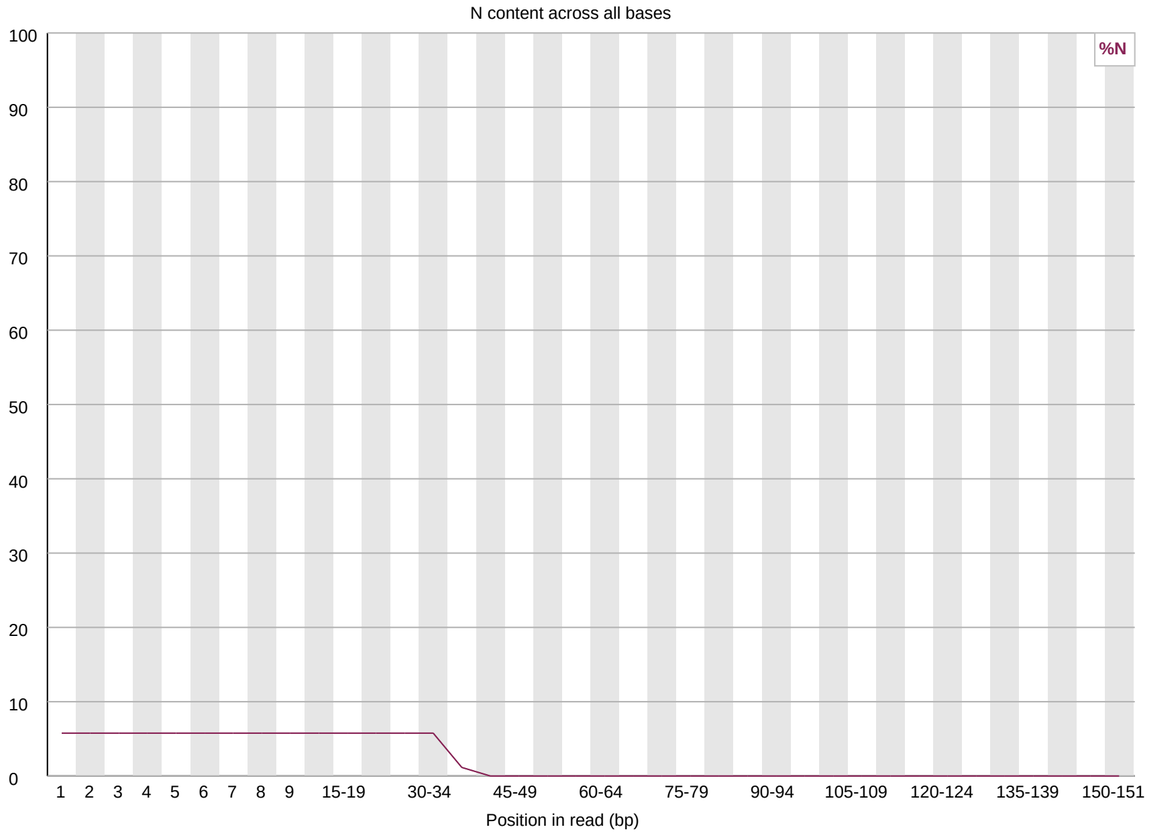
If a sequencer is unable to make a base call with sufficient confidence then it will normally substitute an N rather than a conventional base call

This module plots out the percentage of base calls at each position for which an N was called.

It's not unusual to see a very low proportion of Ns appearing in a sequence, especially nearer the end of a sequence.

However, if this proportion rises above a few percent it suggests that the analysis pipeline was unable to interpret the data well enough to make valid base calls.

The most common reason for the inclusion of significant proportions of Ns is a general loss of quality.



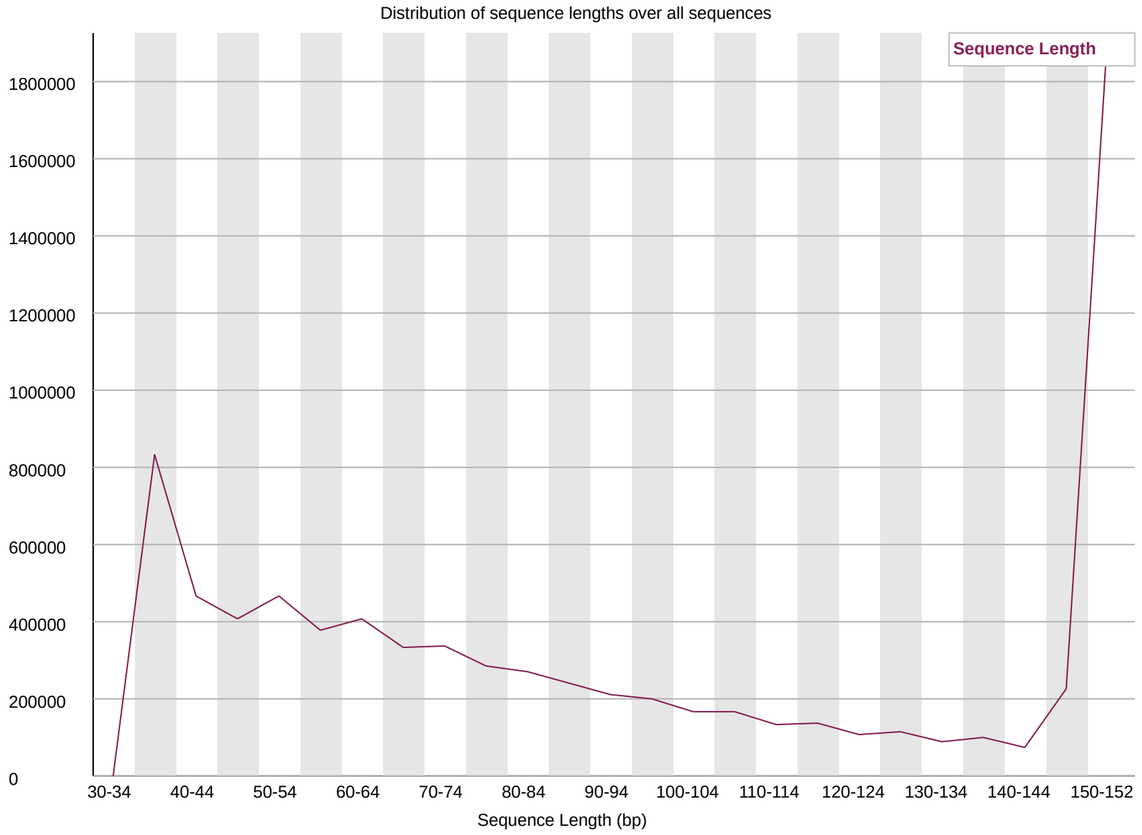
Sequence length distribution

Some high throughput sequencers generate sequence fragments of uniform length, but others can contain reads of wildly varying lengths.

Even within uniform length libraries some pipelines will trim sequences to remove poor quality base calls from the end.

This module generates a graph showing the distribution of fragment sizes in the file which was analysed.

In many cases this will produce a simple graph showing a peak only at one size, but for variable length FastQ files this will show the relative amounts of each different size of sequence fragment.



Duplication

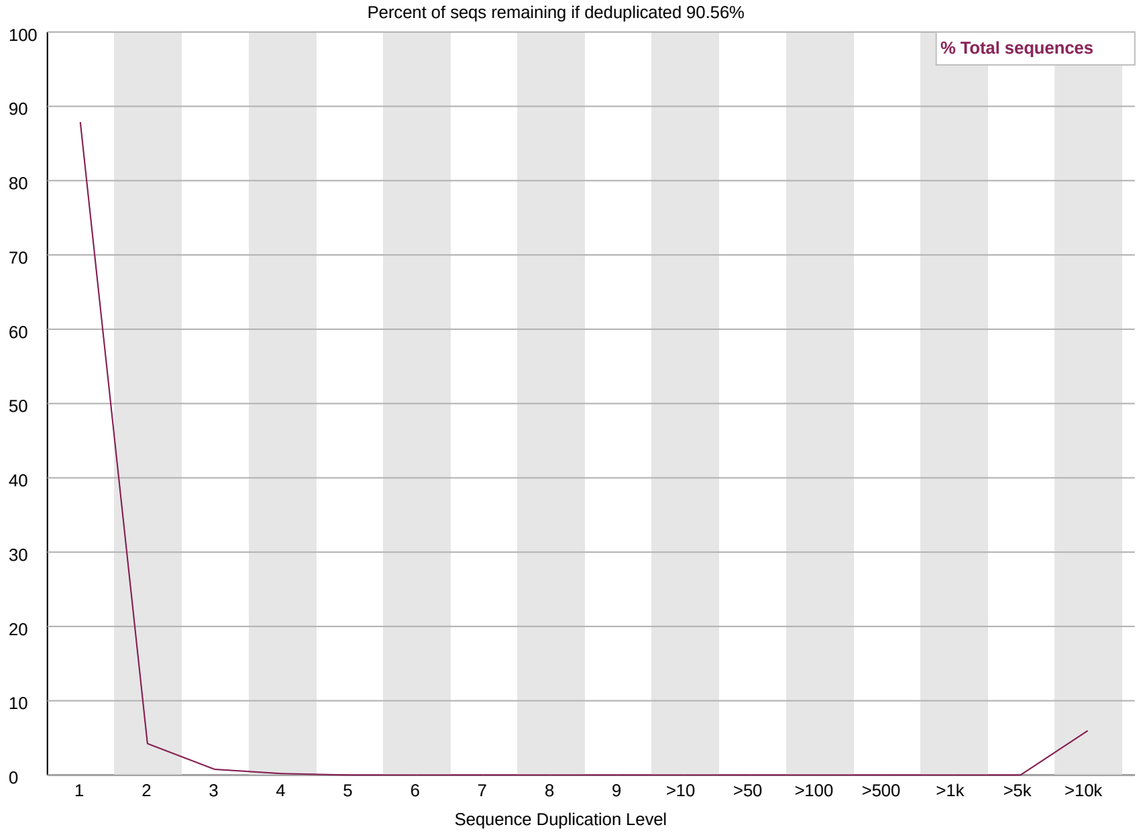
In the red plot the sequences are de-duplicated and the proportions shown are the proportions of the deduplicated set which come from different duplication levels in the original data.

In a properly diverse library most sequences should fall into the far left of the plot

A general level of enrichment, indicating broad oversequencing in the library will tend to flatten the lines, lowering the low end and generally raising other categories.

More specific enrichments of subsets, or the presence of low complexity contaminants will tend to produce spikes towards the right of the plot.

The module also calculates an expected overall loss of sequence were the library to be deduplicated. This headline figure is shown at the top of the plot and gives a reasonable impression of the potential overall level of loss.



A warning or error in this module is simply a statement that you have exhausted the diversity in at least part of your library and are re-sequencing the same sequences.

Overrepresented sequence

Sequence	Count	Percentage	Possible Source
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	367042	3.943674017750734	No Hit
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG	9882	0.10617691338705858	No Hit

A normal high-throughput library will contain a diverse set of sequences, with no individual sequence making up a tiny fraction of the whole. Finding that a single sequence is very overrepresented in the set either means that it is highly biologically significant, or indicates that the library is contaminated, or not as diverse as you expected.

For each overrepresented sequence the program will look for matches in a database of common contaminants and will report the best hit it finds.

Adapters

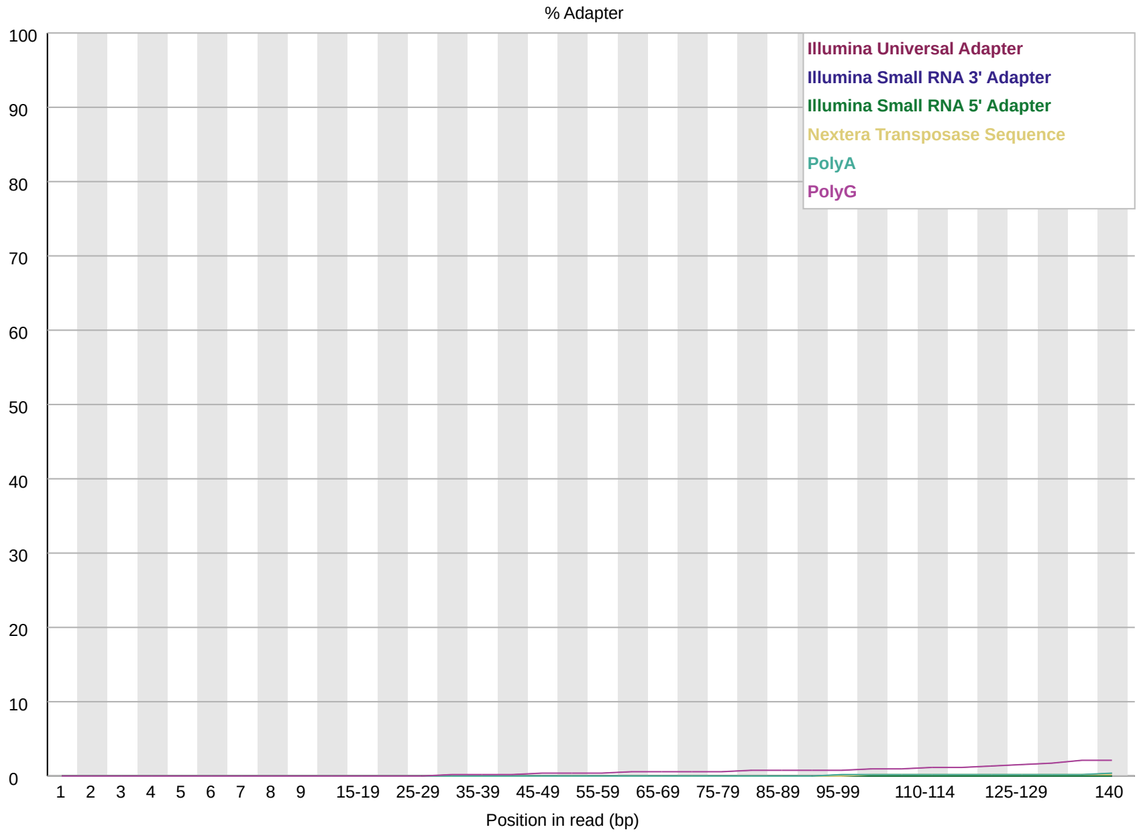
This module does a specific search for a set of separately defined Kmers and will give you a view of the total proportion of your library which contain these Kmers.

A results trace will always be generated for all of the sequences present in the adapter config file so you can see the adapter content of your library, even if it's low.

It is useful to know if your library contains a significant amount of adapter in order to be able to assess whether you need to adapter trim or not.

The plot itself shows a cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

Once a sequence has been seen in a read it is counted as being present right through to the end of the read so the percentages you see will only increase as the read length goes on.



Trimming

Trimmomatic is a fast multithreaded command line tool that can be used to trim and crop Illumina (FASTQ) data as well as to remove adapters.

Trimmomatic works with FASTQ files (using phred + 33 or phred + 64 quality scores) and support compressed files.

How to know the phred score used in my data?

Basic Statistics

Measure	Value
Filename	M19-88_METAG_R2.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	9307108
Total Bases	1 Gbp
Sequences flagged as poor quality	0
Sequence length	35-151
%GC	57

Variant	ASCII Range	Offset	Quality Range	Notes
sanger	33 to 126	33	0 to 93	Equivalent to OBF's fastq-sanger.
illumina1.3	64 to 126	64	0 to 62	Equivalent to OBF's fastq-illumina. Use this if your data was generated using Illumina 1.3-1.7 software.
illumina1.8	33 to 95	33	0 to 62	Equivalent to sanger but with 0 to 62 quality score range check. Use this if your data was generated using Illumina 1.8 software or later.
solexa	59 to 126	64	-5 to 62	Not currently implemented.

Trimming

The current trimming steps are:

- **ILLUMINACLIP**: Cut adapter and other illumina-specific sequences from the read.
- **SLIDINGWINDOW**: Performs a sliding window trimming approach. It starts scanning at the 5‟ end and clips the read once the average quality within the window falls below a threshold.
- **MAXINFO**: An adaptive quality trimmer which balances read length and error rate to maximise the value of each read
- **LEADING**: Cut bases off the start of a read, if below a threshold quality
- **TRAILING**: Cut bases off the end of a read, if below a threshold quality CROP: Cut the read to a specified length by removing bases from the end
- **HEADCROP**: Cut the specified number of bases from the start of the read
- **MINLEN**: Drop the read if it is below a specified length
- **AVGQUAL:** Drop the read if the average quality is below the specified level
- **TOPHRED33**: Convert quality scores to Phred-33
- **TOPHRED64**: Convert quality scores to Phred-64

Example

```
trimmomatic PE -threads 96 -phred33 -trimlog trim_log.log -summary trim_sum.log raw_data/M19-88_METAG_R1.fastq raw_data/M19-88_METAG_R2.fastq M19-88_f_p.fastq M19-88_f_u.fastq M19-88_r_p.fastq M19-88_r_u.fastq LEADING:10 TRAILING:10 SLIDINGWINDOW:5:20
```

The different processing steps occur in the order in which the steps are specified on the command line.

In this case:

1. Remove leading low quality or N bases (below quality 10)
2. Remove trailing low quality or N bases (below quality 10)
3. Scan the read with a 5-base wide sliding window, cutting when the average quality per base drops below 20

Assembly

Genome assembly is the reconstruction of genomes from the smaller DNA segments called **reads** which are generated by a sequencing experiment.

2 kind of assembly:

- **de novo** assembly involves reconstructing genomes directly from the read data
- **comparative** assembly involves reconstructing genomes helped by reference genomes

The general problem of de novo assembly is that cannot be solved efficiently.
The most widely used strategies are greedy, overlap-layout-consensus (OLC), and **De Bruijn graph**.

In De Brujin graph, assembly problem reduces to finding an **eulerian** path – a path through the graph that visits each edge once.

MEGAHIT is a NGS assembler that makes use of succinct de Bruijn graphs (SdBG; Bowe et al., 2012), which are compressed representation of de Bruijn graphs.

Ghurye JS, Cepeda-Espinoza V, Pop M. Metagenomic Assembly: Overview, Challenges and Applications. Yale J Biol Med. 2016;89(3):353-362. Published 2016 Sep 30.

De Brujin Graph

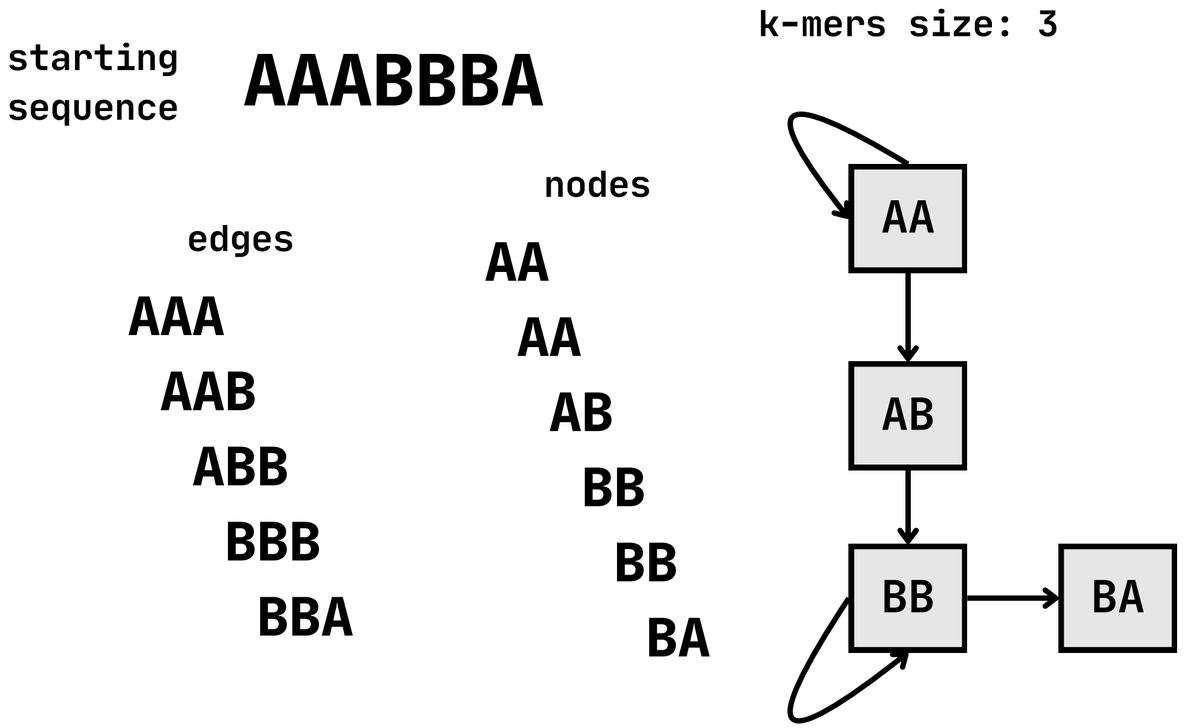
Construction

K-mer size *n*

Nodes:
Each node in the graph corresponds to a string of length *n-1*.

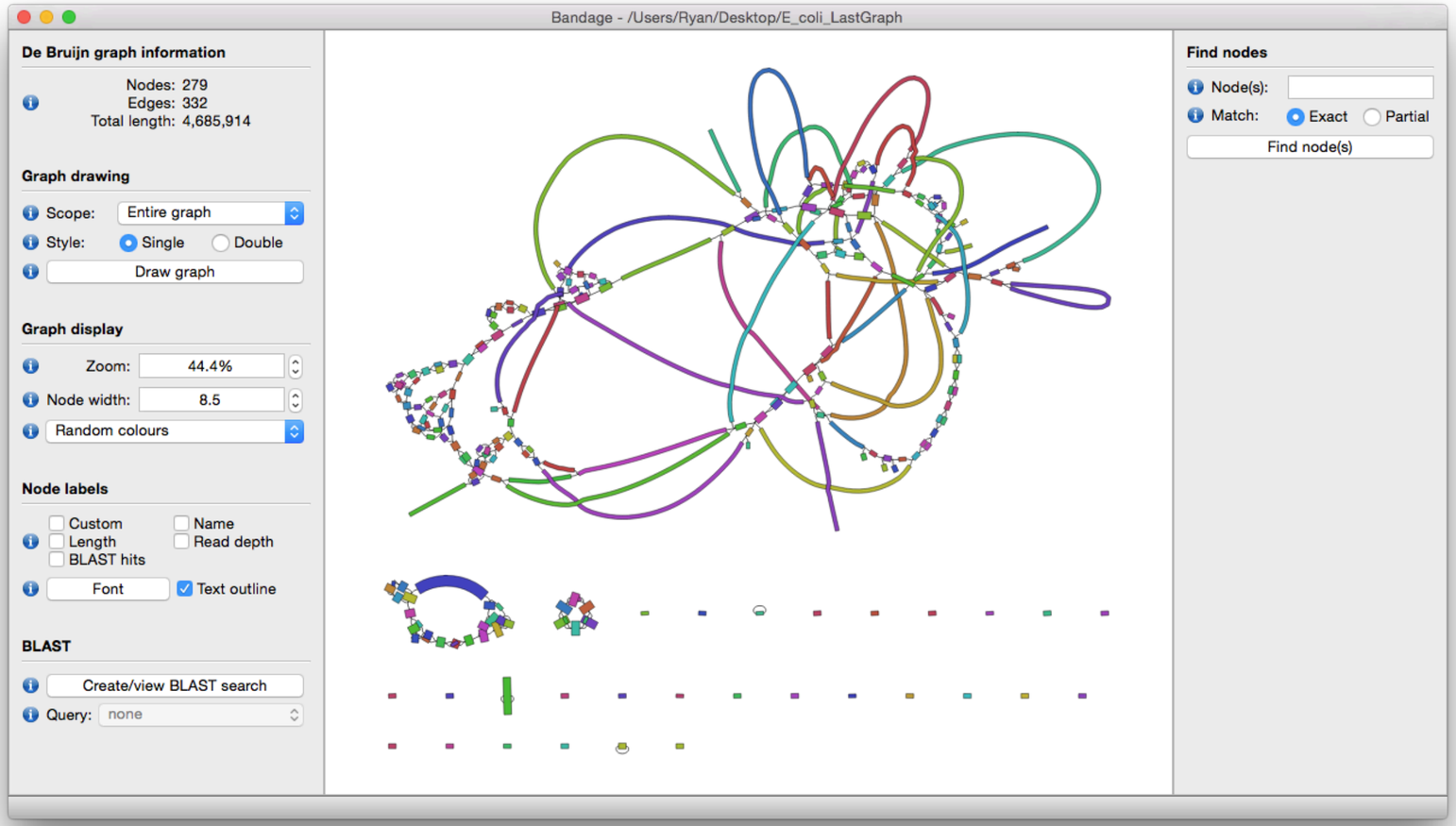
Edges:
Directed edge corresponding to an overlap of length *n-2* (representing the k-mer of length *n*) between two nodes.

Sequences can be joined using this kind of grapth to form **contigs**.



Genomic assembly can be visualized and manually refined with ad-hoc softwares like **Bandage**.

However, this process is time consuming and present challenges.



Quast QUality ASsessment Tool

QUAST evaluates genome/metagenome assemblies by computing various metrics.

The QUAST package works both with and without reference genomes.

Metrics based only on contigs (without reference genomes):

Statistics without reference			
# contigs	459 106	20	13
# contigs (>= 0 bp)	6 708 752	45	32
# contigs (>= 1000 bp)	85 916	12	7
# contigs (>= 5000 bp)	2437	5	6
# contigs (>= 10000 bp)	621	5	5
# contigs (>= 25000 bp)	97	3	3
# contigs (>= 50000 bp)	23	1	1
Largest contig	200 426	55 106	54 221
Total length	400 087 974	180 190	172 703
Total length (>= 0 bp)	2 001 423 590	186 593	177 895
Total length (>= 1000 bp)	156 137 253	173 575	168 101
Total length (>= 5000 bp)	23 869 179	160 170	164 654
Total length (>= 10000 bp)	11 761 585	160 170	157 666
Total length (>= 25000 bp)	4 049 568	131 533	131 139
Total length (>= 50000 bp)	1 677 793	55 106	54 221
N50	827	48 458	49 658
N90	538	3898	10 428
auN	2108.6	36 839	38 107
L50	134 490	2	2
L90	381 860	6	5
GC (%)	...	...	...

N50:
This is the length of the shortest contig in the assembly such that the sum of lengths of this contig and all longer contigs covers 50% of the total assembly length. It's a measure of the assembly's contiguity, with a higher N50 indicating a better assembly.

N90:
Similar to N50, but it represents the length of the shortest contig that covers 90% of the total assembly length. It focuses on the longer contigs that make up most of the assembly.

L50:
This is the minimum number of contigs needed to cover 50% of the total assembly length. A lower L50 indicates a more contiguous assembly because fewer, longer contigs are needed to reach 50% coverage.

L90:
Like L50, but it refers to the number of contigs needed to cover 90% of the total assembly length. It provides insight into how many contigs are needed to account for the bulk of the assembly.

auN:
This is the area under the cumulative contig length curve normalized by the maximum possible area. It provides a more comprehensive summary of the assembly quality, taking into account both the number and length of contigs. A higher auN value indicates a better overall assembly.

Statistics without reference

# contigs	459 106	20	13
# contigs (>= 0 bp)	6 708 752	45	32
# contigs (>= 1000 bp)	85 916	12	7
# contigs (>= 5000 bp)	2437	5	6
# contigs (>= 10000 bp)	621	5	5
# contigs (>= 25000 bp)	97	3	3
# contigs (>= 50000 bp)	23	1	1
Largest contig	200 426	55 106	54 221
Total length	400 087 974	180 190	172 703
Total length (>= 0 bp)	2 001 423 590	186 593	177 895
Total length (>= 1000 bp)	156 137 253	173 575	168 101
Total length (>= 5000 bp)	23 869 179	160 170	164 654
Total length (>= 10000 bp)	11 761 585	160 170	157 666
Total length (>= 25000 bp)	4 049 568	131 533	131 139
Total length (>= 50000 bp)	1 677 793	55 106	54 221
N50	827	48 458	49 658
N90	538	3898	10 428
auN	2108.6	36 839	38 107
L50	134 490	2	2
L90	381 860	6	5
GC (%)	...	...	...

Statistics without reference

# contigs	459 106	20	13
# contigs (>= 0 bp)	6 708 752	45	32
# contigs (>= 1000 bp)	85 916	12	7
# contigs (>= 5000 bp)	2437	5	6
# contigs (>= 10000 bp)	621	5	5
# contigs (>= 25000 bp)	97	3	3
# contigs (>= 50000 bp)	23	1	1
Largest contig	200 426	55 106	54 221
Total length	400 087 974	180 190	172 703
Total length (>= 0 bp)	2 001 423 590	186 593	177 895
Total length (>= 1000 bp)	156 137 253	173 575	168 101
Total length (>= 5000 bp)	23 869 179	160 170	164 654
Total length (>= 10000 bp)	11 761 585	160 170	157 666
Total length (>= 25000 bp)	4 049 568	131 533	131 139
Total length (>= 50000 bp)	1 677 793	55 106	54 221
N50	827	48 458	49 658
N90	538	3898	10 428
auN	2108.6	36 839	38 107
L50	134 490	2	2
L90	381 860	6	5
GC (%)	...	...	...

Binning

Most of the current assemblers do not represent complete microbial genomes with single scaffolds. Many metagenome binning tools have been developed to **group** the scaffolds into clusters to represent the whole genome of an organism (**bins**).

These draft genomes are often derived from individual species or “population genomes”, representing consensus sequences of different strains. They approximate full genomes as they can contain a near full set of genes.

MetaBAT2 integrates empirical probabilistic distances of genome abundance (**read depths**) and tetranucleotide frequency (**TNF**) for metagenome binning.

TNF: This strategy divides the genome in k-mers to group sequences based on the assumption that sequences from the same genome will have similar k-mers frequencies.

read depths: This method leverages variations in sequence coverage across multiple samples or different regions of the genome. Sequences with similar coverage patterns are grouped into the same bin, assuming they belong to the same organism.

Yang C, Chowdhury D, Zhang Z, et al. A review of computational tools for generating metagenome-assembled genomes from metagenomic sequencing data. Comput Struct Biotechnol J. 2021;19:6301-6314. Published 2021 Nov 23. doi:10.1016/j.csbj.2021.11.028

Bowtie2

Metabat2 is fed with 2 types of input files: the **assembly (FASTA)** and the **alignment (BAM)**.
The alignment is composed by one or more BAM files that represent the alignment between the assembly and the raw data and it is used by Metabat2 to retrieve the read depth and successfully group the contig/scaffolds in bins.

In order to obtain such files, it is necessary to perform a sequence alignment. This is possible by using **Bowtie2**.

A general pipeline with Bowtie2 is to:

1. create the index file of the “reference genome”
2. aling the query to the index
3. use **samtools** to sort the alignment and convert it to binary

After that, it is possible to run Metabat2

Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012 Mar 4;9(4):357-9. doi: 10.1038/nmeth.1923.

Bowtie2

bowtie2-build

bowtie2-build builds an index from a set of DNA sequences composed of a set of 6 files. These files are all that is needed to align reads to that reference.

The index construction is based on the FM Index of Ferragina and Manzini, which in turn is based on the Burrows-Wheeler transform. The algorithm used to build the index is based on the blockwise algorithm of Kärkkäinen's Blockwise algorithm

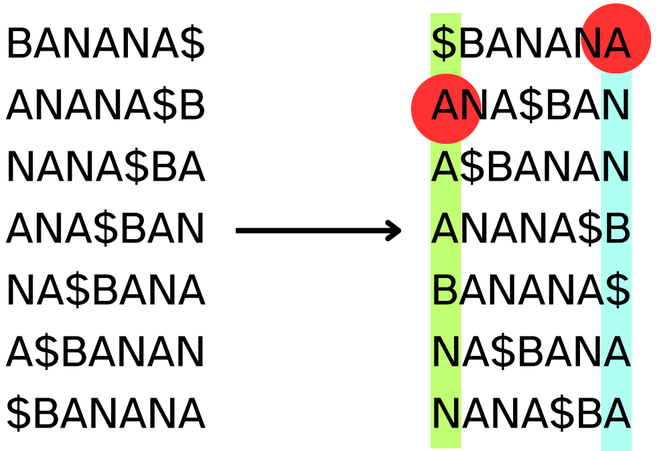
Burrows-Wheeler Transform

BWT is a data transformation algorithm rearranges a string of text into runs of similar characters to be easily compressed. The compression is important for the FM index-based querying.

Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012 Mar 4;9(4):357-9. doi: 10.1038/nmeth.1923.

BWT Query

Input String: BANANA$



BWT: ANNB$AA
F: $AAABNN
→ Compression: A2NB$2A

BWT is the last column of the lexicographically ordered rotation matrix

F is the first column of the same matrix or the BWT ordered lexicographically

Property that make BWT (T) reversible is the LF mapping.
ith occurence of a character in the last column (BWT) is the same text occurence as the ith occurence in the first column (F).

1st A in BWT is the 1st A in F … … …

Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012 Mar 4;9(4):357-9. doi: 10.1038/nmeth.1923.

BWT Query

$	A	$$_1$$	$$A_1$$
A	N	$$A_1$$	$$N_1$$
A	N	$$A_2$$	$$N_2$$
A	B	$$A_3$$	$$B_1$$
B	$	$$B_1$$	$$\$_1$$
N	A	$$N_1$$	$$A_2$$
N	A	$$N_2$$	$$A_3$$

We assign a number i depending on the ith occurence of the letter in the string.

$ is the last character of the string by definition.

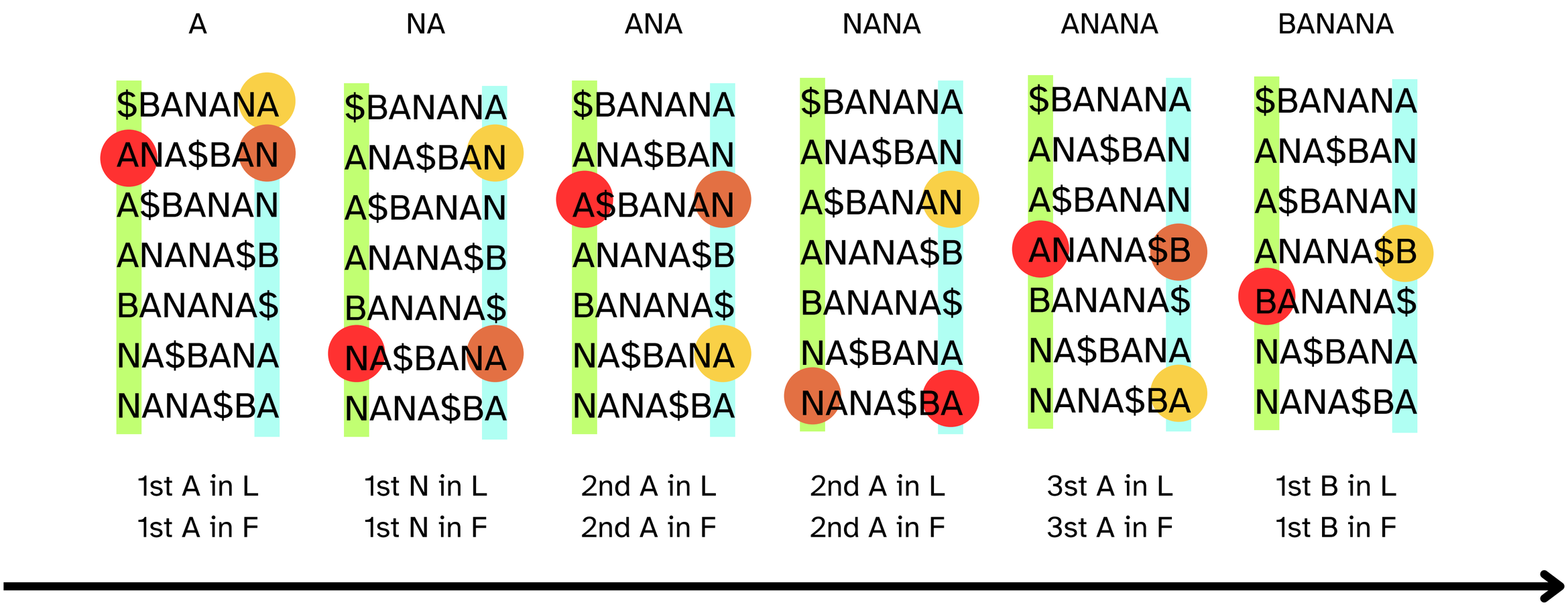
Since the matrix is based on the rotation of the string we know that A_1 is the **last** character of the string and B_1 is the **first** character of the string.

We then reconstruct the string.

$$B_1, A_3, N_2, A_2, N_1, A_1, \$_1$$

Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012 Mar 4;9(4):357-9. doi: 10.1038/nmeth.1923.

BWT Query



Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012 Mar 4;9(4):357-9. doi: 10.1038/nmeth.1923.

BWT Query

i	F	L	LF(i)
0	$$\$_1$$	$$A_1$$	1
1	$$A_1$$	$$N_1$$	5
2	$$A_2$$	$$N_2$$	6
3	$$A_3$$	$$B_1$$	4
4	$$B_1$$	$$\$_1$$	0
5	$$N_1$$	$$A_2$$	2
6	$$N_2$$	$$A_3$$	3

Query: AN

LF(0) = 1 because th item in row 0 in L ($$A_1$$) is in the row 1 in the F string

This means:

- LF(1) = 5
- LF(2) = 6
- LF(3) = 4
- LF(4) = 0
- LF(5) = 2
- LF(6) = 3

The query is searched backward (starting from N and ending with A).
initial i range [0,6]
N instances: $$N_1$$, $$N_2$$ --> LF($$N_1$$) = 5, LF($$N_2$$) = 6
A instances in the i range [5,6]: $$A_2$$, $$A_3$$
$$A_2$$, $$A_3$$ tells where the query start in the orginal string

$$B_1, A_3, N_2, A_2, N_1, A_1, \$_1$$

$$B_1, \mathbf{A_3, N_2}, \mathbf{A_2, N_1}, A_1, \$_1$$

BWT Query

i	F	L	LF(i)
0	$\$_1$	A_1	1
1	A_1	N_1	5
2	A_2	N_2	6
3	A_3	B_1	4
4	B_1	$\$_1$	0
5	N_1	A_2	2
6	N_2	A_3	3

Query: NAN

LF(0) = 1 because th item in row 0 in L (A_1) is in the row 1 in the F string

This means:

- LF(1) = 5
- LF(2) = 6
- LF(3) = 4
- LF(4) = 0
- LF(5) = 2
- LF(6) = 3

The query is searched backward (starting from N and ending with N).
initial i range [0,6]
N instances: N_1, N_2 --> LF(N_1) = 5, LF(N_2) = 6
A instances in the i range [5,6]: A_2 , A_3 --> LF(A_2) = 2, LF(A_3) = 3
N instances in the i range [2,3]: N_2
N_2 tells where the query start in the orginal string

$B_1, A_3, N_2, A_2, N_1, A_1, \$_1$

$B_1, A_3,$ **N_2, A_2, N_1**$, A_1, \$_1$

Quality Control and MAG definition

To generate a MAG, metagenomic sequence reads are assembled into contigs (**assembly**) and contigs are grouped into scaffold (**assembly**), and these groups are then assigned to discrete population bins (**binning**).

The 3 most important criteria for assessing MAG quality are **assembly quality**, **genome completeness**, and a measure of **contamination**.

Finished	Single contiguous sequence / consensus error rate >= Q50
High Quality	Presence of 23S, 16S, and 5S rRNA genes and >= 18 tRNAs. **Completion** >90% **Contamination** <5%
Medium Quality	**Completion** >50% **Contamination** <10%
Low Quality	**Completion** <50% **Contamination** <10%

Bowers, R., Kyrpides, N., Stepanauskas, R. et al. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. Nat Biotechnol 35, 725–731 (2017). https://doi.org/10.1038/nbt.3893

CheckM

CheckM provides a set of tools for assessing the quality of genomes recovered from isolates, single cells, or metagenomes.

It provides estimates of genome completeness and contamination by using collocated sets of genes that are ubiquitous and single-copy within a phylogenetic lineage.

Bin Id	Marker lineage	# genomes	# markers	# marker sets	0	1	2	3	4	5+	Completeness	Contamination	Strain heterogeneity
bin.12	o__Actinomycetales (UID1663)	488	310	185	26	279	5	0	0	0	92.31	2.24	80.00
bin.8	k__Bacteria (UID203)	5449	104	58	9	58	37	0	0	0	89.86	32.84	29.73
bin.5	k__Bacteria (UID203)	5449	103	58	30	26	46	1	0	0	84.77	52.41	93.88
bin.11	s__algicola (UID2847)	33	496	263	152	203	112	25	2	2	72.20	36.97	16.89
bin.1	k__Bacteria (UID203)	5449	104	58	34	53	17	0	0	0	65.83	4.80	5.88
bin.6	f__Rhodobacteraceae (UID3361)	46	654	332	248	403	3	0	0	0	57.96	0.75	0.00
bin.10	k__Archaea (UID2)	207	149	107	95	27	16	9	1	1	30.46	16.25	0.00
bin.7	k__Bacteria (UID203)	5449	104	58	55	33	10	6	0	0	28.68	5.02	14.29
bin.9	k__Bacteria (UID203)	5449	104	58	91	13	0	0	0	0	18.97	0.00	0.00
bin.2	root (UID1)	5656	56	24	45	0	6	0	4	1	9.38	14.47	12.50
bin.4	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00
bin.3	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00

CheckM

Bin Id	Marker lineage	# genomes	# markers	# marker sets	0	1	2	3	4	5+	Completeness	Contamination	Strain heterogeneity
bin.12	o__Actinomycetales (UID1663)	488	310	185	26	279	5	0	0	0	92.31	2.24	80.00
bin.8	k__Bacteria (UID203)	5449	104	58	9	58	37	0	0	0	89.86	32.84	29.73
bin.5	k__Bacteria (UID203)	5449	103	58	30	26	46	1	0	0	84.77	52.41	93.88
bin.11	s__algicola (UID2847)	33	496	263	152	203	112	25	2	2	72.20	36.97	16.89
bin.1	k__Bacteria (UID203)	5449	104	58	34	53	17	0	0	0	65.83	4.80	5.88
bin.6	f__Rhodobacteraceae (UID3361)	46	654	332	248	403	3	0	0	0	57.96	0.75	0.00
bin.10	k__Archaea (UID2)	207	149	107	95	27	16	9	1	1	30.46	16.25	0.00
bin.7	k__Bacteria (UID203)	5449	104	58	55	33	10	6	0	0	28.68	5.02	14.29
bin.9	k__Bacteria (UID203)	5449	104	58	91	13	0	0	0	0	18.97	0.00	0.00
bin.2	root (UID1)	5656	56	24	45	0	6	0	4	1	9.38	14.47	12.50
bin.4	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00
bin.3	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00

these columns report how many marker genes were:

1. missing (0)
2. found once (1)
3. found twice (2)
4. ...
5. found five or more than five times (5+)

CheckM

Bin Id	Marker lineage	# genomes	# markers	# marker sets	0	1	2	3	4	5+	Completeness	Contamination	Strain heterogeneity
bin.12	o__Actinomycetales (UID1663)	488	310	185	26	279	5	0	0	0	92.31	2.24	80.00
bin.8	k__Bacteria (UID203)	5449	104	58	9	58	37	0	0	0	89.86	32.84	29.73
bin.5	k__Bacteria (UID203)	5449	103	58	30	26	46	1	0	0	84.77	52.41	93.88
bin.11	s__algicola (UID2847)	33	496	263	152	203	112	25	2	2	72.20	36.97	16.89
bin.1	k__Bacteria (UID203)	5449	104	58	34	53	17	0	0	0	65.83	4.80	5.88
bin.6	f__Rhodobacteraceae (UID3361)	46	654	332	248	403	3	0	0	0	57.96	0.75	0.00
bin.10	k__Archaea (UID2)	207	149	107	95	27	16	9	1	1	30.46	16.25	0.00
bin.7	k__Bacteria (UID203)	5449	104	58	55	33	10	6	0	0	28.68	5.02	14.29
bin.9	k__Bacteria (UID203)	5449	104	58	91	13	0	0	0	0	18.97	0.00	0.00
bin.2	root (UID1)	5656	56	24	45	0	6	0	4	1	9.38	14.47	12.50
bin.4	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00
bin.3	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00

Completeness is reported as percentage of marker genes found in the organism, ranging from 0 to 100.

CheckM

Bin Id	Marker lineage	# genomes	# markers	# marker sets	0	1	2	3	4	5+	Completeness	Contamination	Strain heterogeneity
bin.12	o__Actinomycetales (UID1663)	488	310	185	26	279	5	0	0	0	92.31	2.24	80.00
bin.8	k__Bacteria (UID203)	5449	104	58	9	58	37	0	0	0	89.86	32.84	29.73
bin.5	k__Bacteria (UID203)	5449	103	58	30	26	46	1	0	0	84.77	52.41	93.88
bin.11	s__algicola (UID2847)	33	496	263	152	203	112	25	2	2	72.20	36.97	16.89
bin.1	k__Bacteria (UID203)	5449	104	58	34	53	17	0	0	0	65.83	4.80	5.88
bin.6	f__Rhodobacteraceae (UID3361)	46	654	332	248	403	3	0	0	0	57.96	0.75	0.00
bin.10	k__Archaea (UID2)	207	149	107	95	27	16	9	1	1	30.46	16.25	0.00
bin.7	k__Bacteria (UID203)	5449	104	58	55	33	10	6	0	0	28.68	5.02	14.29
bin.9	k__Bacteria (UID203)	5449	104	58	91	13	0	0	0	0	18.97	0.00	0.00
bin.2	root (UID1)	5656	56	24	45	0	6	0	4	1	9.38	14.47	12.50
bin.4	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00
bin.3	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00

Contamination, as completeness, is reported as percentage ranging from 0 to 100. It refers to the number of marker genes found multiple times, indicating the MAG/SAG present genes from other genomes.

CheckM

Bin Id	Marker lineage	# genomes	# markers	# marker sets	0	1	2	3	4	5+	Completeness	Contamination	Strain heterogeneity
bin.12	o__Actinomycetales (UID1663)	488	310	185	26	279	5	0	0	0	92.31	2.24	80.00
bin.8	k__Bacteria (UID203)	5449	104	58	9	58	37	0	0	0	89.86	32.84	29.73
bin.5	k__Bacteria (UID203)	5449	103	58	30	26	46	1	0	0	84.77	52.41	93.88
bin.11	s__algicola (UID2847)	33	496	263	152	203	112	25	2	2	72.20	36.97	16.89
bin.1	k__Bacteria (UID203)	5449	104	58	34	53	17	0	0	0	65.83	4.80	5.88
bin.6	f__Rhodobacteraceae (UID3361)	46	654	332	248	403	3	0	0	0	57.96	0.75	0.00
bin.10	k__Archaea (UID2)	207	149	107	95	27	16	9	1	1	30.46	16.25	0.00
bin.7	k__Bacteria (UID203)	5449	104	58	55	33	10	6	0	0	28.68	5.02	14.29
bin.9	k__Bacteria (UID203)	5449	104	58	91	13	0	0	0	0	18.97	0.00	0.00
bin.2	root (UID1)	5656	56	24	45	0	6	0	4	1	9.38	14.47	12.50
bin.4	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00
bin.3	root (UID1)	5656	56	24	56	0	0	0	0	0	0.00	0.00	0.00

Strain heterogeneity indicates the proportion of the contamination that appears to be from the same or similar strains, giving an indication of the source of the contamination (i.e., highly similar or more divergent organisms).

If it is **high**, the majority of contamination is from very similar species,
if it is **low**, all the contamination is likely from other species.

CheckM

(M) > checkm tree <bin folder> <output folder>

The tree command places genome bins into a reference genome tree.

(R) > checkm tree_qa <output folder>

(optionally) indicate the number of phylogenetically informative marker genes found in each genome bin along with a taxonomic string indicating its approximate placement in the tree.

This because:

(I) genome bins with few phylogenetically marker genes may be removed in order to reduce the computational requirements of the following commands.

(II) if only genomes from a particular taxonomic group are of interest these can be moved to a new directory and analyzed separately.

(M) > checkm lineage_set <output folder> <marker file>

The lineage_set command creates a marker file indicating lineage-specific marker sets suitable for evaluating each genome.

CheckM

(M) > checkm analyze <marker file> <bin folder> <output folder>

The marker file from lineage_set is passed to the analyze command in order to identify marker genes and estimate the completeness and contamination of each genome bin.

(M) > checkm qa <marker file> <output folder>

qa command can be used to produce different tables summarizing the quality of each genome bin.

For convenience, the 4 mandatory steps can be executed using a **single-command**:
> checkm lineage_wf <bin folder> <output folder>

GTDB-Tk

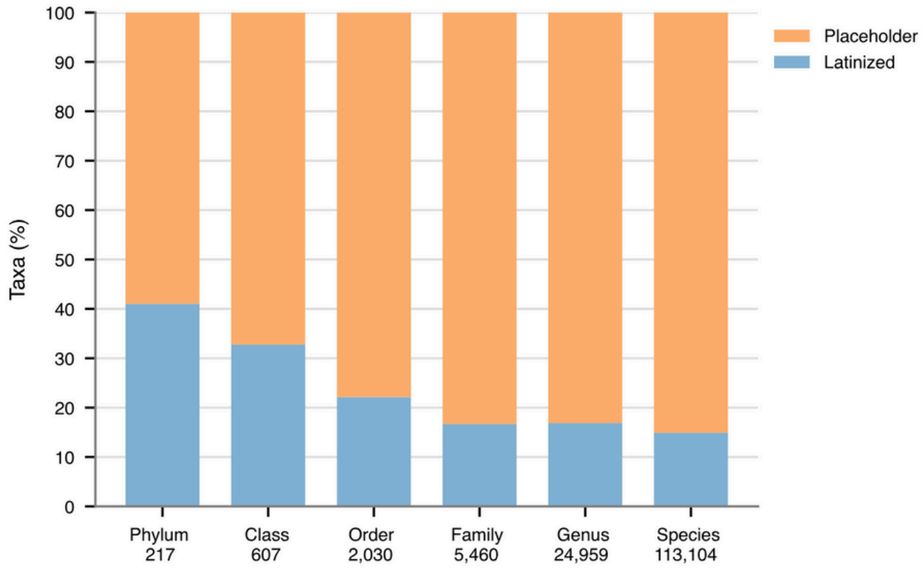
GTDB-Tk is a software toolkit for assigning taxonomic classifications to bacterial and archaeal genomes based on the Genome Database Taxonomy GTDB.

GTDB is a database with the important feature of including draft genomes of uncultured microorganisms obtained from metagenomes and single cells, ensuring improved genomic representation of the microbial world.

Taxon overview

GTDB R220 spans 596,859 genomes organized into 113,104 species clusters.

	Bacteria	Archaea	Total
Phylum	175	19	194
Class	538	64	602
Order	1,840	166	2,006
Family	4,870	564	5,434
Genus	23,112	1,847	24,959
Species	107,235	5,869	113,104



GTDB-Tk

The classify workflow consists of 4 steps:

ani_screen: compares user genomes against a Mash database of all GTDB representative genomes, then verify the best mash hits using skani.

identify: calls genes using Prodigal, and uses HMM models and the HMMER package to identify the 120 bacterial and 53 archaeal marker genes used for phylogenetic inference. MSA are obtained by aligning marker genes to their respective HMM model.

align: concatenates the aligned marker genes and filters the concatenated MSA to approximately 5,000 AAs.

classify: uses pplacer to find the maximum-likelihood placement of each genome in the GTDB-Tk reference tree. GTDB-Tk classifies each genome based on its placement in the reference tree, its relative evolutionary divergence, and/or average nucleotide identity (ANI) to reference genomes.

! Results can be impacted by a lack of marker genes or contamination.


jupyterhub

Prodigal

Prodigal is a protein-coding gene prediction software tool for bacterial and archaeal genomes.

It use unsupervised machine learning approach to identify and predict coding sequences (CDS) in prokaryotic genomes.

```
prodigal -i my.genome.fna -o my.genes -a my.proteins.faa
```

- my.genes is a gene coordinates file with the location of each gene
- my.proteins.faa consists of all the proteins from all the sequences in multiple FASTA format.

KofamKOALA - KEGG Orthology Search

K number assignment based on KO-dependent scoring criteria

KofamKOALA assigns K numbers to the user's sequence data by HMMER/HMMSEARCH against KOfam (a customized HMM database of KEGG Orthologs (KOs)).

The K number assignments facilitate the interpretation of the annotation results by linking the user's sequence data to the KEGG pathways and EC numbers.

```
1  >k141_88225_1 # 3 # 1532 # 1 #
   ID=1_1;partial=10;start_type=Edge;rbs_motif=None;rbs_spacer=None;gc_cont=0.576
2  PGFWTSVRLTLWIGFAATFLSLTLATCFCATAHARMTGRSAARWLAPLLAAPHTAVAIGL
3  AFVLAPSGWIARLLAPLVGWVRPPDLALINDPWGIALILGLMIKEIPFLLLVMLAALSQI
4  PVQQHMAAARGLGYRRSVVWVKIIAPQVWPLIRLPVLVVLAYSLSTVEMGIILGPSNPPV
5  FSVFVMRLFMAPDLAMILPASAGALMLAFLMGVASATLFVAERLVRYTGLWWIRAGGRSS
6  TVKPLLRIASLCVVALFVIGALAMVSLVFWSLAWRWPWPSMLPETWSFQPWRNAAGVLGS
7  ALGNTLLIAGASVFVSLALAILWLEGEDRKGRGRAGWGEVLIYLPLLLPQIAFLYGLNML
8  FLRLGISGGIGAVIWAQVLFVFPYVMIVLSDPWRALDPRLIRAAASLGAGPLRQLLTVKI
9  PLLLSPILIAAAIGTAVSVAQYLPTLFMGAGRISTLTTEAVTLSSGADRRITGVYASLQA
10 ILSFAAYAVAFMVPAFVFRNRAALKGAGQ*
```

```
1 k141_88225_1 K02011
2 k141_88225_1 K02063
3 k141_88225_1 K02053
4 k141_88225_1 K11070
5 k141_88225_1 K11082
```

KofamKOALA - KEGG Orthology Search

K number assignment based on KO-dependent scoring criteria

KofamKOALA assigns K numbers to the user's sequence data by HMMER/HMMSEARCH against KOfam (a customized HMM database of KEGG Orthologs (KOs)).

The K number assignments facilitate the interpretation of the annotation results by linking the user's sequence data to the KEGG pathways and EC numbers.

	gene name	KO	thrshld	score	E-value	KO definition
*	k141_88225_1	K05778	234.33	582.0	1.5e-175	putative thiamine transport system permease protein
	k141_88225_1	K02011	343.43	127.8	4.4e-38	iron(III) transport system permease protein
	k141_88225_1	K02063	334.97	118.5	3.5e-35	thiamine transport system permease protein
	k141_88225_1	K02053	247.40	111.3	5.1e-33	putative spermidine/putrescine transport system permease protein
	k141_88225_1	K11070	275.63	76.9	1.3e-22	spermidine/putrescine transport system permease protein
	k141_88225_1	K11082	264.70	55.2	6.1e-16	2-aminoethylphosphonate transport system permease protein
	k141_88225_1	K11083	292.13	51.6	7e-15	2-aminoethylphosphonate transport system permease protein

KofamKOALA - KEGG Orthology Search

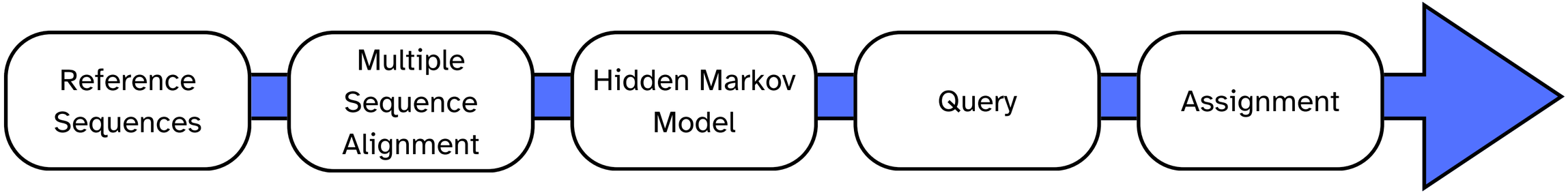
K number assignment based on KO-dependent scoring criteria

```
exec_annotation --profile=profiles --ko-list=ko_list -o ko.txt proteins.faa
```

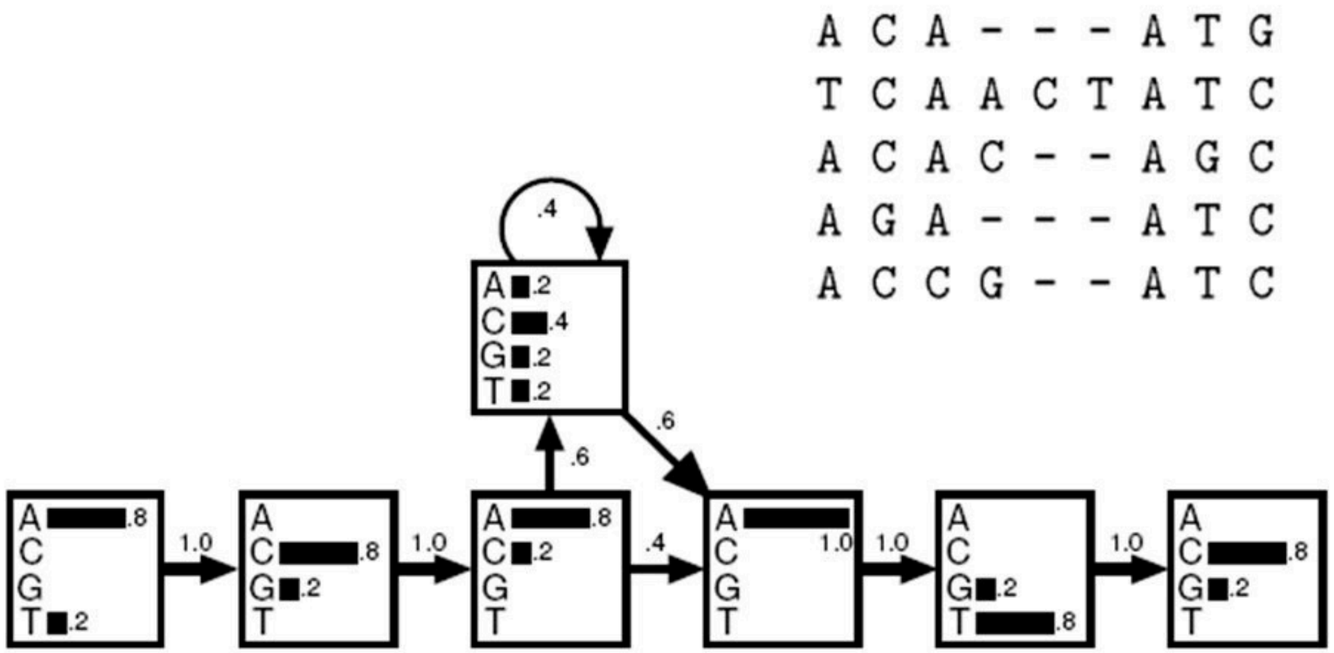
- profiles contains HMM models for every KO in KEGG
- KO_list contains score values for every KO in KEGG

KofamKOALA aligns all the query genes to the pre-computed models to find the best matches using the score values as threshold for significance.

Sequence matches with scores above the thresholds are considered more reliable than other matches and thus highlighted with ‘*’ marks in the output of the tool.



Markov Models



Consensus sequence: ACACATC

HMM is a probabilistic model that represents a sequence of observations as the output of a system with hidden (unobservable) states.

In the context of protein sequences:

- The observable symbols are the amino acids in the protein sequence.
- The hidden states represent underlying structural or functional elements of the protein.

once an HMM is established based on the training sequences, it can be used to determine how well an unknown sequence matches the model

KEGG MAPPER

KEGG Mapper is a collection of tools for KEGG mapping.
There are five KEGG Mapper tools as summarized below.

Reconstruct is the basic mapping tool used for linking KO annotation (K number assignment) data to KEGG pathway maps, BRITE hierarchies and tables, and KEGG modules.

Search is the traditional tool for searching mapped objects in the user's dataset and mark them in red.

Color is another traditional tool for searching mapped objects in the user's dataset and mark them in any combination of background and foreground colors. This tool now applies only to KEGG pathway maps. Use the Join tool for coloring of Brite hierarchies.

Join is a tool to combine a Brite hierarchy file and a binary relation file, effectively adding a new column to the hierarchy file.

MWsearch is a variant of the Search tool performing conversion of mass spectropy data, either as a set of molecular masses or molecular formulas, to a set of numbers.

KYOTO ENCYCLOPEDIA OF GENES AND GENOMES

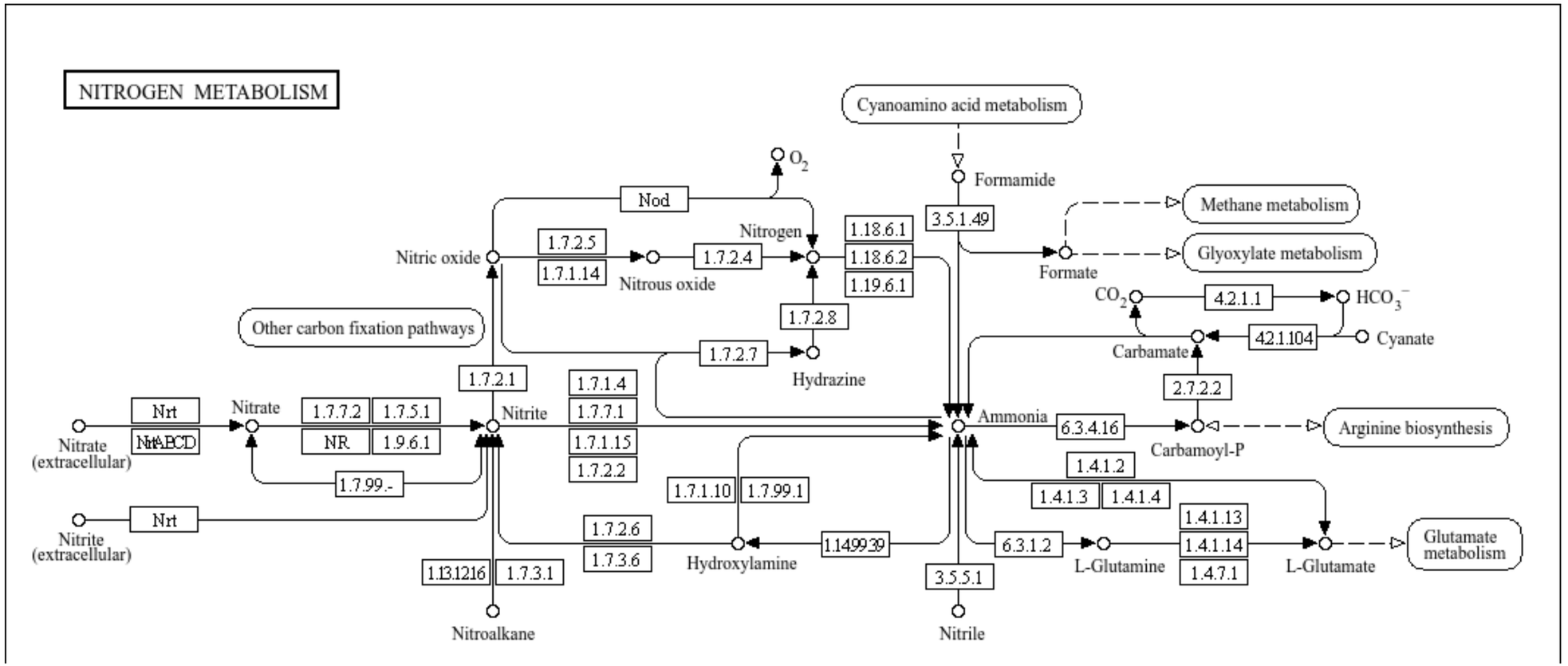
KEGG is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism and the ecosystem, from molecular-level information, especially large-scale molecular datasets generated by genome sequencing and other high-throughput experimental technologies.

Systems information	
KEGG PATHWAY	572 (1,223,443)
KEGG BRITE	201 (398,073)
KEGG MODULE	486
	48

Data-oriented entry points

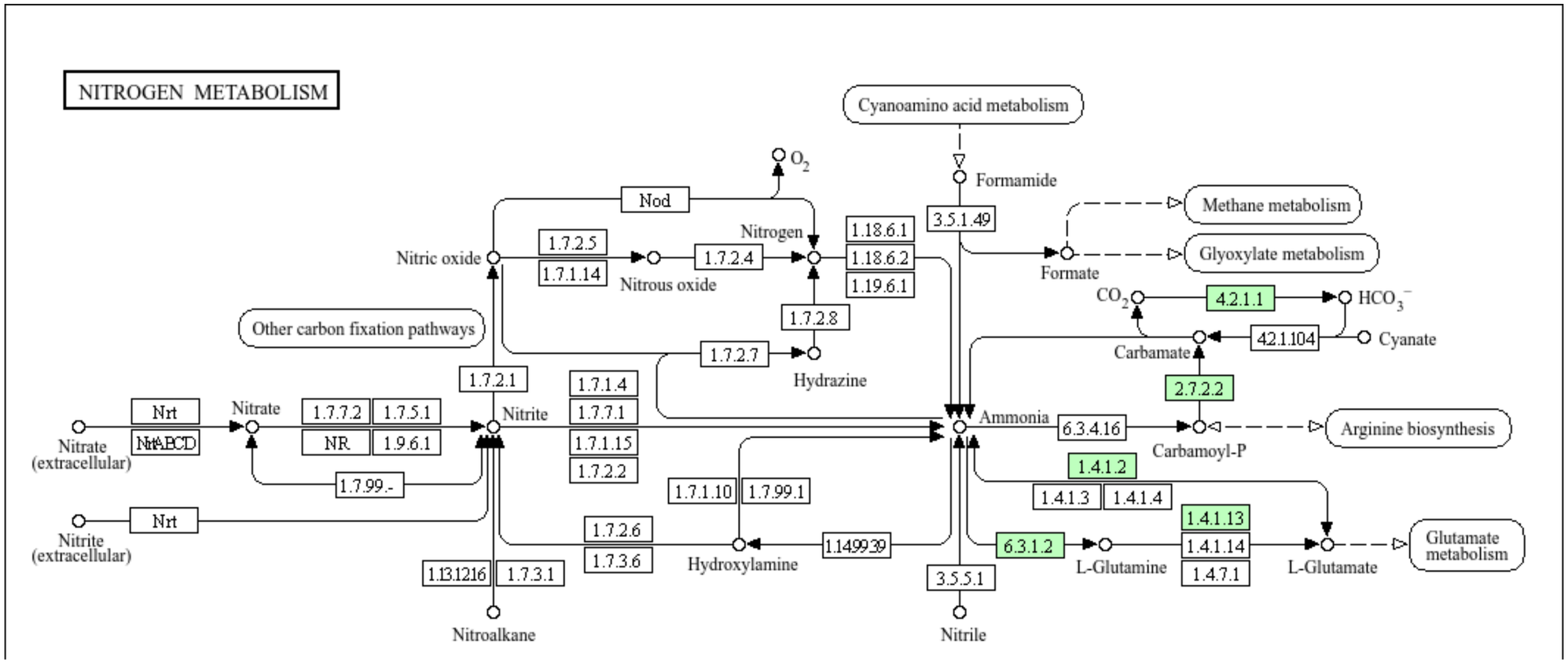
KEGG PATHWAY	KEGG pathway maps
KEGG BRITE	BRITE hierarchies and tables
KEGG MODULE	KEGG modules
KEGG ORTHOLOGY	KO functional orthologs
KEGG GENES	Genes and proteins [Annotation]
KEGG GENOME	Genomes [KEGG Virus | Syntax]
KEGG COMPOUND	Small molecules
KEGG GLYCAN	Glycans
KEGG REACTION	Biochemical reactions [RModule]
KEGG ENZYME	Enzyme nomenclature
KEGG NETWORK	Disease-related network variations
KEGG DISEASE	Human diseases
KEGG DRUG	Drugs [New drug approvals]
KEGG MEDICUS	Health information resource

KYOTO ENCYCLOPEDIA OF GENES AND GENOMES

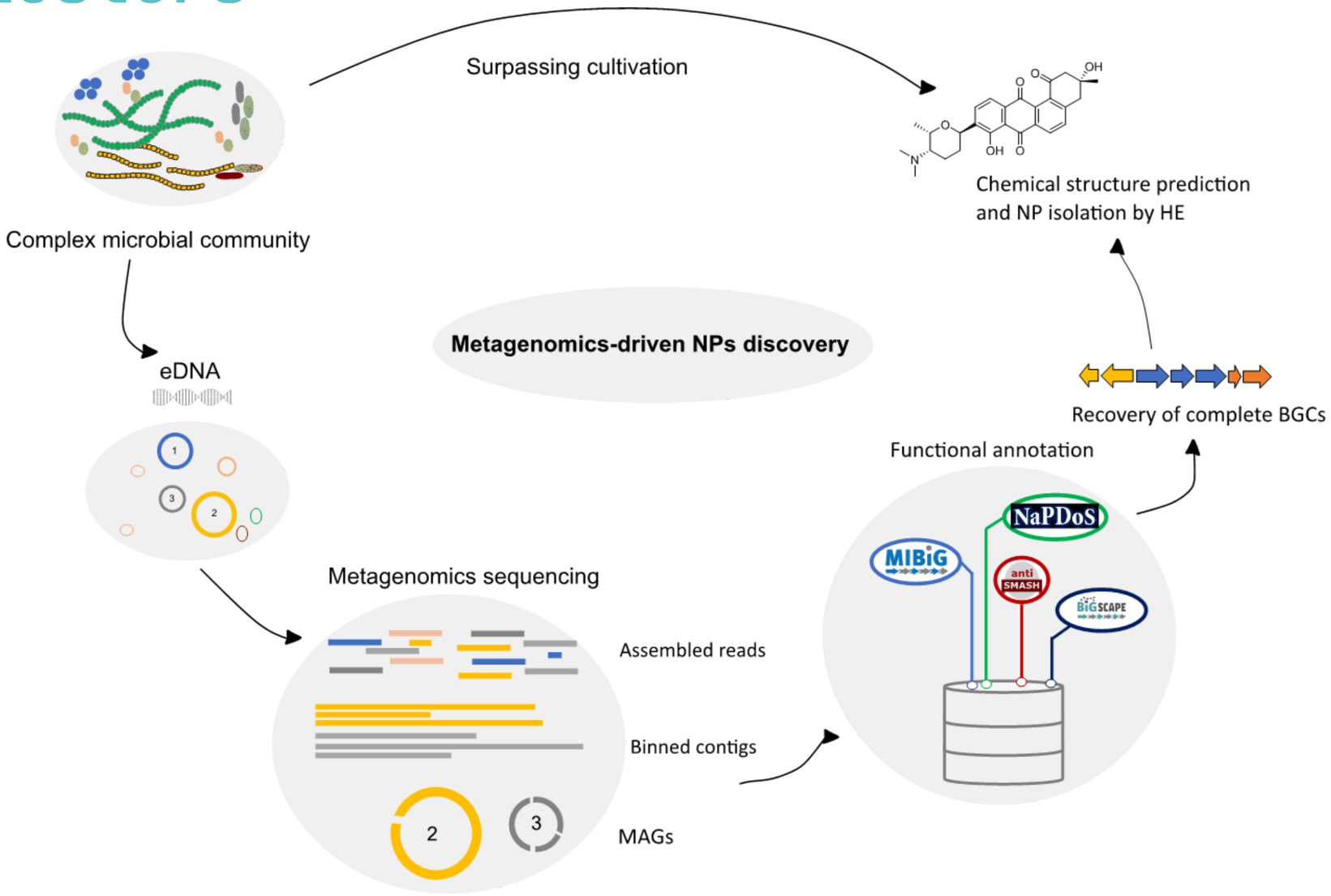


KEGG MAPPER

Genes present in our MAG are mapped on known and curated pathways present in the database



Functional annotation of biosynthetic gene clusters



Identification of biosynthetic gene clusters

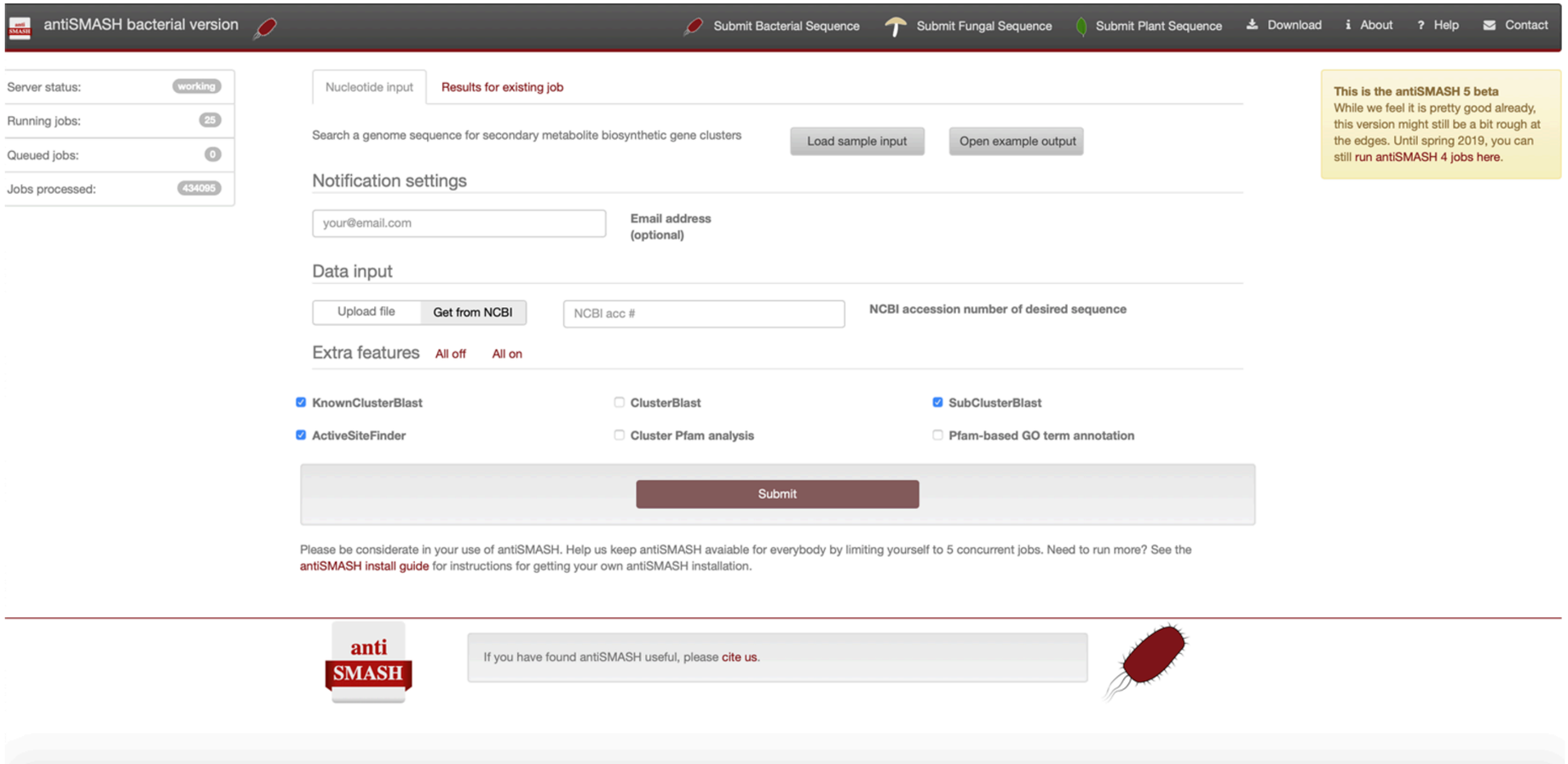
antiSMASH (**anti**biotics & **S**econdary **M**etabolite **A**nalysis **Sh**ell) allows the detection of clusters of co-occurring biosynthesis genes in genomes, called **B**iosynthetic **G**ene **C**lusters (**BGCs**). BGCs often contain all the genes required for the biosynthesis of one or more **N**atural **P**roducts (**NPs**), also known as **specialized** or **secondary metabolites**.

- allows the rapid genome-wide **identification**, **annotation** and analysis of secondary metabolite biosynthetic gene clusters in bacterial and fungal (meta)genomes.

-based on profile **hidden Markov models (HMMs**) of genes that are specific for certain types of gene clusters, antiSMASH is able to **accurately identify the gene clusters encoding secondary metabolites** of all known broad chemical classes. antiSMASH not only **detects the gene clusters, but also offers detailed sequence analysis**.

How to Use antiSMASH - Public Web Version

https://antismash.secondarymetabolites.org/#!/start



How to Use antiSMASH - Local Installation

```
antismash MAG_example.fa --cb-general --cb-subclusters --cb-knownclusters --genefinding-tool prodigal --output-dir MAG_example_antismash_output
```

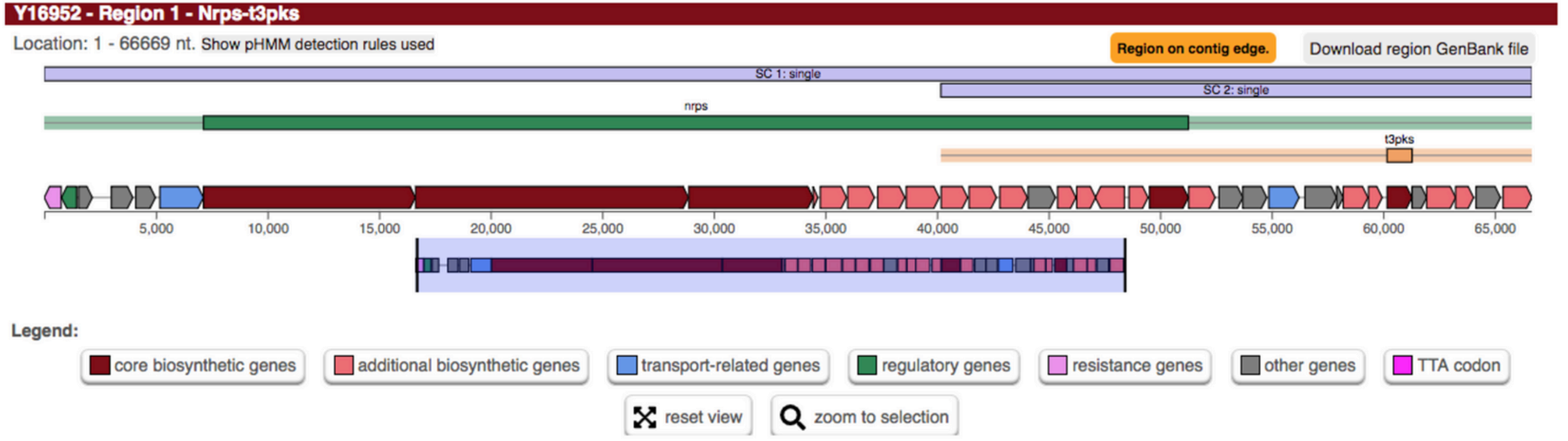
Subcluster Blast analysis: the identified clusters are searched against a database containing operons involved in the biosynthesis of common secondary metabolite building blocks (e.g. the biosynthesis of non-proteinogenic amino acids).

KnownClusterBlast analysis: the identified clusters are searched against the MIBiG repository.

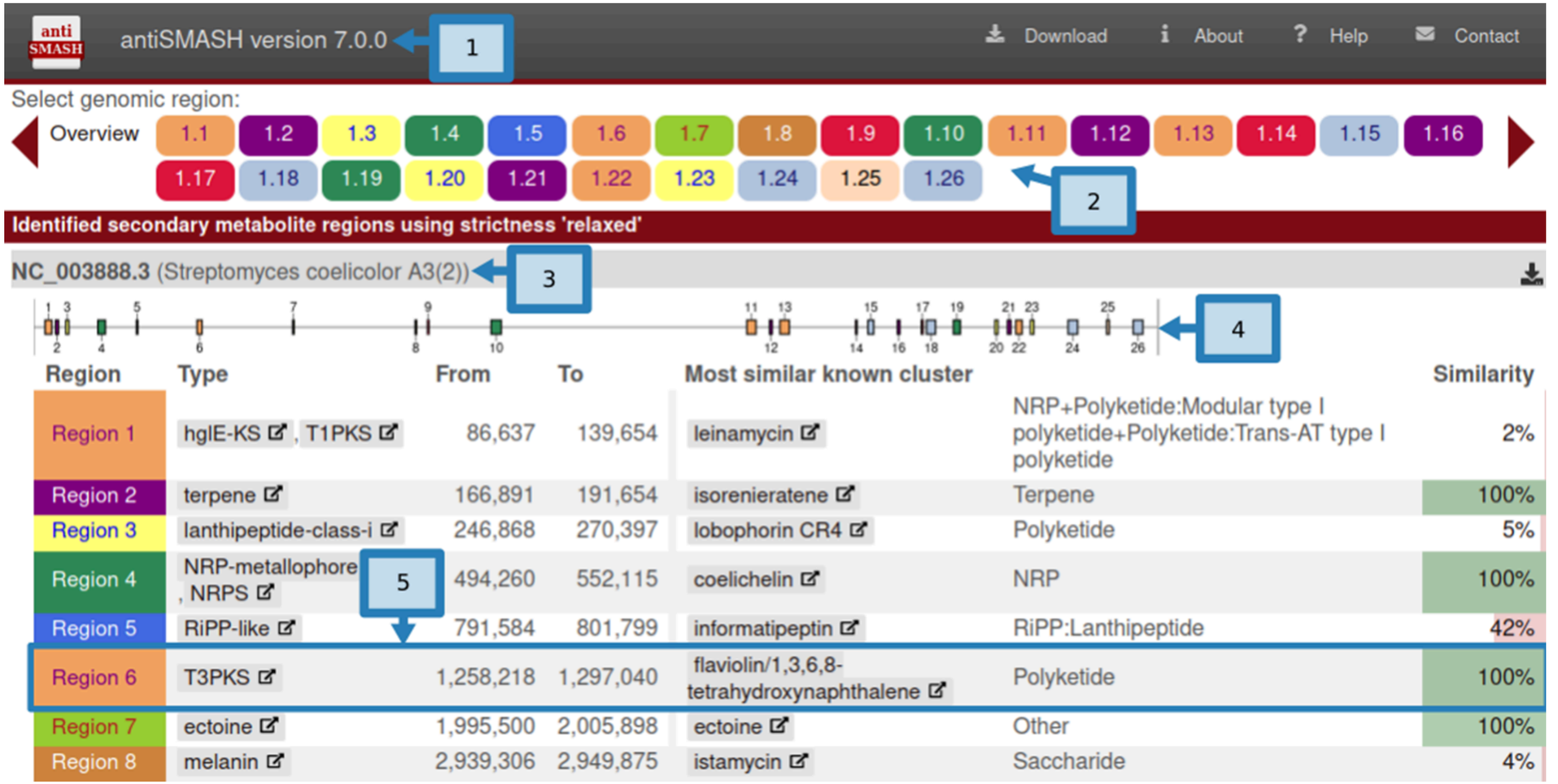
ClusterBlast analysis: the identified clusters are searched against a comprehensive gene cluster database and similar clusters are identified.

How are antiSMASH regions/gene clusters defined?

In the first step, all gene products of the analyzed sequence are searched against a database of **highly conserved enzyme HMM profiles** (core-enzymes), which are indicative of a specific BGC type. In a second step, pre-defined cluster rules are employed to define individual ***protoclusters*** encoded in the region. These make up a *core*, which is extended by its *neighbourhood*, which constiutes of genes encoded up- and downstream of the *core*.

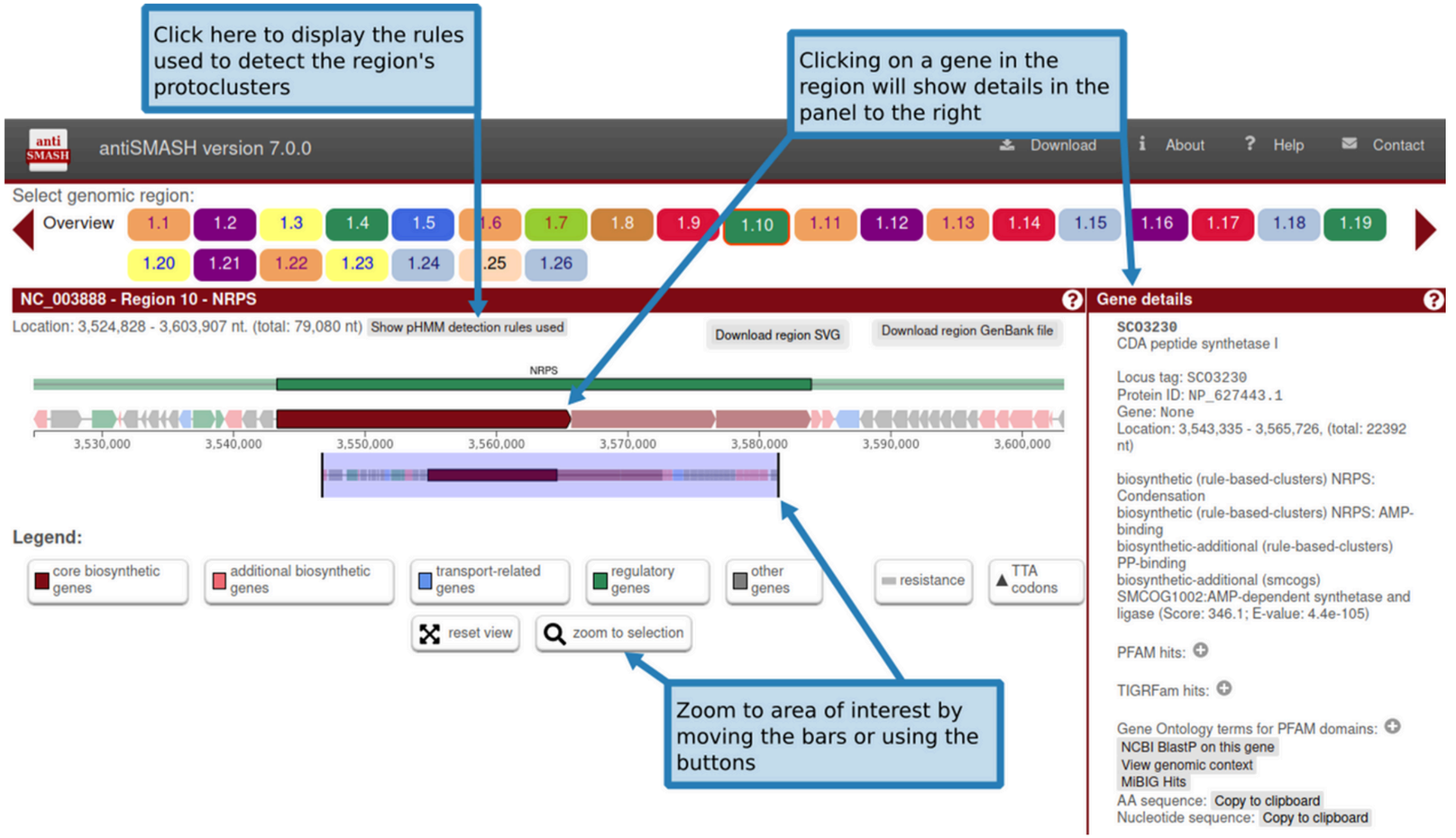


Results Overview page

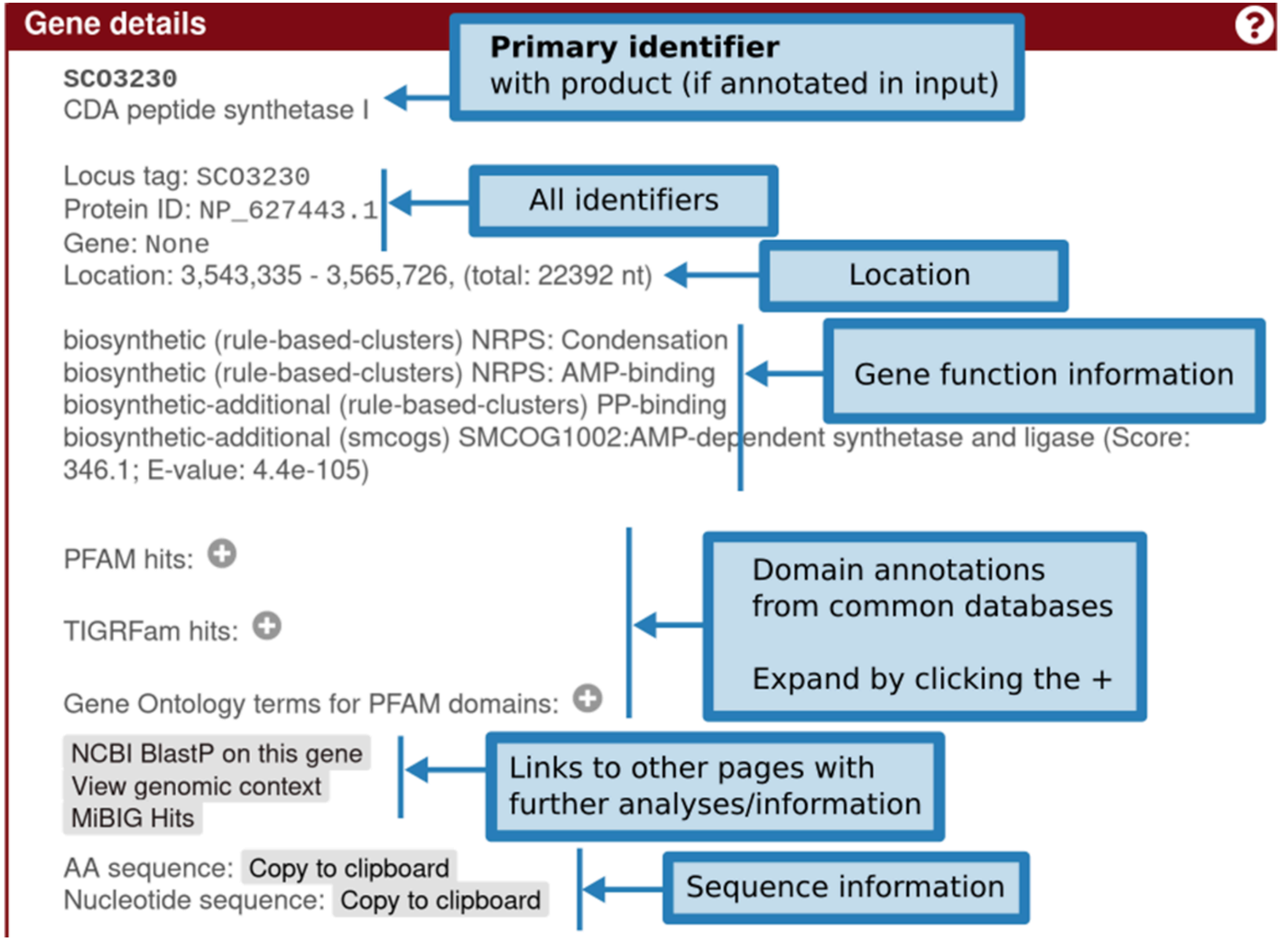


Region	Type	From	To	Most similar known cluster		Similarity
Region 1	hglE-KS, T1PKS	86,637	139,654	leinamycin	NRP+Polyketide:Modular type I polyketide+Polyketide:Trans-AT type I polyketide	2%
Region 2	terpene	166,891	191,654	isorenieratene	Terpene	100%
Region 3	lanthipeptide-class-i	246,868	270,397	lobophorin CR4	Polyketide	5%
Region 4	NRP-metallophore, NRPS	494,260	552,115	coelichelin	NRP	100%
Region 5	RiPP-like	791,584	801,799	informatipeptin	RiPP:Lanthipeptide	42%
Region 6	T3PKS	1,258,218	1,297,040	flaviolin/1,3,6,8-tetrahydroxynaphthalene	Polyketide	100%
Region 7	ectoine	1,995,500	2,005,898	ectoine	Other	100%
Region 8	melanin	2,939,306	2,949,875	istamycin	Saccharide	4%

Region results page



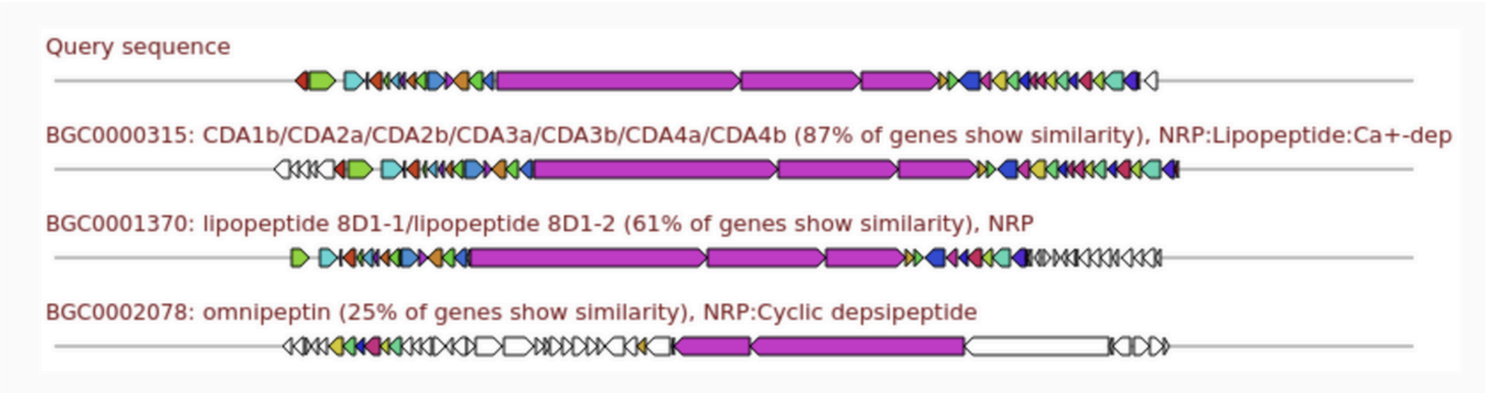
Gene details



Interpreting results – ClusterBlast

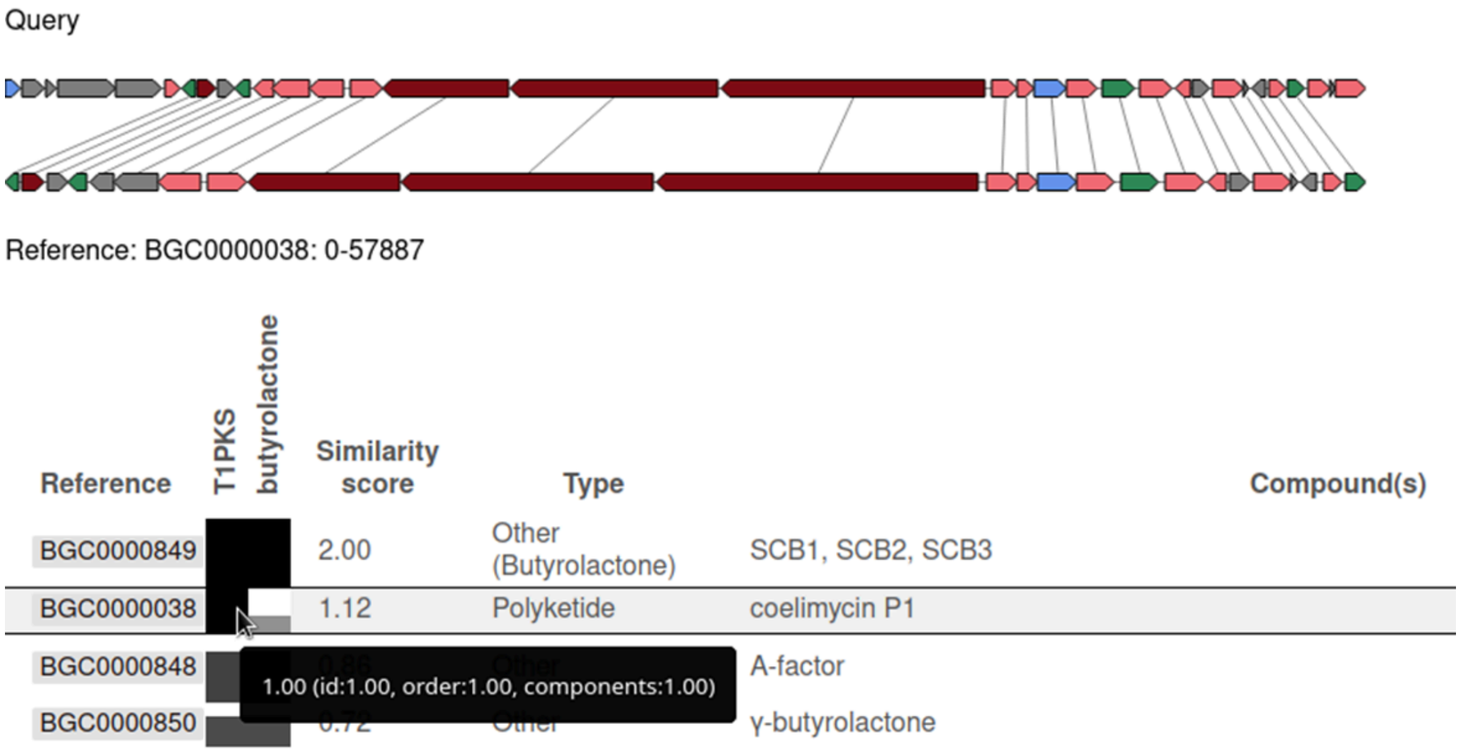
There are three modes of the ClusterBlast alogorithm, all of which use the same algorithm and results visualisation, varying only in reference dataset:

- **ClusterBlast**: shows regions from the antiSMASH Database that are similar to the current region
- **KnownClusterBlast**: shows clusters from MIBiG that are similar to the current region
- **SubClusterBlast**: shows sub-cluster units related to the current region



Interpreting results – ClusterCompare

ClusterCompare is an algorithm for **cluster comparison**, taking into account presence of biosynthetic profiles, NRPS/PKS module counts and layouts, and gene functions, along with the sequence identity and synteny used by the clusterblast module.
The range of scores for any pairing is between 0 and 1, with 1 being a theoretical perfect score.



BGCs similarity clustering

BiG-SCAPE (Biosynthetic Gene Similarity Clustering and Prospecting Engine) is a software package that constructs **sequence similarity networks** of Biosynthetic Gene Clusters (BGCs) and groups them into **Gene Cluster Families** (GCFs). BiG-SCAPE does this by rapidly calculating a distance matrix between gene clusters based on a comparison of their protein domain content, order, copy number and sequence identity.
As input, BiG-SCAPE takes GenBank files from the output of **antiSMASH** with BGC predictions, as well as reference BGCs from the **MIBiG repository**. As output, BiG-SCAPE generates tab-delimited output files, as well as a rich **HTML** visualization.

How to Use BiG-SCAPE - Local Installation

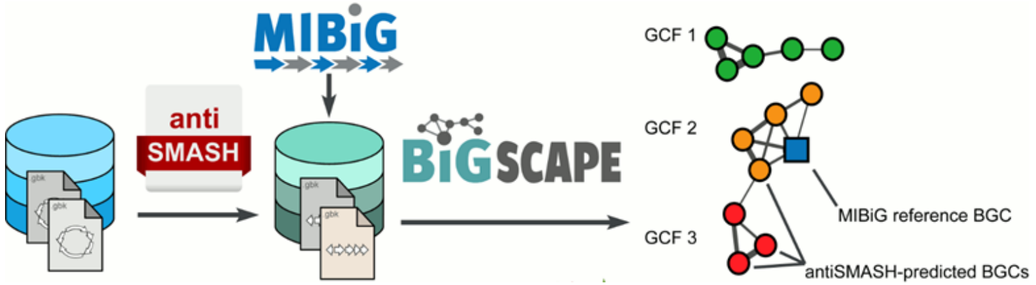

```
python bigscape.py -i  antismash_gbk_files  --mix --mibig --include_singletons --cutoffs 0.3, 0.7 -o bigscape_output
```

MIBiG Minimum Information about a Biosynthetic Gene cluster

General statistics database contains

Total Secondary Metabolite Clusters:			**2502**
Minimal entries:	**1574**	**Non-minimal entries:**	**928**
Complete entries:	**533**	**Incomplete entries:**	**33**
Retired entries:	**218**	**Pending entries:**	**1**

How to Use BiG-SCAPE - Local Installation




```
python bigscape.py -i  antismash_gbk_files  --mix --mibig --
include_singletons --cutoffs 0.3, 0.7 -o bigscape_output
```

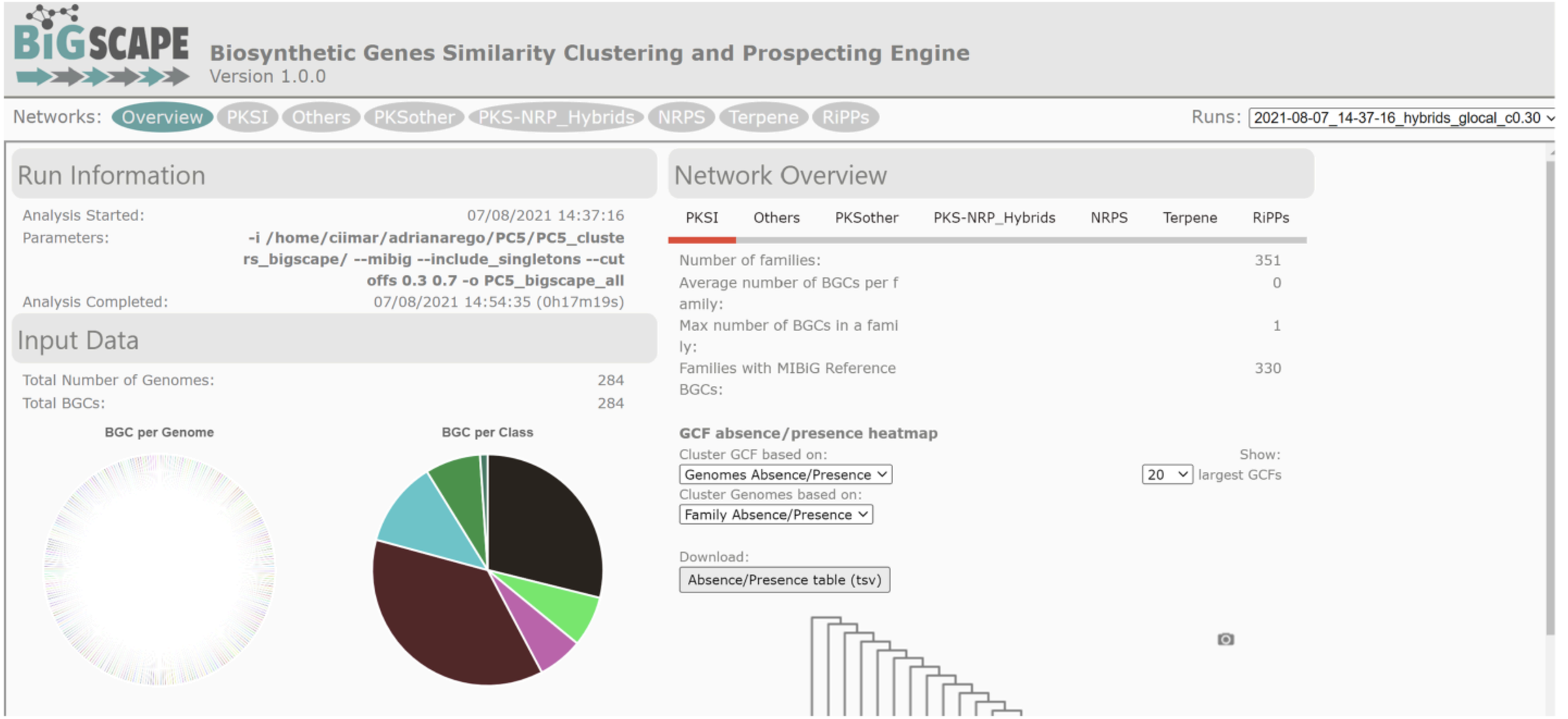
--mix - By default, BiG-SCAPE separates the analysis according to the BGC product and will create network directories for each class. Adding the –mix option will include an analysis mixing all classes.

--mibig - BGCs from the MIBiG database are included in the analysis.

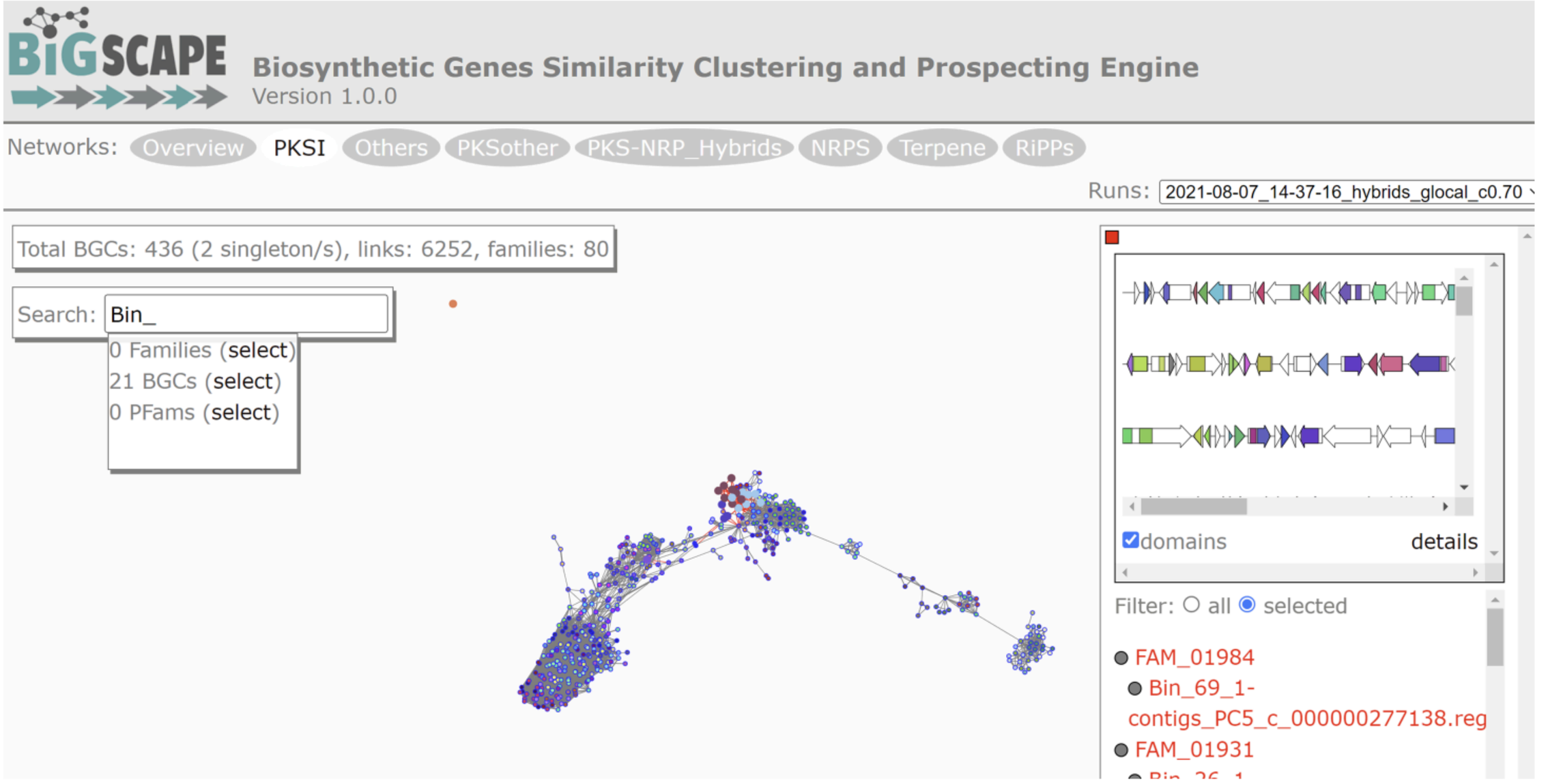
--inlcude_singletons - This will include BGCs that don't have a distance lower than the cutoff distance specified.

--cutoffs - Generate networks using multiple raw distance cutoff values. Automatic clustering of Gene Cluster Families will be done using each cutoff. Example – 0.3 cutoff, only with GCFs with at least 70% of similarity will be connected in the SSN.

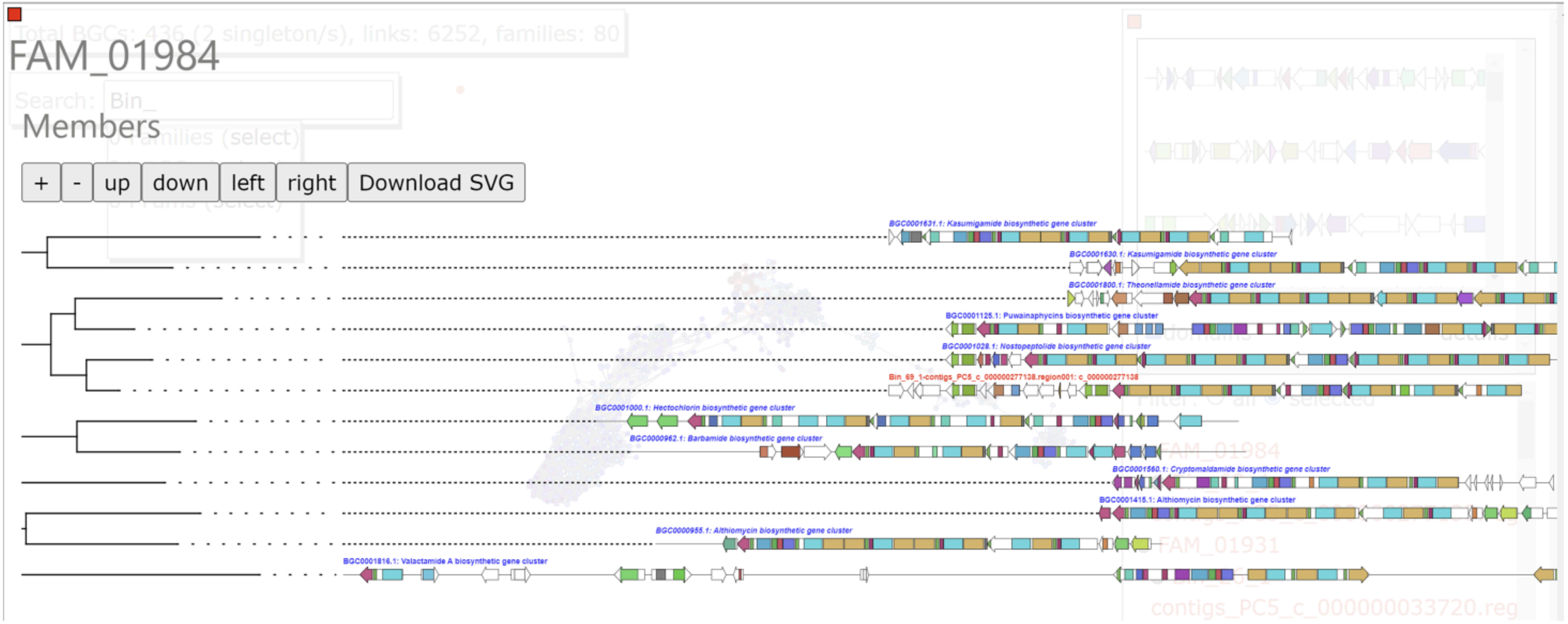
Results Overview page



Results overview page by BGC class - visualization of gene cluster families



Results overview page - selected gene cluster family



Hands-On Metagenomics

Adriana Rego, Nicola Gambardella

Thank You for your Attention!

ciimar

BIOPORTUGAL S.A.
Químico, Farmacêutica

SPECO