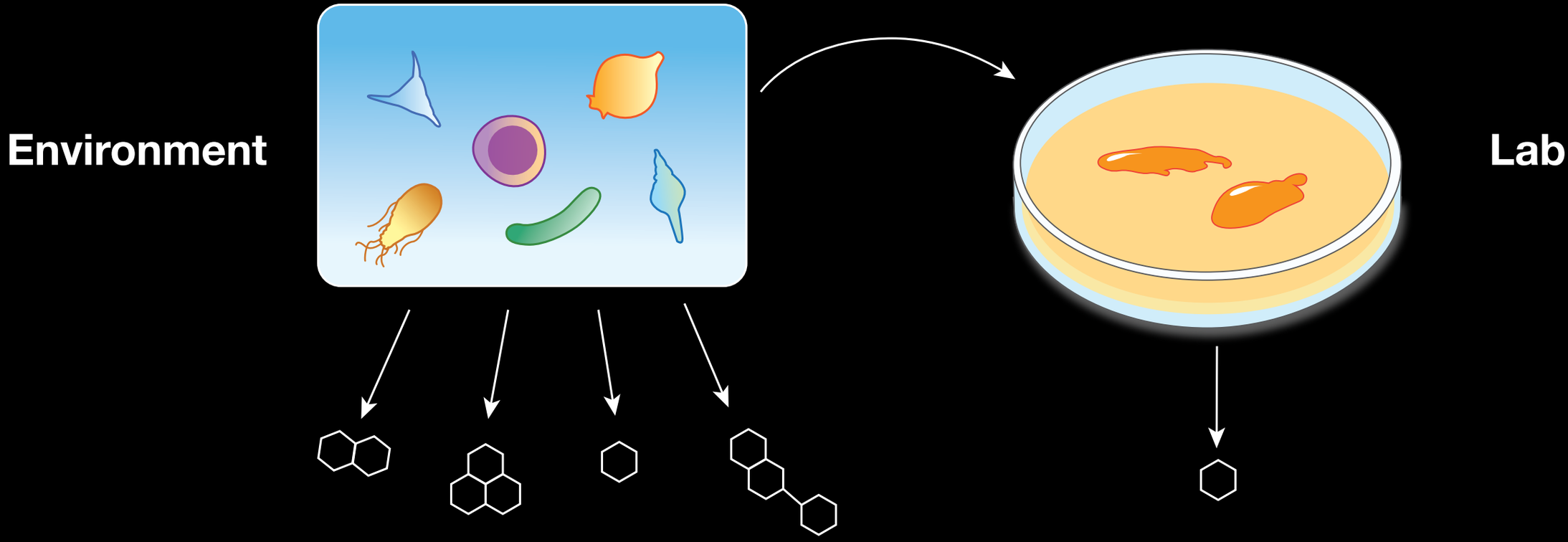
Automated binning from complex shotgun metagenomes

Jason C. Kwan
Division of Pharmaceutical Sciences
ComBEE Monthly Meeting
November 16 2018

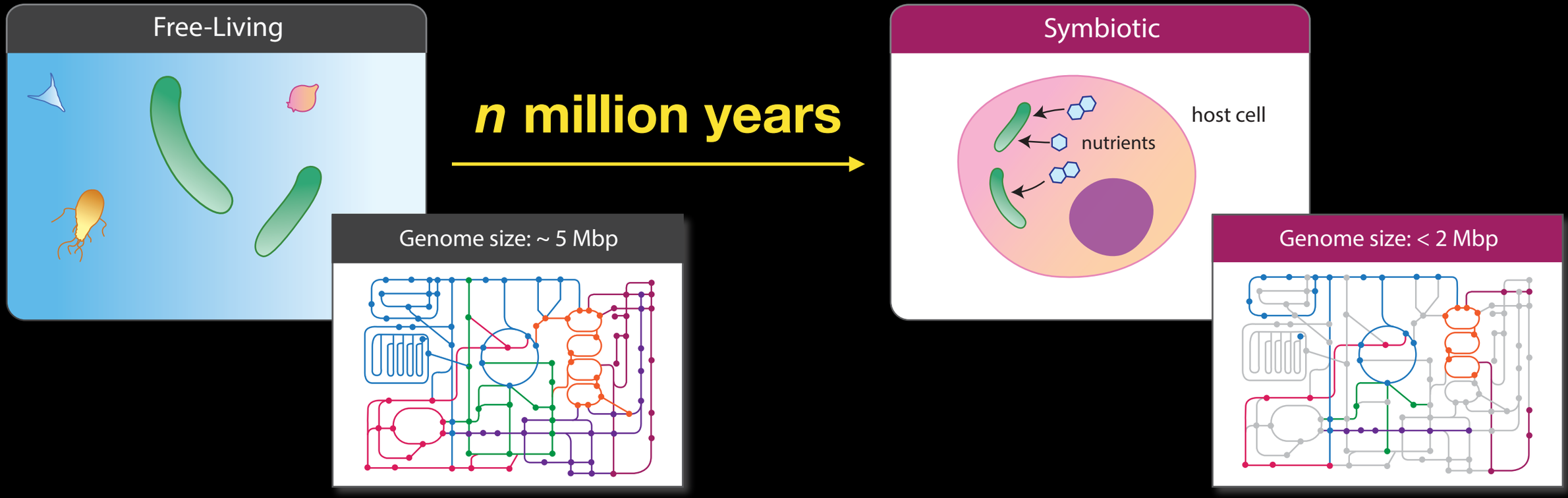
@kwan_lab

Most microbes have never been cultured



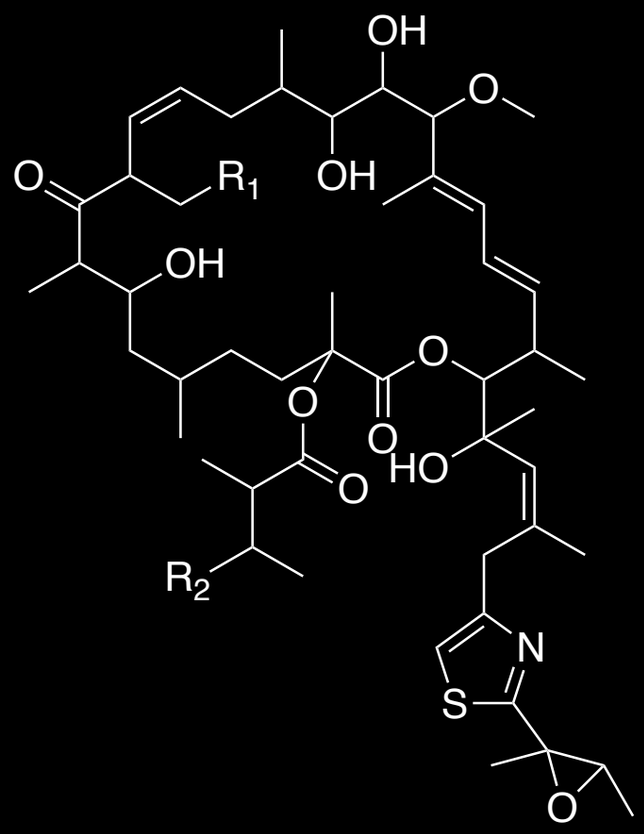
See video abstract of *mSystems* 2017, *2*, e00096-17

Symbionts can provide chemical defenses, but can be dependent on the host

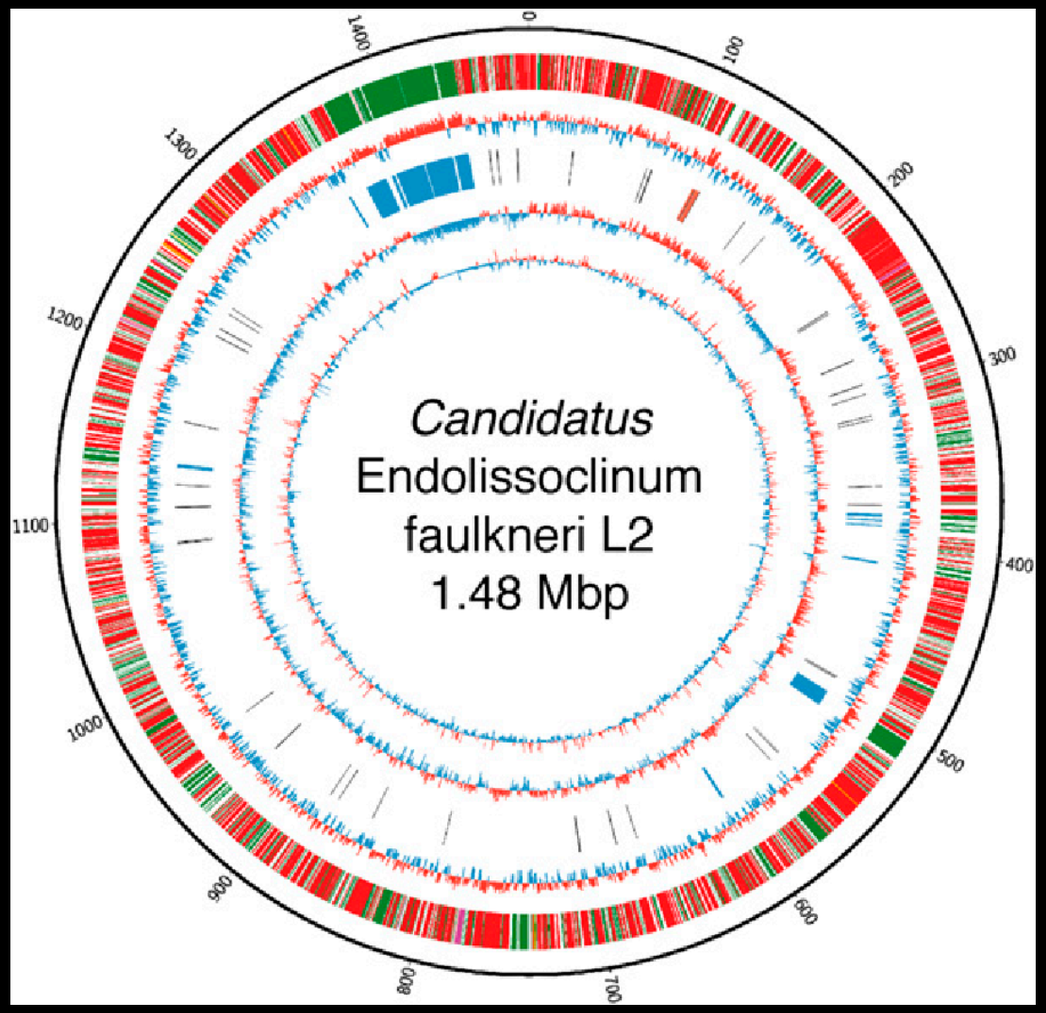


See video abstract of *mSystems* 2017, *2*, e00096-17

BGCs can be fragmented in symbiont genomes



Patellazoles

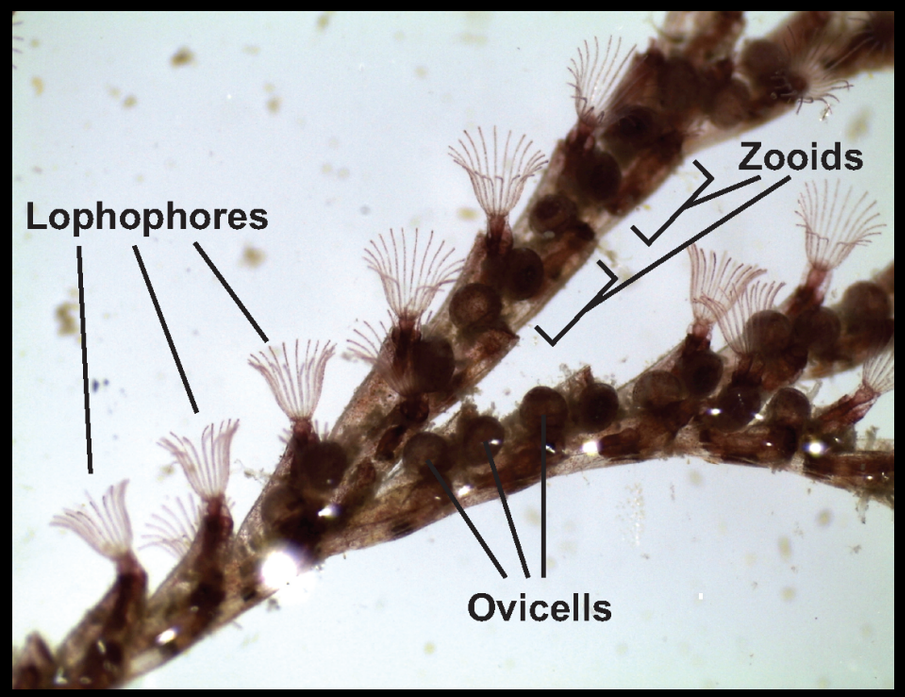


***PNAS* 2012, *109*, 20655–20660**

For our model of why/how this happens, see:
mSystems 2017**, *2*, e00096-17**

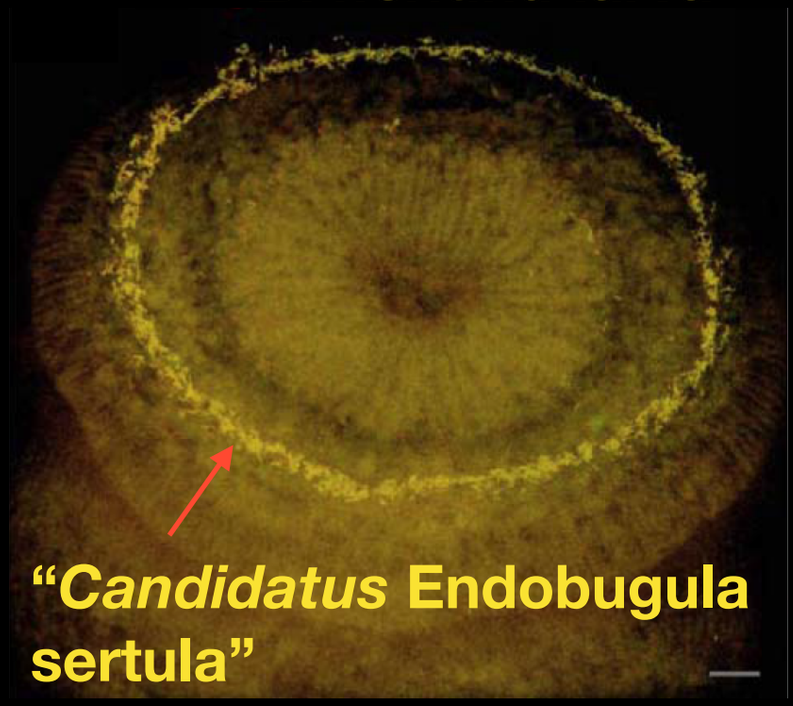
Bryostatins

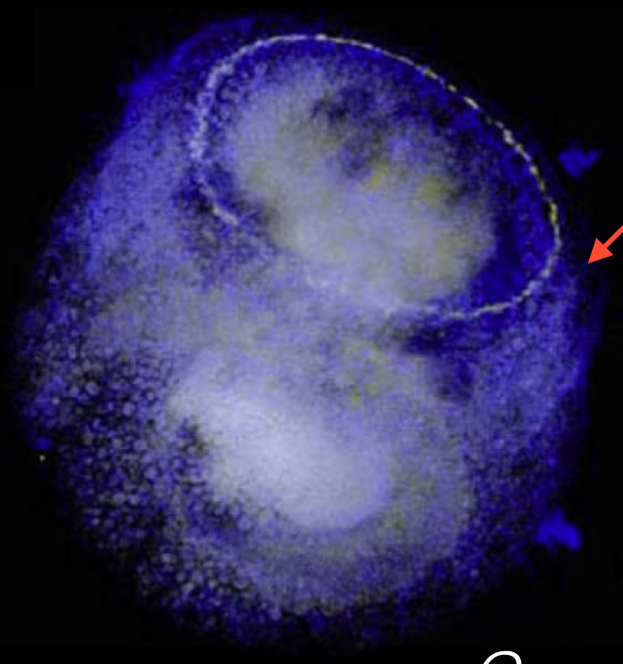
AEM 2016**, *82*, 6573–6583**



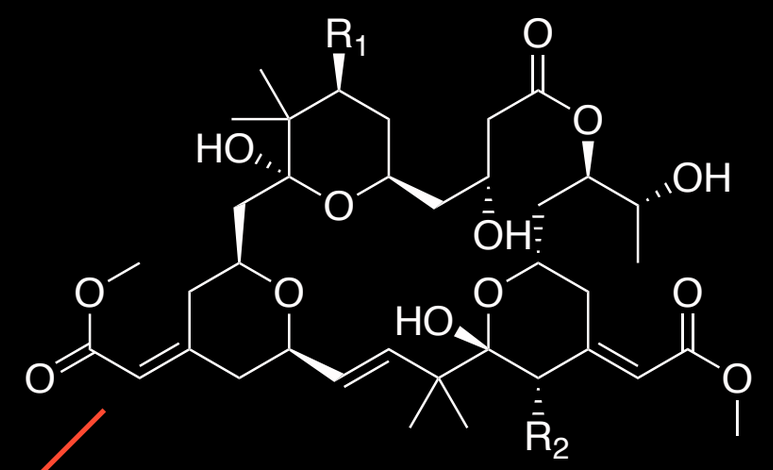
***Bugula neritina* (bryozoan)**

***B. neritina* larva**





Bryostatins



Potent PKC activators
Clinical trials for cancer, HIV and Alzheimer's

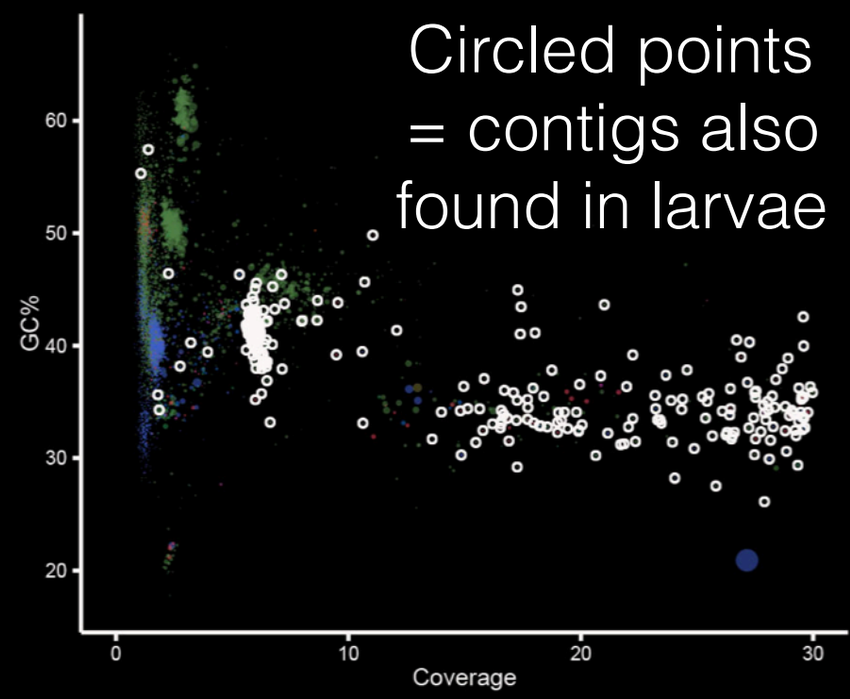
Curr. Opin. Biotechnol. **2010**, *21*, 834–842

B. neritina metagenome

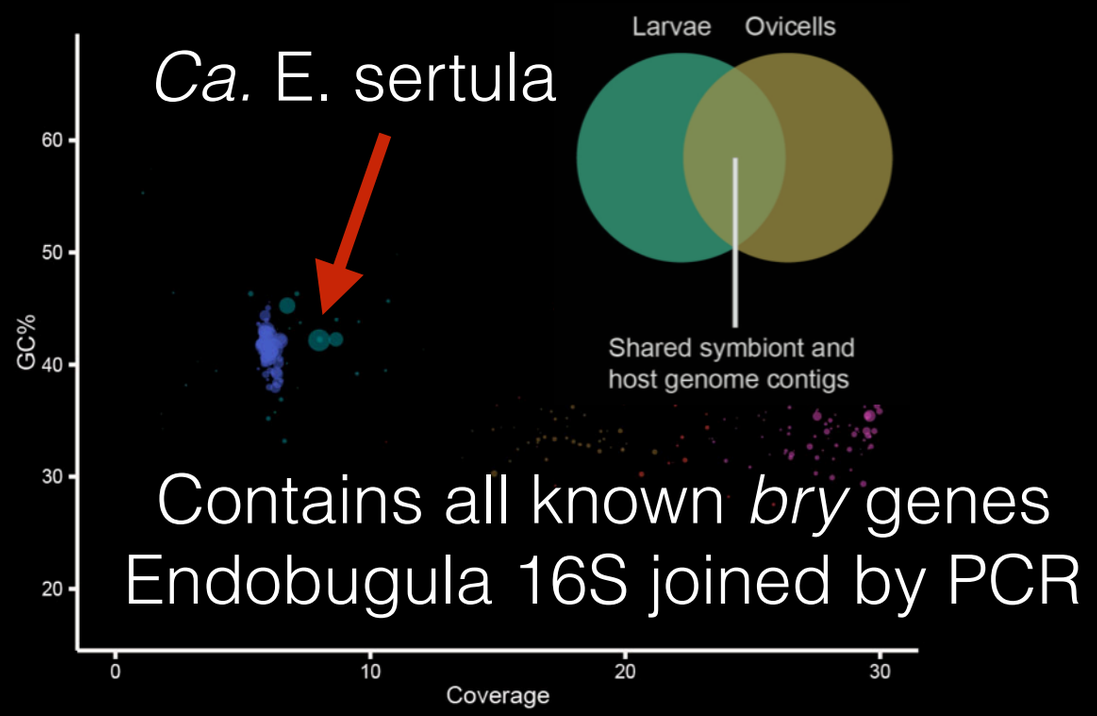
Ca. E. sertula contigs identified by comparing adult (**sample AB1**) and larval metagenomes

7 additional genomes separated using coverage, GC% and inferred taxonomy

Adult metagenome

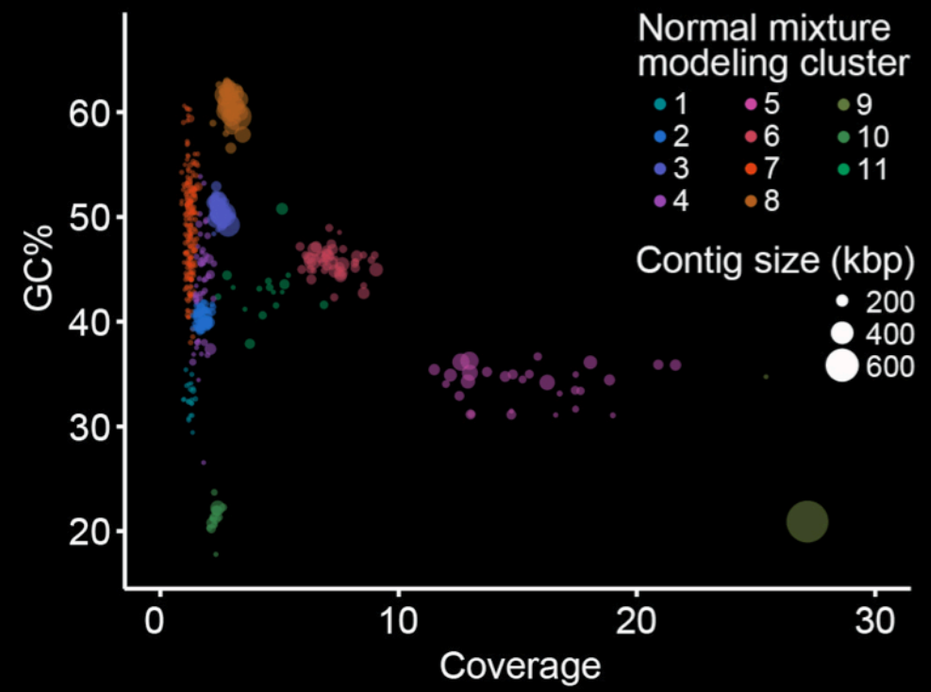


Contigs found in larvae and adult



Contains all known *bry* genes
Endobugula 16S joined by PCR

Contigs only found in adult



Microbial genomes found in *B. neritina*

Genome	Contigs	Size (Mbp)	N50 (kbp)	Completeness (%)	Purity (%)	16S identity
AB1_bin11	315	1.59	4.64	47.5	98.6	-
AB1_bin12	207	7.44	69.3	100	98.6	-
AB1_div	108	1.91	27.3	96.4	98.6	90%*‡
AB1_endobugula	112	3.34	49.5	100	99.3	99%
AB1_endozoicomonas	272	4.05	20.8	82.0	89.2	97%
AB1_lowgc	1	0.593	59.3	23.0	100	65%*‡†
AB1_lowcov	8	0.436	83.8	48.9	100	85%*‡
AB1_phaeo	106	4.67	124.1	98.6	97.8	92%

Identity cutoffs:
Rev. Microbiol.
2014, *12*, 635–645

New genus

New family

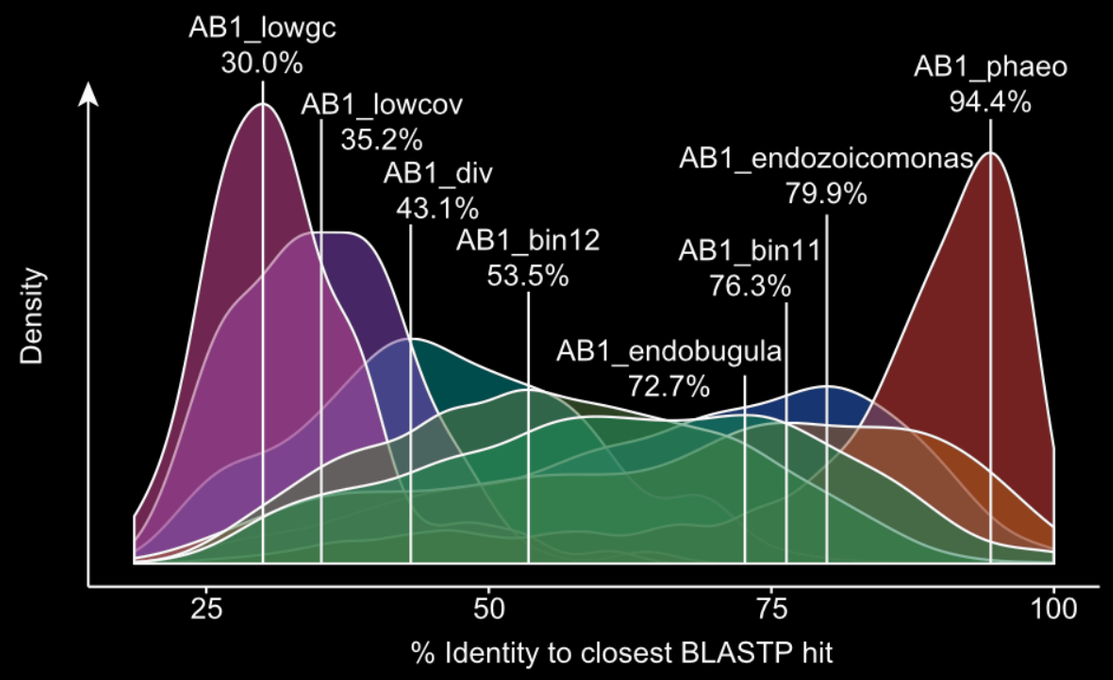
Candidate division NPL-UPA2

New species

New phylum

New family

New genus



Microbial dark matter:

* not detectable with universal 16S primers 27F/1492R

† not detectable with 16S amplicon primers S-D-Bact-0341-b-2-17/S-D-Bact-0785-a-A-21

‡ No binding site for eubacterial probe EUB338

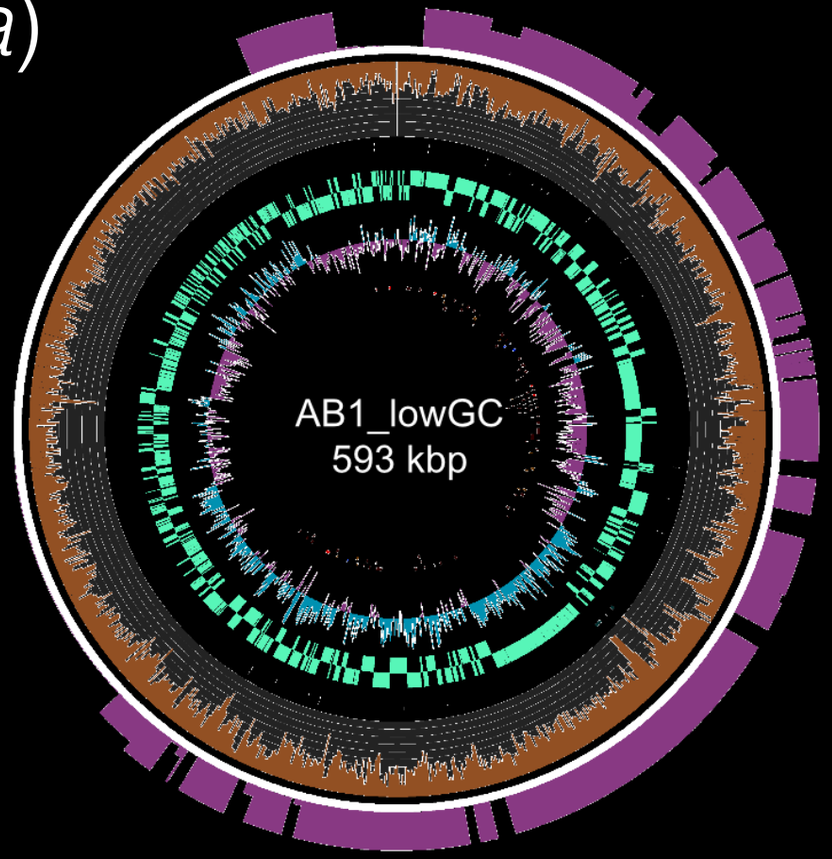
Dynamics of *B. neritina* metagenome

Some species found in AB1 also found in other individuals

- **AB1_div** (only found in *B. neritina*)
- AB1_endozoicomonas
- AB1_phaeo

Divergent species unique to AB1

- AB1_lowgc
- AB1_lowcov



Amplicon sequencing	*Bugula neritina*																Controls		
Animal	AB1	MHD	AB9		AB12		AB14		AB17		AB20	AB23	AB25		IMS1		MHDW1	MHDW4	Control
Tissue	Ovicells	Larvae	Ovicells	Larvae	Ovicells	Larvae	Ovicells	Larvae	Ovicells	Larvae	Ovicells	Ovicells	Ovicells	Larvae	Ovicells	Larvae	Whole	Whole	Seawate
AB1_div	3.080	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.001	0.008	0.000	0.000	0.130	0.001	0.000	0.000	0.000
AB1_endobugula	2.423	5.584	3.096	0.384	1.073	3.335	8.963	36.814	3.927	11.642	14.518	8.102	0.805	54.542	12.669	0.032	0.000	0.000	0.000
AB1_endozoicomonas	12.143	0.055	3.066	0.200	0.423	0.328	2.051	0.018	12.343	0.009	10.998	12.147	0.788	0.237	34.378	0.027	0.007	0.205	0.004
AB1_lowcov	0.120	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AB1_phaeo	8.906	0.149	0.062	0.001	0.011	0.000	0.020	0.000	0.014	0.000	0.033	0.055	0.005	0.000	0.013	0.001	0.301	0.344	0.022
n	58,353	32,879	82,096	76,086	105,995	73,104	71,034	27,199	73,643	74,541	216,383	112,986	305,946	40,431	89,112	413,059	13,950	13,669	26,974
Specific PCR																			
AB1_Endobugula	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-
AB1_Endozoicomonas	+	-	+	+	+	+	+	-	+	-	+	+	+	-	+	+	+	+	+
AB1_lowgc	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Sci. Rep., **2016**, *6*, 34362

Autometa: automated binning pipeline

Secret sauce:

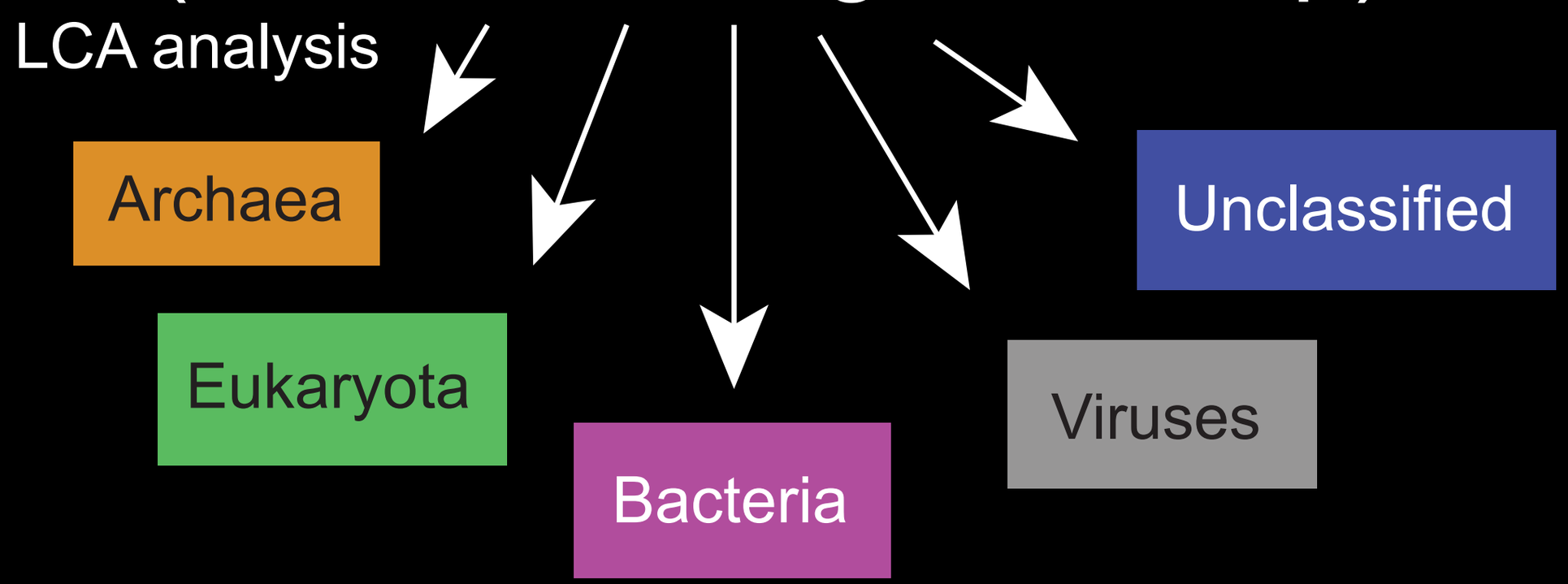
- Can separate non-model eukaryotic host genomes
- Tunes results based on single-copy marker genes
- Utilizes **taxonomic information** to simplify mixtures, thus maintaining performance in complex microbiomes

https://bitbucket.org/jason_c_kwan/autometa
https://hub.docker.com/r/jasonkwan/autometa

bioRxiv **2018**, *doi: 10.1101/251462*

Metagenome assembly

(~100s k contigs, ~1 Gbp)



https://bitbucket.org/jason_c_kwan/autometa
https://hub.docker.com/r/jasonkwan/autometa

bioRxiv **2018**, *doi: 10.1101/251462*

Bacteria

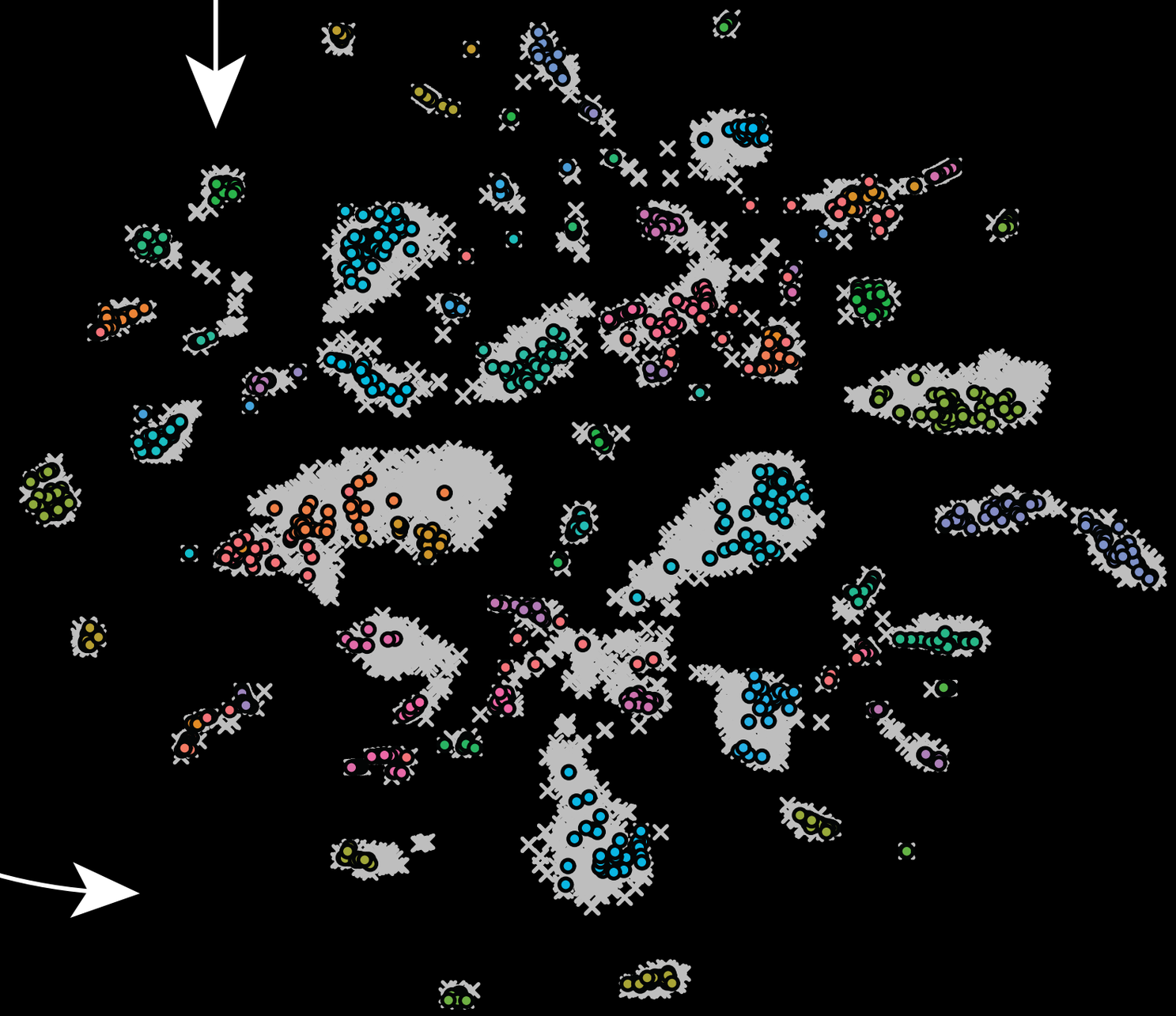
k-mer frequency analysis
Dimension reduction
(BH-SNE)

BH-tSNE + coverage

Genome bins

Unbinned fraction

Iterative
Clustering
(DBSCAN)



https://bitbucket.org/jason_c_kwan/autometa
https://hub.docker.com/r/jasonkwan/autometa

bioRxiv **2018**, *doi: 10.1101/251462*

General unbinned

Split by phylum

Binned

Actinobacteria

Proteobacteria

Verrucomicrobia

Unbinned Actinobacteria

Unbinned Proteobacteria

Unbinned Verrucomicrobia

Split by class

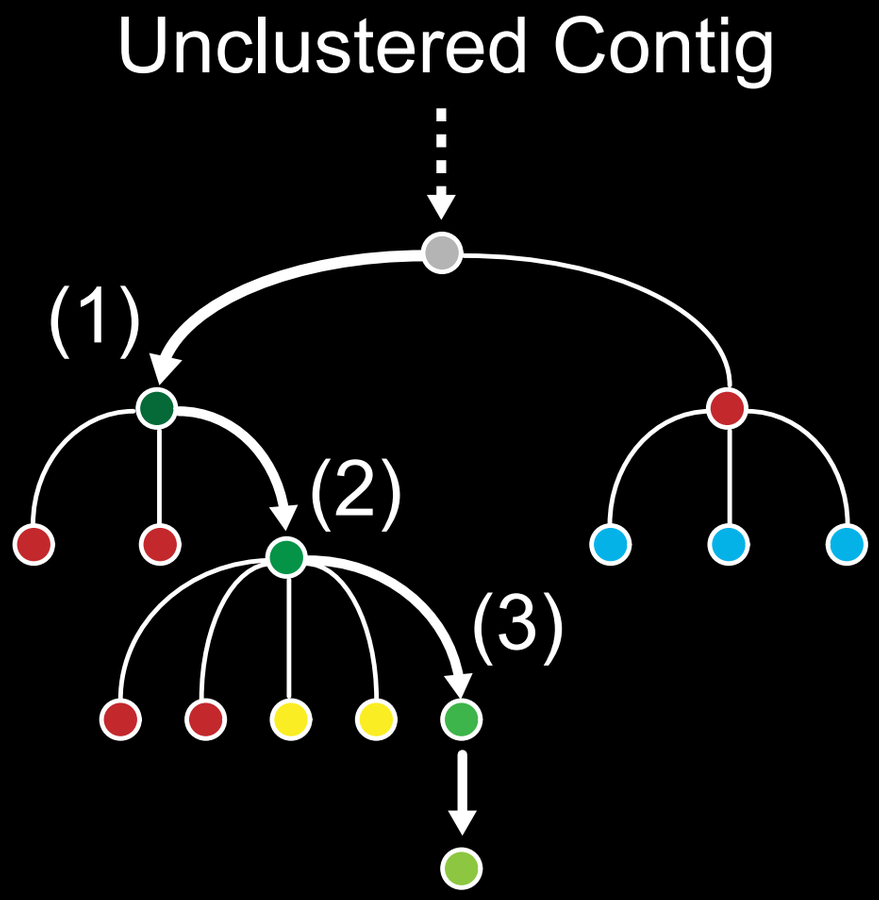
Binned

alpha-proteobacteria

gamma-proteobacteria

delta-proteobacteria

Decision tree classifier

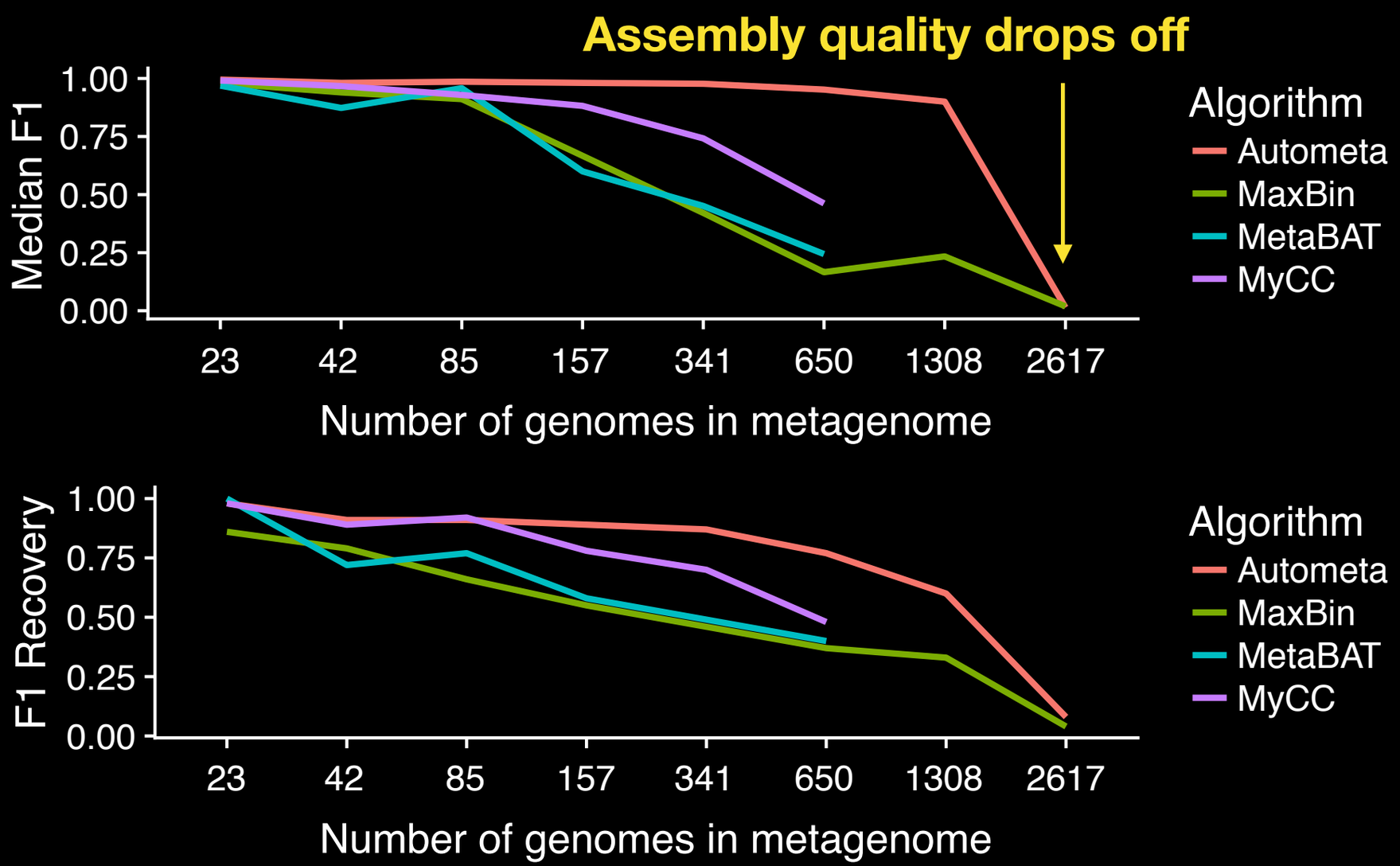


Classification based on:
(1) Composition
(2) Coverage
(3) Homology

https://bitbucket.org/jason_c_kwan/autometa
https://hub.docker.com/r/jasonkwan/autometa

bioRxiv **2018**, *doi: 10.1101/251462*

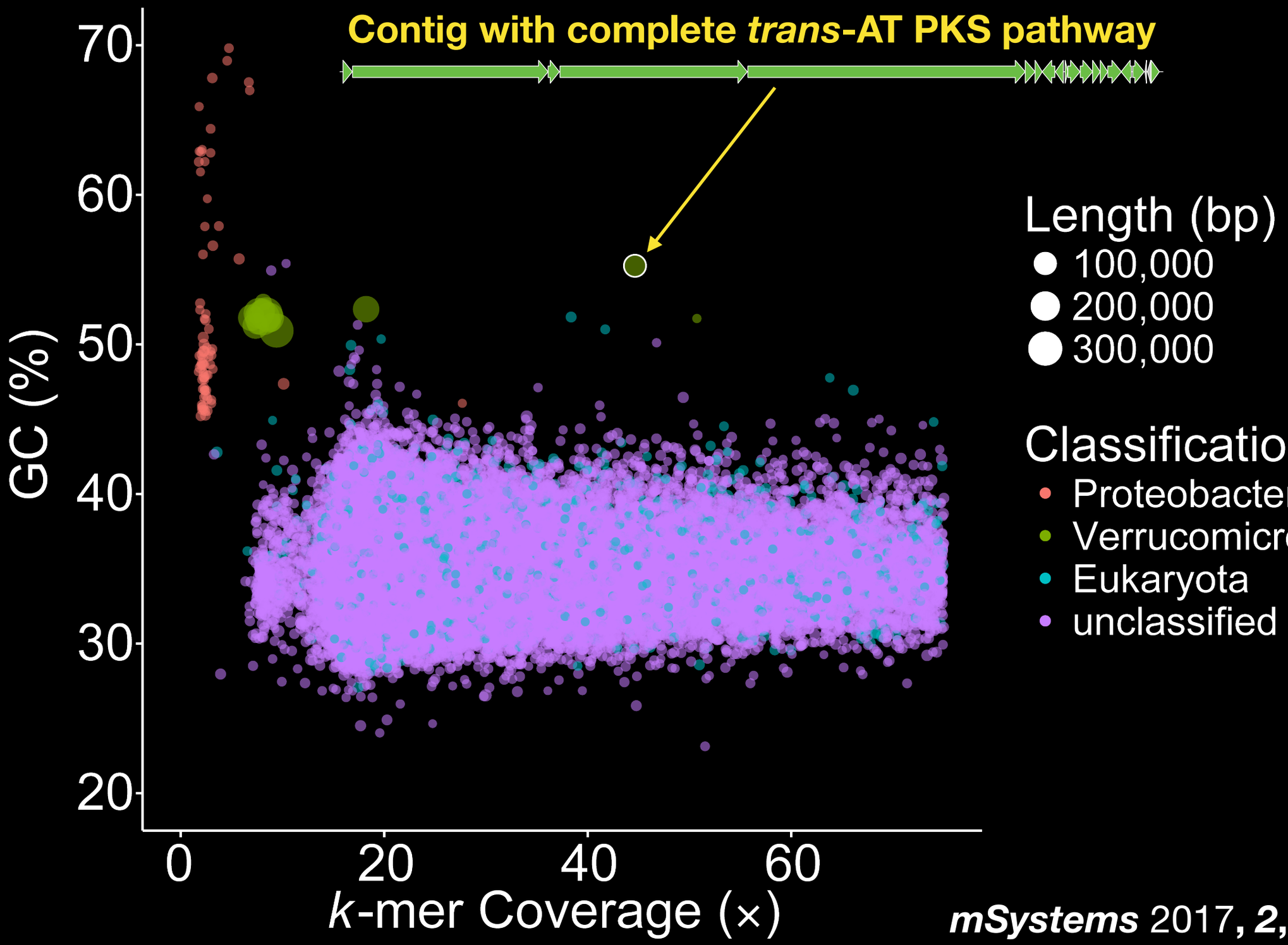
Autometa scales with metagenome complexity



https://bitbucket.org/jason_c_kwan/autometa
https://hub.docker.com/r/jasonkwan/autometa

bioRxiv **2018**, *doi: 10.1101/251462*

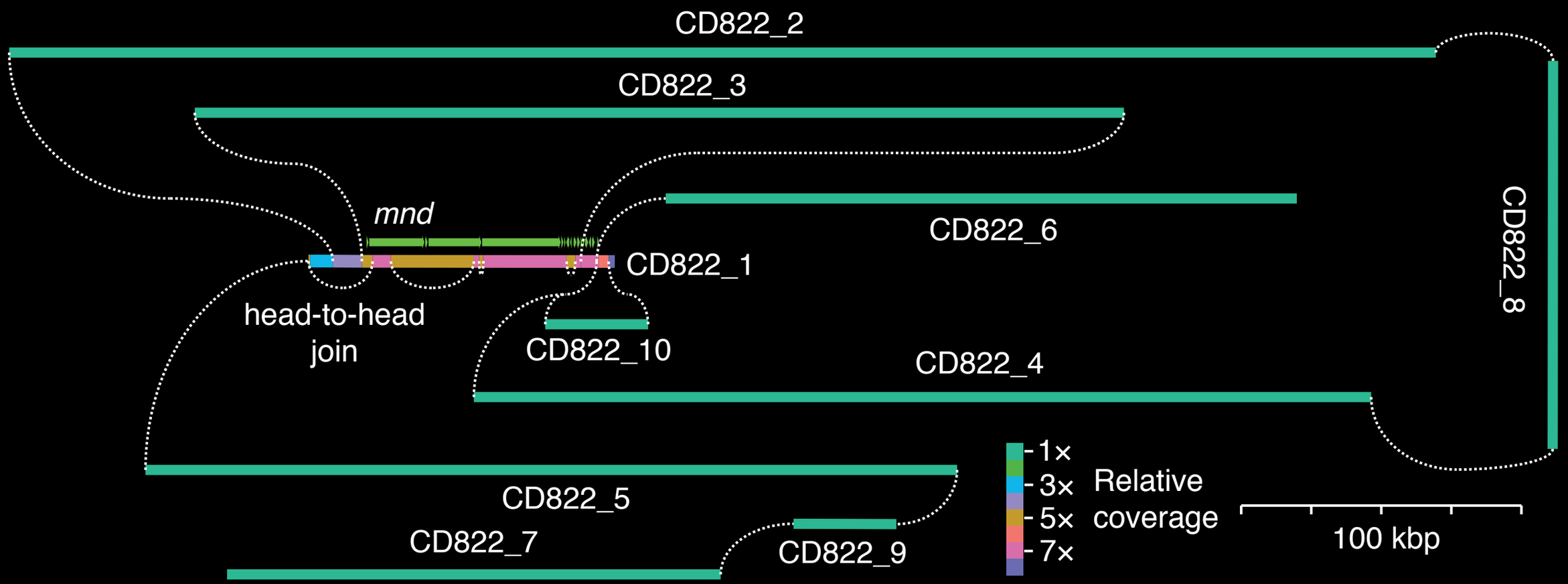
Mandelalides



mSystems 2017**, *2*, e00096-17**

Mandelalides

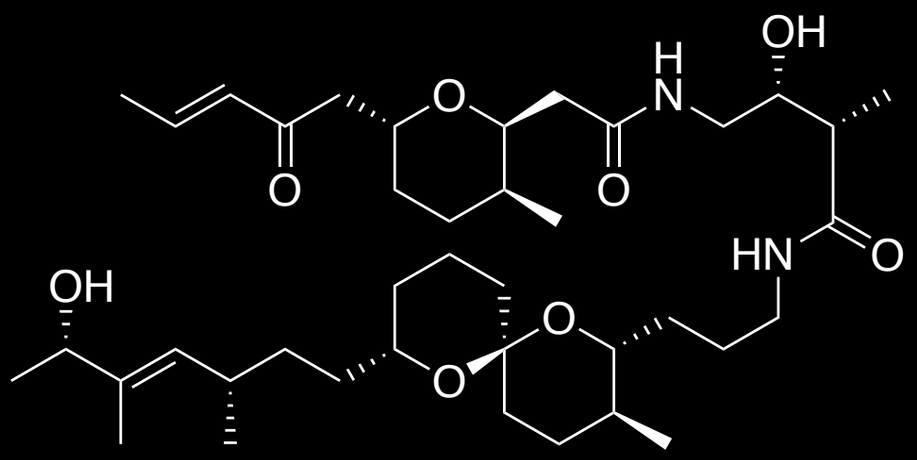
"*Candidatus* Didemnitutus mandela" genome assembly



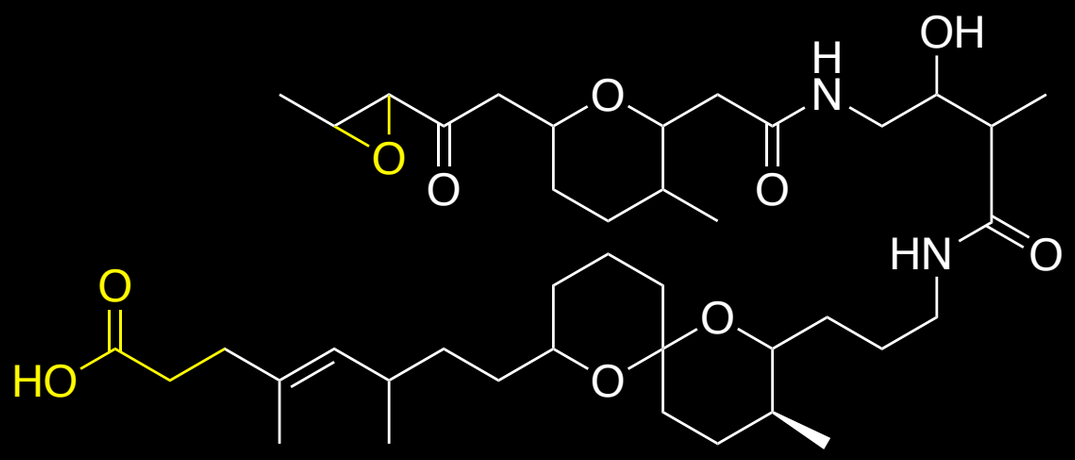
Mnd repeated seven times in an otherwise reduced genome

mSystems 2017**, *2*, e00096-17**

Lagriamide



bistramide A, marine tunicates



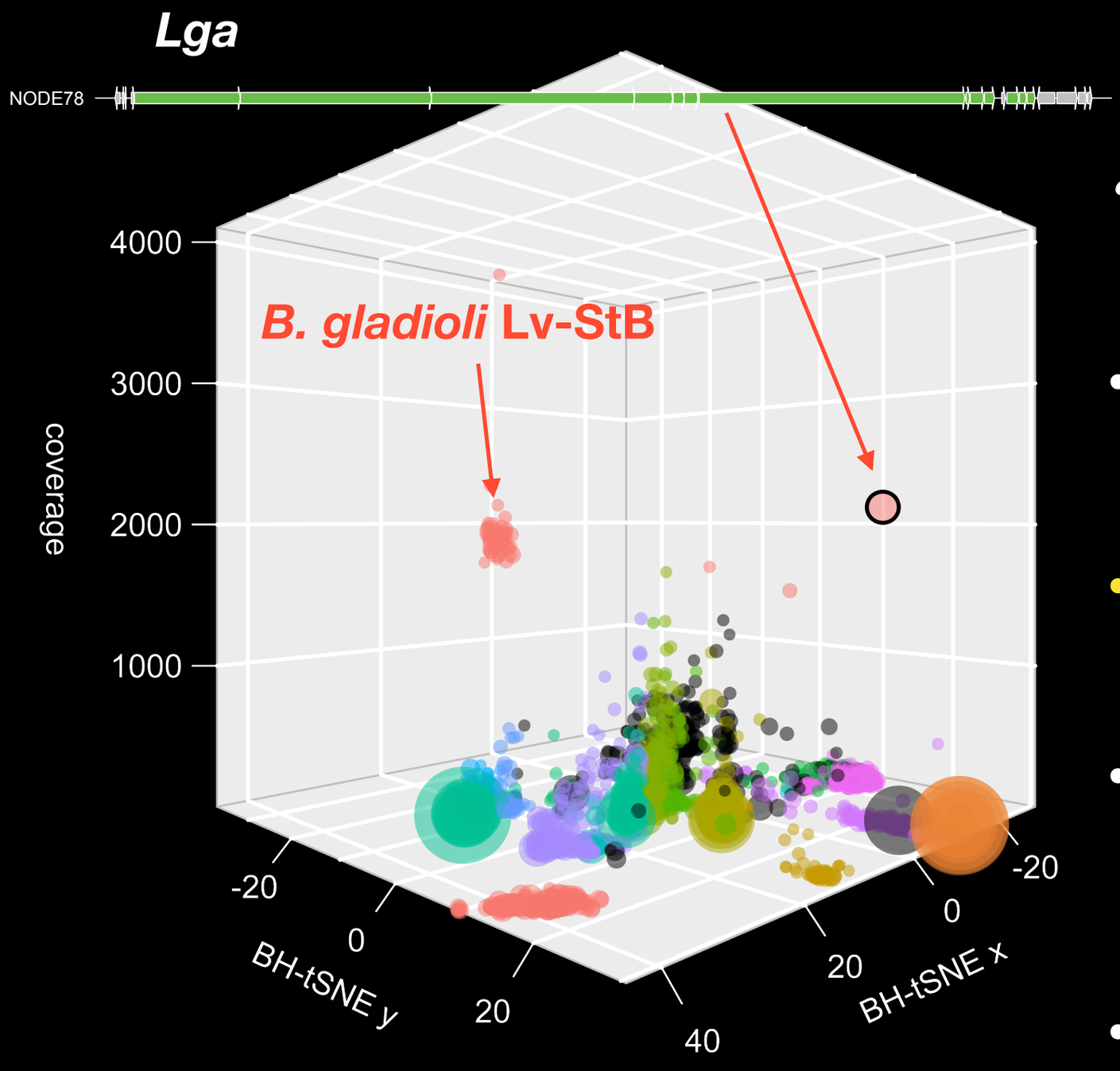
lagriamide (antifungal)

- **Subfamily Lagriinae co-infected with several *Burkholderia gladioli* strains**
- **Lagriamide protects eggs from fungal infection**
- **Cultured strains rare in the field**

Nat. Commun. 2018**, 9, 2478**

***Picture by Laura Flórez**

Lagriamide



Nat. Commun. 2018, **9, 2478**

- ***Lga*** **part of** ***B. gladioli*** **Lv-StB genome (single-cell genomes)**
- **Codon usage in** ***Lga*** **the same but 5-mer frequencies different**
- **Pathway recently horizontally transferred to symbiont**
- **Lv-StB genome *reduced* compared to co-infecting relatives**
- **Why are so many related things present?**

Installation

```
git clone https://bitbucket.org/jason_c_kwan/autometa
docker pull jasonkwan/autometa:latest
```

You could just use download the docker image, but there are docker wrapper scripts in the git repository that are easier to use.

(Also the repository contains the test data)

Separate assembly into kingdom bins

```
make_taxonomy_table_docker.py --assembly ~/autometa/
test_data/scaffolds.fasta --processors 16
--length_cutoff 3000
```

If contigs are NOT named like this:

`NODE_1_length_1389215_cov_225.275`

Then you need to make a coverage table:

```
calculate_read_coverage_docker.py --assembly
~/autometa/test_data/scaffolds.fasta --processors 16
--forward_reads reads_R1.fastq.gz --reverse_reads
reads_R2.fastq.gz
```

Separate assembly into kingdom bins

The process:

- **Identify genes in each contig with Prodigal**
- **Search gene protein sequences against NR with DIAMOND**
- **Determine lowest common ancestor (LCA) of blast hits within 10% of highest bitscore**
- **Determine the taxonomy of each contig**
  - **Consider proteins in contig in descending order of specificity (species, genus, family, order, class, phylum)**
  - **Accept a classification if it represents >50% of proteins in the contig, IF PROTEINS WITH LOWER SPECIFICITY CLASSIFICATIONS ARE ANCESTORS OF THIS CLASSIFICATION.**
  - **If an answer cannot be reached, we calculate the LCA of the contig**

Bin contigs

```
run_autometa_docker.py --assembly Bacteria.fasta
--processors 16 --length_cutoff 3000 --taxonomy_table
taxonomy.tab
```

The process:

- **Find single-copy marker genes with HMMER**
- **Reduce dimensions of 5-mer frequencies with BH-tSNE**
- **Cluster contigs based on BH-tSNE coordinates, coverage and taxonomy**
- **Tune DBSCAN eps clustering parameter to give the highest median completeness for clusters >20% complete and >90% pure**
- **Unclustered contigs are fed into the process again, also further split based on taxonomy**

Recruit unclustered contigs with machine learning

```
ML_recruitment_docker.py --contig_tab
recursive_dbscan_output.tab --recursive --k_mer_matrix
k-mer_matrix –out_table ML_recruitment_output.tab
```

The decision tree classifier “sees” 50 5-mer frequency dimensions (from PCA), contig coverage and taxonomy

All the above in one command

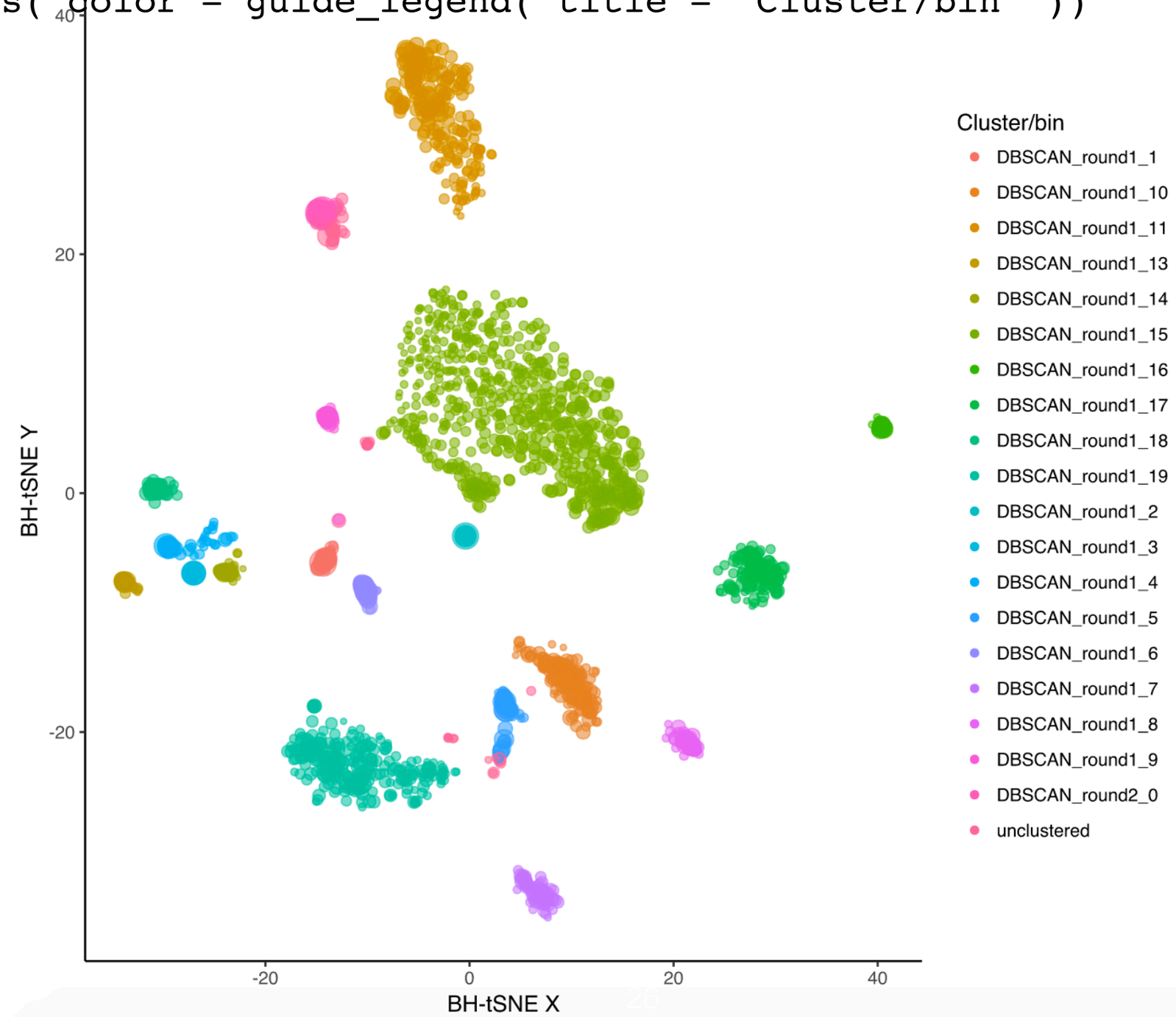
```
run_autometa_docker.py --assembly
~/autometa/test_data/scaffolds.fasta --processors 16
--length_cutoff 3000 --maketaxtable --ML_recruitment
```

Analyze the bins

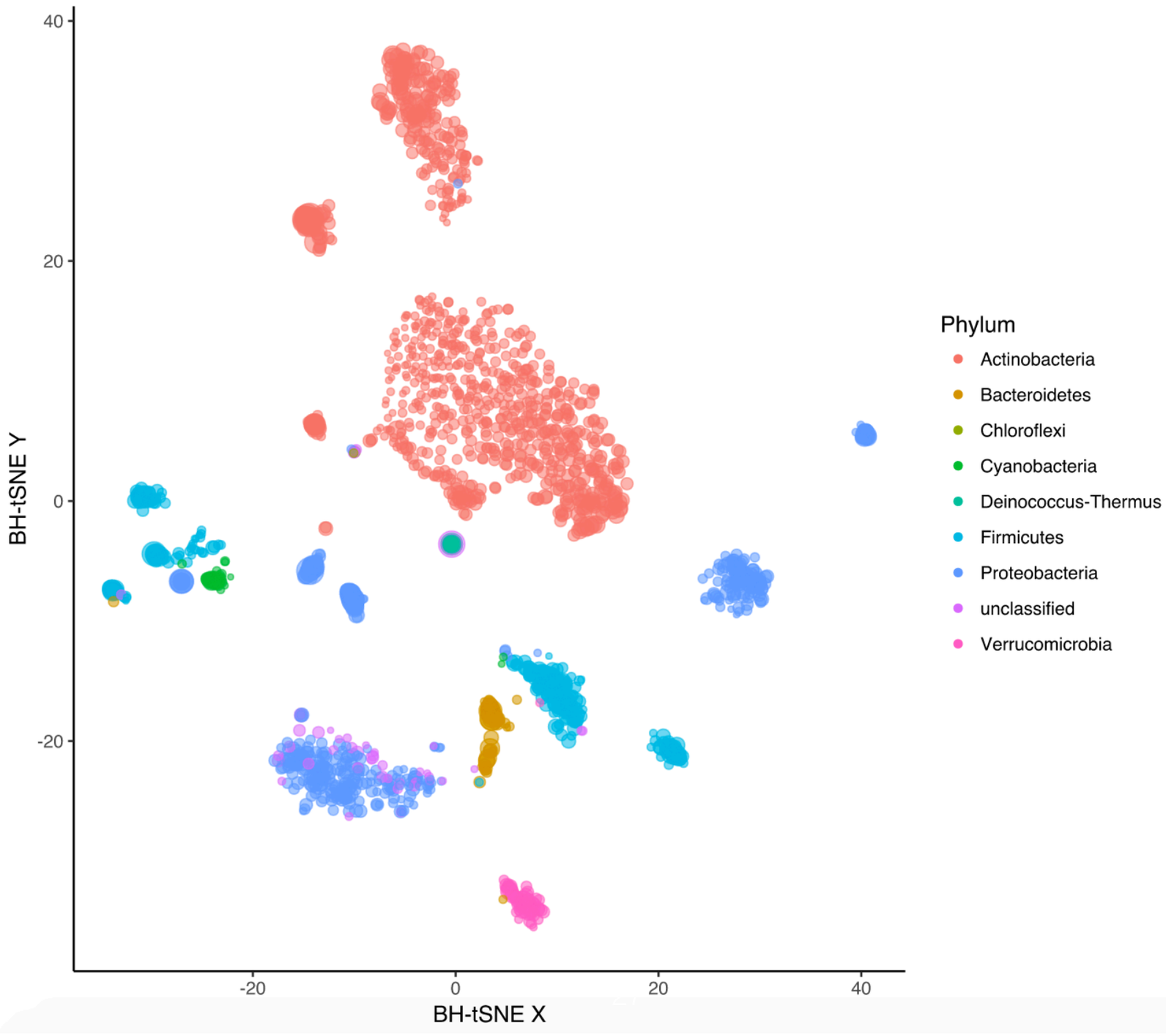
```
cluster_process.py --bin_table
ML_recruitment_output.tab --column
ML_expanded_clustering --fasta Bacteria.fasta --
do_taxonomy --db_dir ~/autometa/databases --output_dir
cluster_process_output
```

- **Creates separate FASTA files for each bin**
- **Summarizes genome size, completeness, purity etc. in a file called “cluster_process_output”**
- **Summarizes bin taxonomies in a file called “cluster_taxonomy.tab”**

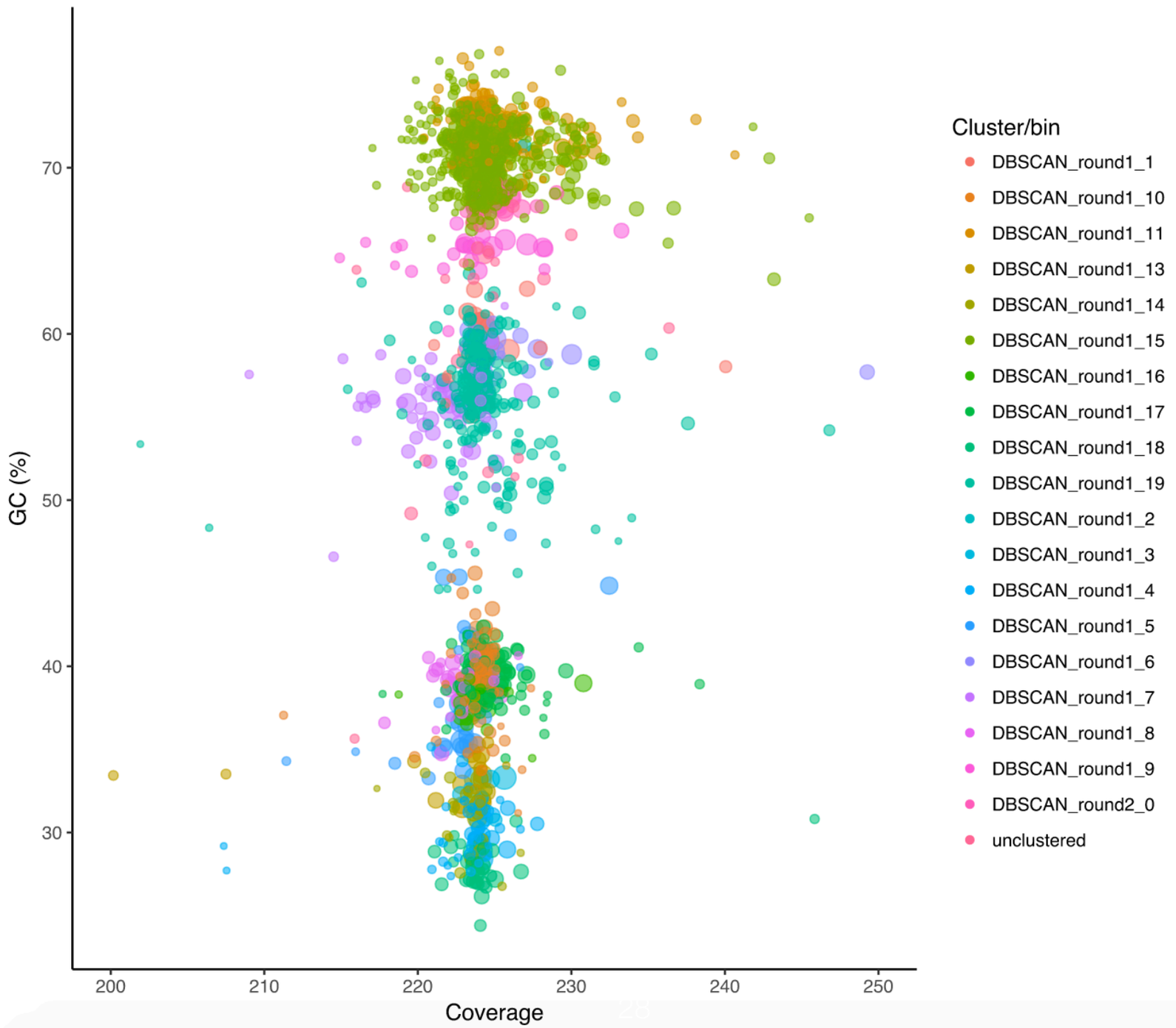
```
library(ggplot2)
data = read.table('ML_recruitment_output.tab', header=TRUE, sep='\t')

ggplot( data, aes( x = bh_tsne_x, y = bh_tsne_y, col =
ML_expanded_clustering )) + \
    geom_point( aes( alphs = 0.5, size = sqrt( data$length ) / 100 )) + \
    guides( color = 'legend', size = 'none', alpha = 'none' ) + \
    theme_classic() + xlab('BH-tSNE X') + ylab('BH-tSNE Y') + \
    guides( color = guide_legend( title = 'Cluster/bin' ))
```



```
ggplot( data, aes( x = bh_tsne_x, y = bh_tsne_y, col = phylum )) + \
    geom_point( aes( alphs = 0.5, size = sqrt( data$length ) / 100 )) + \
    guides( color = 'legend', size = 'none', alpha = 'none' ) + \
    theme_classic() + xlab('BH-tSNE X') + ylab('BH-tSNE Y') + \
    guides( color = guide_legend( title = 'Phylum' ))
```



```
ggplot( data, aes( x = cov, y = gc, col = ML_expanded_clustering )) + \
    geom_point( aes( alphs = 0.5, size = sqrt( data$length ) / 100 )) + \
    guides( color = 'legend', size = 'none', alpha = 'none' ) + \
    theme_classic() + xlab('Coverage') + ylab('GC (%)') + \
    guides( color = guide_legend( title = 'Cluster/bin' )) + \
    scale_x_continuous( limits = c( 200, 250 ))
```



Running Autometa on CHTC

- **Although CHTC supports Docker, it is not running Docker but rather some sort of different compatible software layer (Singularity)**
- **Because the Autometa image is so big, you have to download it to /mnt/gluster ahead of time and run it in a different way to the standard CHTC Docker instructions**

```
# Do this in an interactive job
singularity build autometa.img docker://jasonkwan/autometa:latest
mv autometa.img /mnt/gluster/<user>/
```

You will need:

- **A submit file (autometa.sub)**
- **A singularity wrapper script (run_singularity_gluster.sh)**
- **A script to run Autometa in the container (autometa.sh)**

Running Autometa on CHTC

autometa.sub

```
job = autometa
universe = vanilla
log = $(job)_$(Cluster).log
executable = run_singularity_gluster.sh
#This pulls argument from last line
arguments = "autometa.img autometa.sh --assembly scaffolds.fasta
   --processors 16 --length_cutoff 3000 --maketaxtable –ML_recruitment
   --output_dir autometa_output“
output = $(job)_$(Cluster)_$(Process).out
error = $(job)_$(Cluster)_$(Process).err
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
transfer_input_files = scaffolds.fasta,autometa.sh
transfer_output_files = autometa_output.tar.gz
request_cpus = 16
request_memory = 16GB
request_disk = 16GB
requirements = (OpSysMajorVer == 7) && (Target.HasGluster == true) &&
(Target.HasSingularity == true)
notification = Never
materialize_max_idle = 2000
queue 1
```

Running Autometa on CHTC

run_singularity_gluster.sh

```
#!/bin/bash
img=$1
exec=$2
shift 2
args=$*

cp /mnt/gluster/<user>/$img ./

singularity exec -B $_CONDOR_SCRATCH_DIR:/scratch $img /scratch/$exec $args

rm $img
```

Running Autometa on CHTC

autometa.sh

```
#!/bin/bash

args=$*

run_autometa.py $args

# Compress output folder
tar cvzf autometa_output.tar.gz autometa_output
```

Planned improvements

- Distinguish different divergent species through assignment of placeholder taxids.

- Use taxonomy-specific sets of single-copy marker genes, similar to CheckM.

- Develop pangenome-aware methods to leverage co-occurence of species in multiple samples for binning.