Genomics Quality Control with BUSCO - Genome Mode

Objectives	☐ Assess a genome assembly for completeness using BUSCO ☐ Examine the various analysis steps of a BUSCO assessment ☐ Investigate the results of a BUSCO genome assessment
Expected Background Knowledge	☐ Knowledge of what the Benchmarking Universal Single-Copy Orthologue (BUSCO) assessment tool if designed for ☐ Knowledge of working on a terminal to execute analysis commands and navigate the file system (e.g. UNIX Fundamentals) ☐ Knowledge of common terms used to describe the quality and features of a genome assembly (e.g. Assembly Quality)
Learning Outcomes	☐ Learn how to run a BUSCO assessment of a genome assembly ☐ Learn about the steps taken by BUSCO during an assessment ☐ Learn how to interpret the results of a BUSCO assessment

Learning Stage	**Beginner**	~~Intermediate~~	~~Advanced~~
Time Estimate	**Less than 1 hour**	~~1 - 2 hours~~	~~More than 2 hours~~

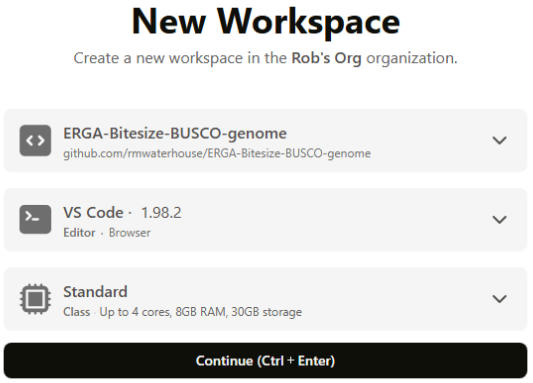
Resources	☐ You will need a GitHub account ☐ BUSCO - Benchmarking Universal Single-Copy Orthologues: Website here; Publication here; Userguide here ☐ OrthoDB - orthology database: Website here; Publication here; Userguide here
Contributors	☐ Robert Waterhouse, SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland, 0000-0003-4199-9052

Before you start: Launching your GitPod Workspace

[1] You must first have a GitHub account
[2] You must first have a GitPod account LINKED to your GitHub account

If you have [1] and [2] then simply clicking this link should launch your Workspace:
https://gitpod.io/#https://github.com/rmwaterhouse/ERGA-Bitesize-BUSCO-genome

Then click Continue with the default settings …



If you need to copy and paste commands from this document into your GitPod Workspace you will need to give GitPod access to your clipboard (a popup message => 'Allow')

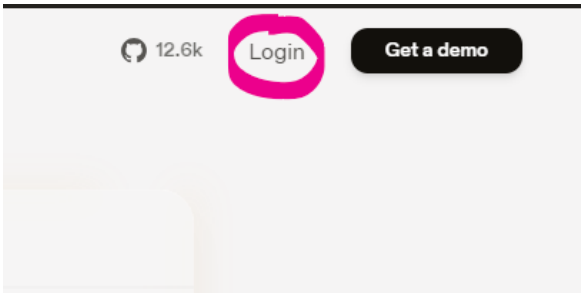
If you do not yet have [1] and [2] then you first need to …

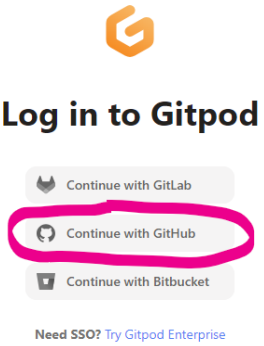
- **Create an account at GitHub**

https://github.com/
Remember your email, your username, and your password!

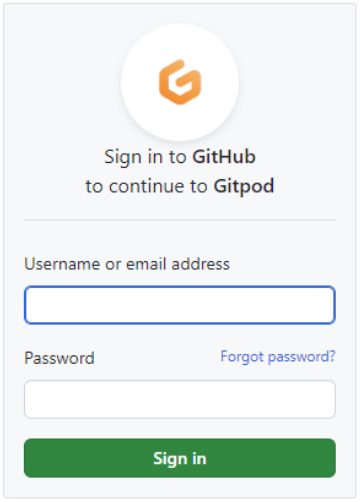
- **Open GitPod and link your GitHub Account**

https://www.gitpod.io/





Authenticate your login with your GitHub credentials!



Now you have [1] and [2], simply clicking this link should launch your Workspace:
https://gitpod.io/#https://github.com/rmwaterhouse/ERGA-Bitesize-BUSCO-genome

Tutorial: BUSCO - the what, why, and how!

Assessing genome assemblies for completeness

- Let's start by fetching some genome data that we wish to assess - We will work on a small genome so that it does not take too long to run the analyses, hence we have chosen *Saccharomyces jurei*, a newly discovered fungal species with a small genome of 12 Mbps

- At NCBI: https://www.ncbi.nlm.nih.gov/datasets/genome/GCA_900290405.1/

- The summary statistics are already provided by NCBI, but not the BUSCO evaluations of the genome assembly

Assembly statistics

	GenBank
Genome size	11.8 Mb
Total ungapped length	11.8 Mb
Number of chromosomes	17
Number of organelles	1
Number of scaffolds	17
Scaffold N50	738.7 kb
Scaffold L50	7
Number of contigs	17
Contig N50	738.7 kb
Contig L50	7
GC percent	38
Genome coverage	250.0x
Assembly level	Complete Genome

Chromosomes

Question: What is the "Scaffold N50" and what does it mean?

Hints here if needed: https://en.wikipedia.org/wiki/N50,_L50,_and_related_statistics

- We will use the curl command and the NCBI Datasets framework to fetch the genome assembly in FASTA format and then unzip the downloaded file:
- `curl -OJX GET "https://api.ncbi.nlm.nih.gov/datasets/v2alpha/genome/accession/GCA_900290405.1/download?include_annotation_type=GENOME_FASTA&filename=GCA_900290405.1.zip" -H "Accept: application/zip"`
- `unzip GCA_900290405.1.zip -d my_downloads`

- Your terminal should look something like this:

```
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $ curl -OJX GET "https://api.ncbi.nlm.nih.gov/datasets/v2alpha/genome/accession/GCA_900290405.1/downl
oad?include_annotation_type=GENOME_FASTA&filename=GCA_900290405.1.zip" -H "Accept: application/zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 3690k    0 3690k    0     0  2807k      0 --:--:--  0:00:01 --:--:-- 2806k
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $ unzip GCA_900290405.1.zip -d my_downloads
Archive:  GCA_900290405.1.zip
  inflating: my_downloads/README.md
  inflating: my_downloads/ncbi_dataset/data/assembly_data_report.jsonl
  inflating: my_downloads/ncbi_dataset/data/GCA_900290405.1/GCA_900290405.1_SacJureiUoM1_genomic.fna
  inflating: my_downloads/ncbi_dataset/data/dataset_catalog.json
  inflating: my_downloads/md5sum.txt
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main)
```

- We will take a look at the first few lines of the FASTA file ('head' command):
- `head my_downloads/ncbi_dataset/data/GCA_900290405.1/GCA_900290405.1_SacJureiUoM1_genomic.fna`

```
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $ head my_downloads/ncbi_dataset/data/GCA_900290405.1/GCA_900290405.1_SacJureiUoM1_genomic.fna
>LT986461.1 Saccharomyces jurei genome assembly, chromosome: I.1_XIII.2
acacccacacaccacaccacacacccacacccacacccacacccacacccacacccacacccacacccacacccacacc
cacacccacacccacacccacacccacacccacacccacacccacacccacacccacacccacacccacacccacacca
cacacccacacccacacacccacacccacacccacacccacacccacacccacacccacacccacacccacacccacaca
accacacccacacaccacacccacacccacacccacacccacacccacacccacacccacacccacacccacacccaca
cccaaCTCTAACTCTAACTCTAACTCTAACACTCTCACTCTAACACTCTCACTTGCTCTAACAATAACCCTGATTATCAC
AGTATCTCTTACCCTGATTATGCTACCAGCCCTGATCCAACCCTACCTCAAGCCCTGTCTCTCTGACTAACCCTACCTCT
CTATCTCTCATCCCTACCTGCTTCTCTATCTCCCACCCCTACCTGCCATCCTTCACATCTCACCACTACTCTCGAACTAC
CACACCAACTACCCCCTACCAACTATCTATCACGAAACCACCACTTCCACTTATCCTACCATCTACCATCTACCACCTAC
CATCTACCatatgtcaacaatcacatgtcaacaatcacatgtcaacaatcacatgtcaacaatcatatgtaaacaatcat
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $
```

- Now we have the genome, we can run BUSCO assessments in the genome mode to quantify gene completeness of this genome assembly

- You can see the command line options explained on the BUSCO website (also below)

Running BUSCO

Command Line Options

Mandatory parameters

```
busco -i [SEQUENCE_FILE] -m [MODE] [OTHER OPTIONS]
```

`-i` or `--in` defines the input file to analyse which is either a nucleotide fasta file or a protein fasta file, depending on the BUSCO mode. As of v5.1.0 the input argument can now also be a directory containing fasta files to run in batch mode.

`-m` or `--mode` sets the assessment MODE: genome, proteins, transcriptome

Recommended parameters

`-l` or `--lineage_dataset` Specify the name of the BUSCO lineage dataset to be used, e.g. `kitasatospora_odb12`. A full list of available datasets can be viewed by entering `busco --list-datasets`. You should always select the dataset that is most closely related to the assembly or gene set you are assessing. If you are unsure, you can use the `--auto-lineage` option to automatically select the most appropriate dataset. BUSCO will automatically download the requested dataset if it is not already present in the download folder. You can optionally provide a path to a local dataset instead of a name, e.g. `-l /path/to/my/dataset`.

`-c` or `--cpu` Specify the number of threads/cores to use. Unless this is specified BUSCO will only use one CPU, which could cause a long run time.

`-o` or `--out` Give your analysis run a recognisable short name. Output folders and files will be labelled with this name. If not specified the output will take the form "BUSCO_<input_filename>"

- **The four main required input options for us therefore are:**

- -i my_downloads/ncbi_dataset/data/GCA_900290405.1/GCA_900290405.1_SacJureiUoM1_genomic.fna
  - Defines the input file to analyse, here the genome in FASTA format
- -o SacJurei
  - Gives your analysis run a recognisable short name

- -m genome
  - Sets the assessment MODE: genome, proteins, or transcriptome, here we are assessing a genome so we choose the genome mode

- -l eukaryota_odb12
  - Specifies the name of the BUSCO lineage dataset to be used, here we choose to use the Eukaryota lineage dataset from OrthoDB v12

- *We will also specify the job to use 4 CPUs in order to speed up the task:*
  - -c 4

- *We will also specify to use MetaEuk as the gene predictor:*
  - --metaeuk

Special step for suppressing warnings from BUSCO v5.8.2 with respect to Python's updated treatment of escape characters.

To avoid the warnings first execute the following command:

- `export PYTHONWARNINGS="ignore"`

Note, the analysis runs without any errors, only warnings from Python, this will be corrected in future BUSCO releases.

- The whole command will therefore be as follows … go ahead and launch it!
- `busco -i my_downloads/ncbi_dataset/data/GCA_900290405.1/GCA_900290405.1_SacJureiUoM1_genomic.fna -o SacJurei -m genome -l eukaryota_odb12 -c 4 --metaeuk`

- On the terminal you can see which steps BUSCO is executing:
  - Configuration
  - Dataset download
  - MetaEuk ⇐ *note that this is not the default "gene finding" approach … we specifically told BUSCO to use the MetaEuk approach*

Question: What other "gene finding" approaches are possible to use with BUSCO?

- The terminal should look like this, confirming the configuration, the fact that we are running in genome mode, the genome file you want to assess, and the lineage dataset that you want to use for the assessment:

```
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $ export PYTHONWARNINGS="ignore"
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $ busco -i my_downloads/ncbi_dataset/data/GCA_900290405.1/GCA_900290405.1_SacJureiUoM1_genomic.fna
-o SacJurei -m genome -l eukaryota_odb12 -c 4 --metaeuk
2025-06-09 10:30:05 INFO:       ***** Start a BUSCO v5.8.2 analysis, current time: 06/09/2025 10:30:05 *****
2025-06-09 10:30:05 INFO:       Configuring BUSCO with local environment
2025-06-09 10:30:05 INFO:       Running genome mode
2025-06-09 10:30:05 INFO:       Downloading information on latest versions of BUSCO data...
2025-06-09 10:30:07 INFO:       Input file is /workspace/ERGA-Bitesize-BUSCO-genome/my_downloads/ncbi_dataset/data/GCA_900290405.1/GCA_900290405.1_SacJureiUoM1_genomic.fna
2025-06-09 10:30:08 INFO:       Running BUSCO using lineage dataset eukaryota_odb12 (eukaryota, 2025-04-11)
```

- Which BUSCO lineage to choose - we used the “eukaryota” dataset:
    - eukaryota_odb12 ⇒ 129 BUSCOs
        - fungi_odb12 ⇒ 1’122 BUSCOs
            - ascomycota_odb12 ⇒ 2’826 BUSCOs
                - saccharomycetes_odb12 ⇒ 2’319 BUSCOs
                    - saccharomycetaceae_odb12 ⇒ 3’282 BUSCOs

Question: Why would you want to use a more specific (Saccharomycetaceae, family level) or a less specific (fungi, kingdom level; or Eukaryota, domain level) lineage dataset for your BUSCO evaluations?

- The analysis continues with the following steps being printed to the terminal:
    - Once MetaEuk (first round) is completed (yellow), then …
    - The hmmsearch step follows

```
2025-06-09 10:30:08 INFO:       Running BUSCO using lineage dataset eukaryota_odb12 (eukaryota, 2025-04-11)
2025-06-09 10:30:08 INFO:       Running 1 job(s) on bbtools, starting at 06/09/2025 10:30:08
2025-06-09 10:30:10 INFO:       [bbtools]       1 of 1 task(s) completed
2025-06-09 10:30:10 INFO:       Running 1 job(s) on metaeuk, starting at 06/09/2025 10:30:10
2025-06-09 10:31:34 INFO:       [metaeuk]       1 of 1 task(s) completed
2025-06-09 10:31:34 INFO:       ***** Run HMMER on gene sequences *****
2025-06-09 10:31:34 INFO:       Running 129 job(s) on hmmsearch, starting at 06/09/2025 10:31:34
2025-06-09 10:31:36 INFO:       [hmmsearch]     13 of 129 task(s) completed
2025-06-09 10:31:36 INFO:       [hmmsearch]     26 of 129 task(s) completed
2025-06-09 10:31:36 INFO:       [hmmsearch]     39 of 129 task(s) completed
2025-06-09 10:31:37 INFO:       [hmmsearch]     52 of 129 task(s) completed
2025-06-09 10:31:37 INFO:       [hmmsearch]     65 of 129 task(s) completed
2025-06-09 10:31:37 INFO:       [hmmsearch]     78 of 129 task(s) completed
2025-06-09 10:31:38 INFO:       [hmmsearch]     91 of 129 task(s) completed
2025-06-09 10:31:39 INFO:       [hmmsearch]     104 of 129 task(s) completed
2025-06-09 10:31:40 INFO:       [hmmsearch]     117 of 129 task(s) completed
2025-06-09 10:31:41 INFO:       [hmmsearch]     129 of 129 task(s) completed
2025-06-09 10:31:41 INFO:       153 exons in total
```

Question: What is the hmmsearch step doing?

- The analysis continues with the following steps being printed to the terminal:
  - The extraction of missing and fragmented buscos (blue)
  - A second round of metaeuk predictions (yellow)
  - Then a second round of hmmsearch follows (green)
  - To finally give the results …

```
2025-06-09 10:31:41 INFO:      153 exons in total
2025-06-09 10:31:41 INFO:      Extracting missing and fragmented buscos from the file refseq_db.faa...
2025-06-09 10:31:42 INFO:      Running 1 job(s) on metaeuk, starting at 06/09/2025 10:31:42
2025-06-09 10:33:57 INFO:      [metaeuk]       1 of 1 task(s) completed
2025-06-09 10:33:57 INFO:      ***** Run HMMER on gene sequences *****
2025-06-09 10:33:57 INFO:      Running 20 job(s) on hmmsearch, starting at 06/09/2025 10:33:57
2025-06-09 10:33:59 INFO:      [hmmsearch]     2 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     4 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     6 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     8 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     10 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     12 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     14 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     16 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     18 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      [hmmsearch]     20 of 20 task(s) completed
2025-06-09 10:33:59 INFO:      125 exons in total
2025-06-09 10:33:59 INFO:      Results:        C:85.3%[S:84.5%,D:0.8%],F:6.2%,M:8.5%,n:129
```

- The analysis should take about 4 minutes to complete

```
2025-06-09 10:34:00 INFO:

    ---------------------------------------------------
    |Results from dataset eukaryota_odb12             |
    ---------------------------------------------------
    |C:85.3%[S:84.5%,D:0.8%],F:6.2%,M:8.5%,n:129      |
    |110    Complete BUSCOs (C)                       |
    |109    Complete and single-copy BUSCOs (S)       |
    |1    Complete and duplicated BUSCOs (D)          |
    |8    Fragmented BUSCOs (F)                       |
    |11    Missing BUSCOs (M)                         |
    |129    Total BUSCO groups searched               |
    ---------------------------------------------------
2025-06-09 10:34:00 INFO:       BUSCO analysis done. Total running time: 232 seconds
2025-06-09 10:34:00 INFO:       Results written in /workspace/ERGA-Bitesize-BUSCO-genome/SacJurei
2025-06-09 10:34:00 INFO:       For assistance with interpreting the results, please consult the userguide: https://busco.ezlab.org/busco_userguide.html

2025-06-09 10:34:00 INFO:       Visit this page https://gitlab.com/ezlab/busco#how-to-cite-busco to see how to cite BUSCO
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main)
```

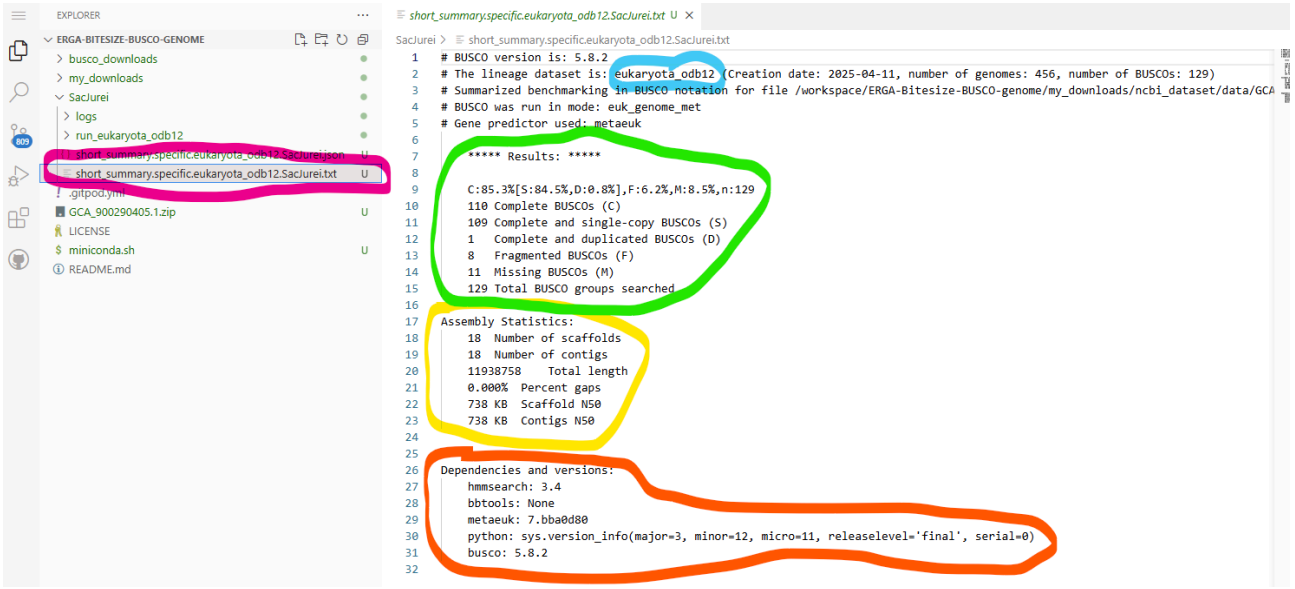
Question: How many BUSCO groups in total are there in this eukaryota_odb12 lineage dataset? How many were found as complete? How many could not be found?

- Let's explore the results of a typical genome assembly assessment run (ls -l lists the files in your output folder SacJurei):
- `ls -l SacJurei/`

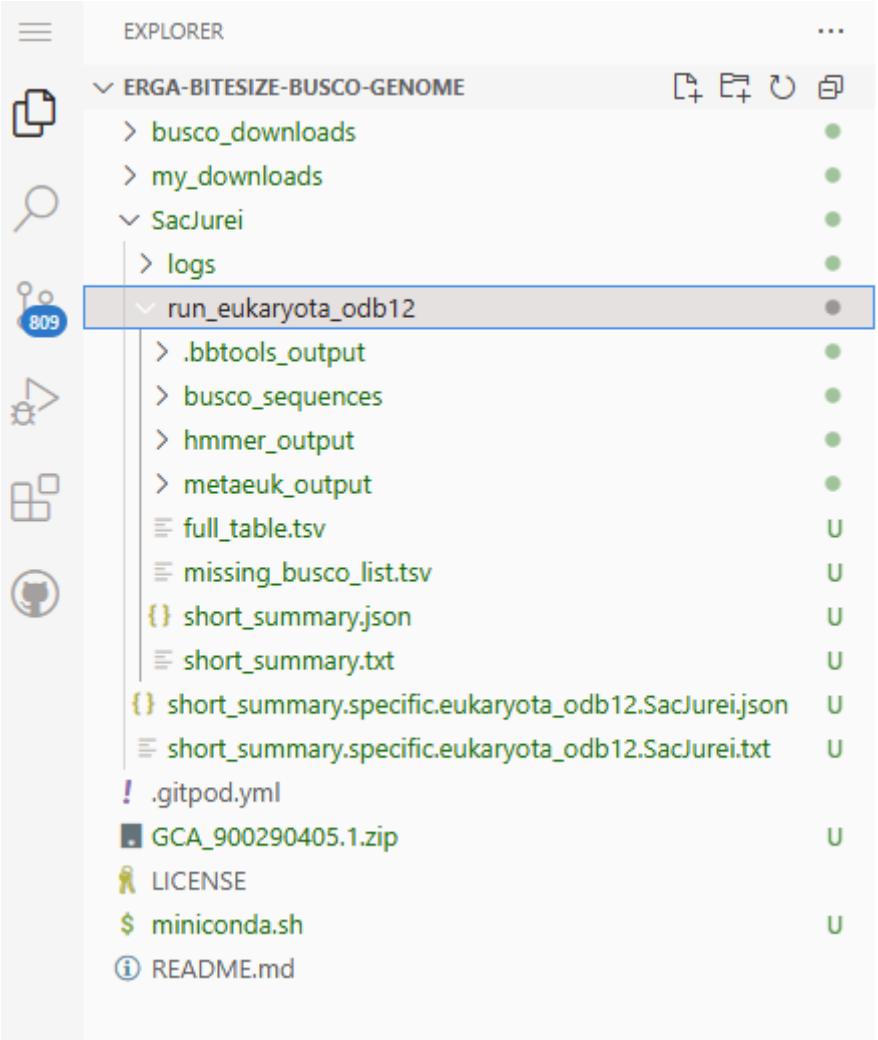
```
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $ ls -l SacJurei/
total 16
drwxr-xr-x 2 gitpod gitpod 4096 Jun  9 10:34 logs
drwxr-xr-x 6 gitpod gitpod 4096 Jun  9 10:34 run_eukaryota_odb12
-rw-r--r-- 1 gitpod gitpod 3029 Jun  9 10:34 short_summary.specific.eukaryota_odb12.SacJurei.json
-rw-r--r-- 1 gitpod gitpod 1017 Jun  9 10:34 short_summary.specific.eukaryota_odb12.SacJurei.txt
(bitesize-busco-genome) gitpod /workspace/ERGA-Bitesize-BUSCO-genome (main) $
```

- **logs** - logs of all steps of the assessment workflow, this can be useful if something went wrong and you need to investigate why

- **short_summary.specific.eukaryota_odb12.SacJurei** (TEXT and JSON versions) - open the .txt file in the text editor (pink, explorer)
    - Indicates the lineage dataset that was used (blue)
    - Summarises the main results (green)
    - Provides some assembly statistics (yellow)
    - Lists the versions of all the tools used during this run (orange)



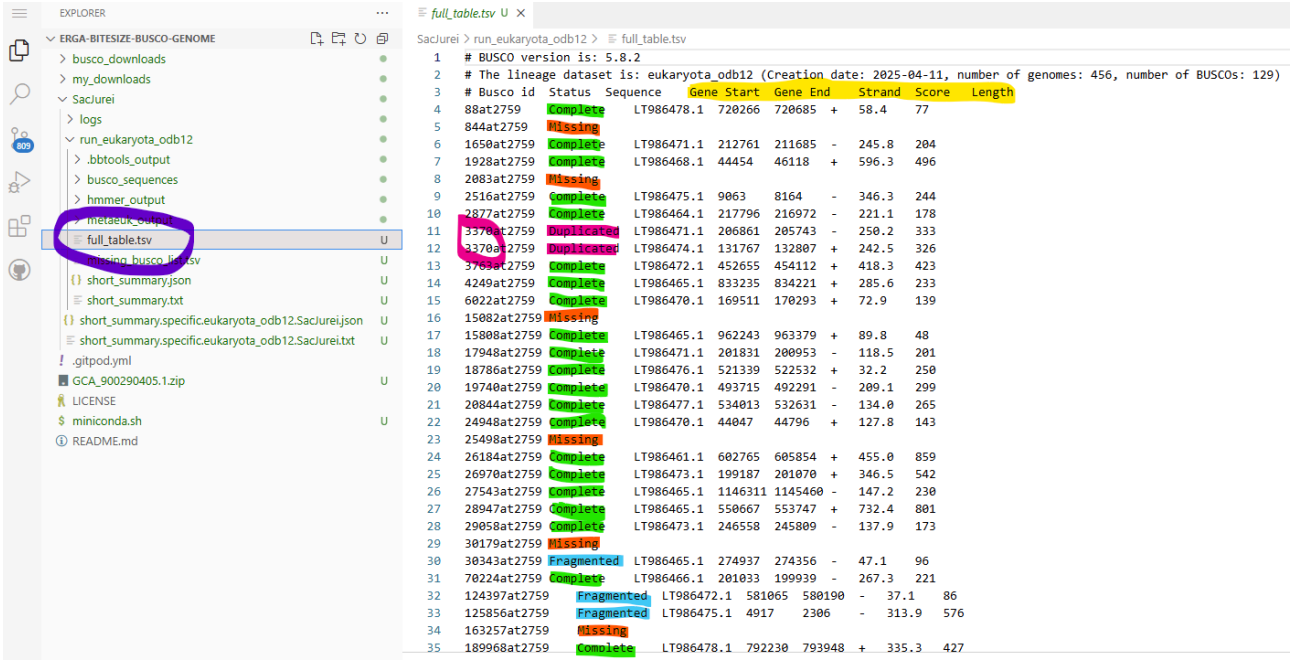
- **run_eukaryota_odb12** - folder with the full results from the run



  - **busco_sequences**: PROTEIN (.faa files) and DNA (.fna file) sequences provided, as well as the gene model annotations (.gff files) for those BUSCOs found to be fragmented, multi-copy complete, or single-copy complete
    - **fragmented_busco_sequences**
    - **multi_copy_busco_sequences**
    - **single_copy_busco_sequences**

Question: Why might it be useful to have access to the sequences that have been predicted for the BUSCO genes found as part of your assessment?

- **full_table.tsv**
  - Details of status (Complete, Duplicated, Fragmented, or Missing), genomic locations, scores, and lengths of all searched BUSCOs

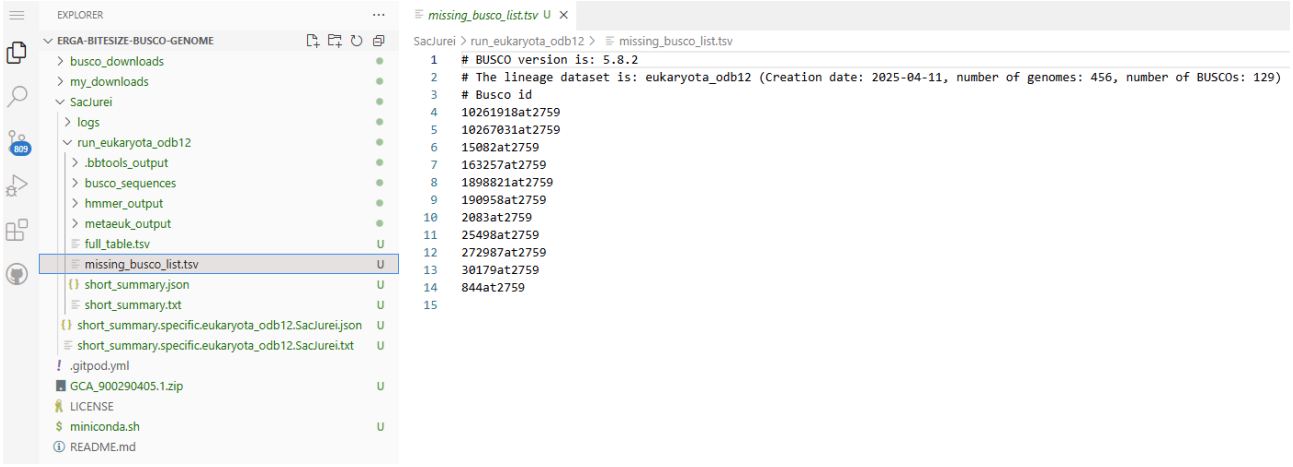


- **hmmer_output** - searching predicted proteins against BUSCO profiles
  - initial_run_results (round 1 search results)
  - rerun_results (round 2 search results)

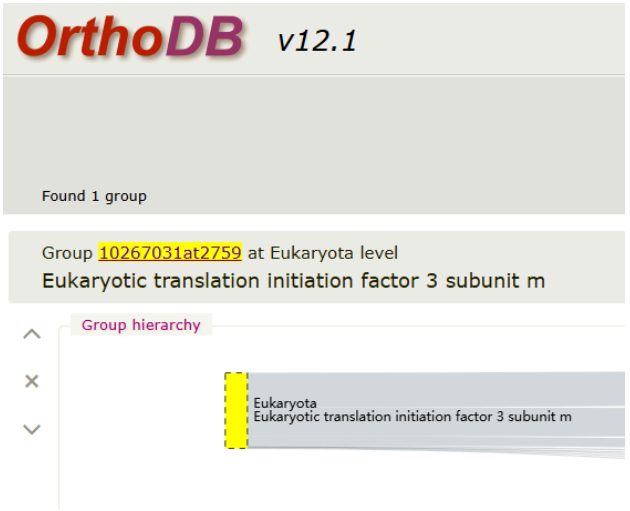
- **metaeuk_output** - the gene prediction results
  - initial_results (round 1 search results)
  - rerun_results (round 2 search results)

- **missing_busco_list.tsv**
  - The BUSCOs that were never found

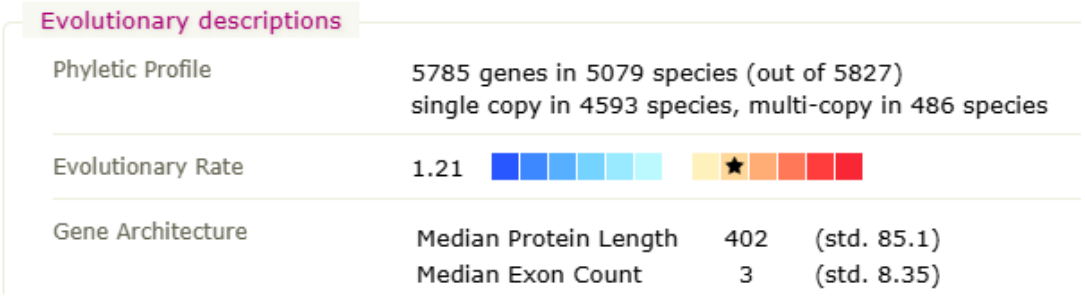
- Let's check some of these apparently missing BUSCOs



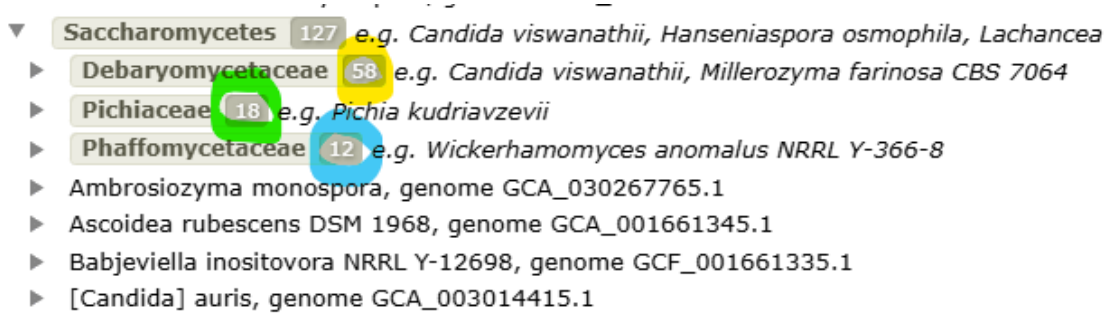
- Search OrthoDB v12 for a missing BUSCO, 10267031at2759
- Description: Eukaryotic translation initiation factor 3 subunit m



- This gene has orthologues in 87% of eukaryotes at OrthoDB v12 (5079 species out of 5827 in total) — of these, it is single-copy in 90% (4593 out of 5079 species)



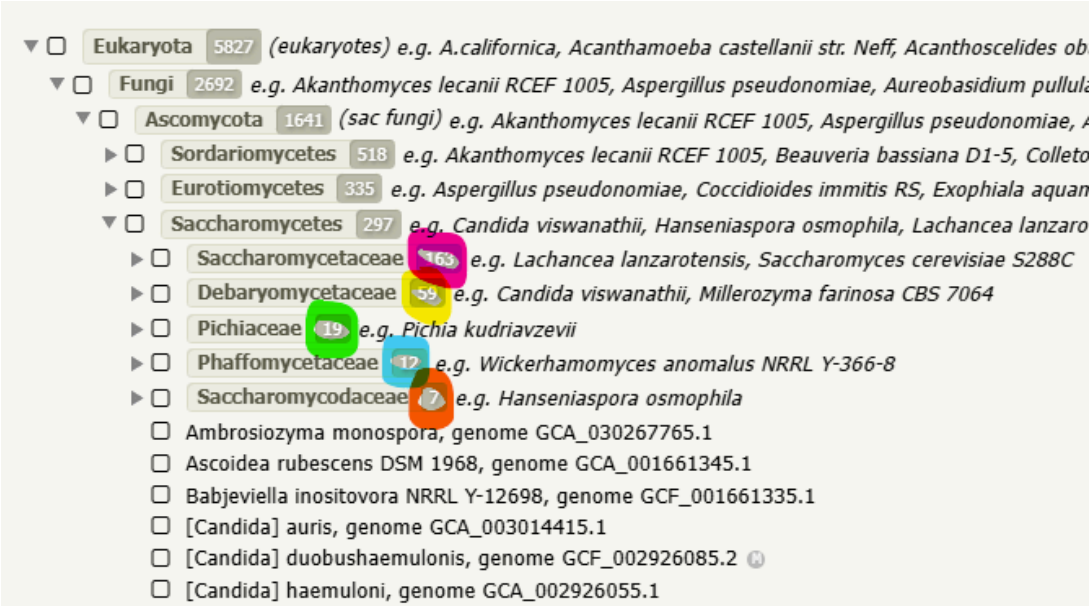
- Checking other Saccharomyces species/assemblies (scroll down the page to find them) at OrthoDB in the same orthogroup, [10267031at2759], reveals that no species of Saccharomycetaceae nor any species of Saccharomycodaceae seem to have an orthologue of Eukaryotic translation initiation factor 3 subunit M, there are only orthologues in Debaryomycetaceae, Pichiaceae, and Phaffomycetaceae



- When we look at all the species/assemblies included in OrthoDB v12 …
  - Click on the 

button to browse the entire tree at OrthoDB

- We can see that there are a total of 297 Saccharomyces species/assemblies included in OrthoDB v12, with **163 Saccharomycetaceae, 59 Debaryomycetaceae, 19 Pichiaceae, 12 Phaffomycetaceae, and 7 Saccharomycodaceae**



- Therefore we have the following scenario of orthologues identified:

Lineage	Total Species/Assemblies	Species with orthologues	Species missing orthologues
Saccharomyces	297	127	170
Saccharomycetaceae	163	0	all
Debaryomycetaceae	59	58	1
Pichiaceae	19	18	1
Phaffomycetaceae	12	12	none
Saccharomycodaceae	7	0	all

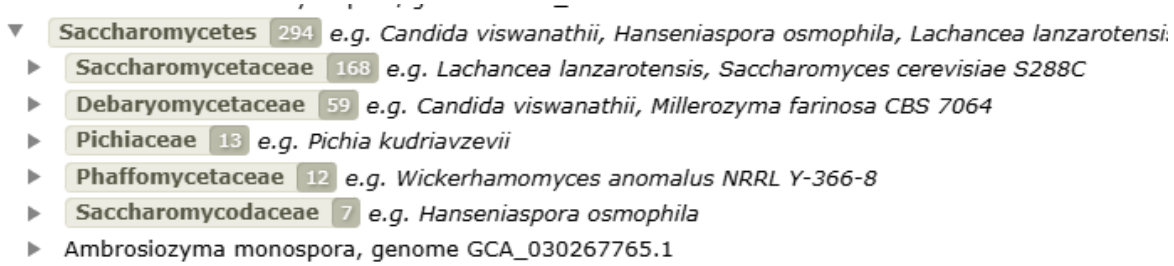
Question: What could this scenario suggest about the evolution of this Eukaryotic translation initiation factor 3 subunit M in Saccharomyces fungi?

- Following the same investigation of apparently missing BUSCO 25498at2759, which groups eukaryotic translation initiation factor 3 subunit F, we find the following scenario for the identified orthologues

Lineage	Total Species/Assemblies	Species with orthologues	Species missing orthologues
Saccharomyces	297	134	163
Saccharomycetaceae	163	0	all
Debaryomycetaceae	59	59	none
Pichiaceae	19	19	none
Phaffomycetaceae	12	12	none
Saccharomycodaceae	7	0	all

- One of the other apparently missing BUSCOs are the group of translation initiation factor 3 subunit L orthologues, 15082at2759, so for the genome assembly of *Saccharomyces jurei* we see that amongst the apparently missing BUSCOs we have subunits F, L, and M that could not be identified by the BUSCO assessment

- Contrast these translation initiation factor 3 subunits with the example of the apparently missing BUSCO 10261918at2759 (a slicing factor), which is found in all Saccharomycetaceae species/assemblies



Question: What can you conclude from this investigation about the putative missing translation initiation factor 3 subunits compared to the putative missing splicing factor?

⇒ Could it be possible that the missing translation initiation factor 3 subunits are in fact the result of a true evolutionary loss in Saccharomycetaceae?

⇒ What about the splicing factor, does this seem to be a true evolutionary loss? Why?

⇒ See the next page for insights into the evolution of translation initiation factor 3 subunits in Saccharomycetaceae

- The loss of several translation initiation factor 3 subunits from Saccharomyces cerevisiae has been recognised in the literature: D, E, F, H, K, L, and M
- Observing that F, L, and M from the BUSCO assessments of *Saccharomyces jurei* were also missing from all other Saccharomyces species/assemblies included in OrthoDB strongly supports the loss of these genes in their common ancestor
- Translation initiation factor 3 (eIF3) has been considered the largest and the most complex of all eIFs ever since its first isolation — The *Saccharomyces cerevisiae* comprises five core essential subunits: a/TIF32, b/PRT1, c/NIP1, i/TIF34, & g/TIF35

Table 1.
Overview of eIF3 subunits and of the eIF3-associated factor eIF3j across species

Subunit	Domains	*S. cerevisiae*			*S. pombe*			*N. crassa*			*A. thaliana*			*H. sapiens*		
		Named	M.W. (kDa)	Essential[a]	Named	M.W. (kDa)	Essential[b]	Named	M.W. (kDa)	Essential[b]	Named	M.W. (kDa)	Essential	named	M.W. (kDa)	Essential[a]
eIF3a	PCI, Spectrin HLD (yeast)	TIF32	110.3	E	p107	107.1	E	p110	120.2	E	p114	114.3	?	p170	166.6	E
eIF3b	WD40, RRM	PRT1	88.1	E	p84	84.0	E	p90	85.6	E	p82	81.9	?	p116	92.5	E
eIF3c	PCI	NIP1	93.2	E	p104	104.4	E	p93	98.4	E	p110	103.0/91.7	?	p110	105.3	E
eIF3d	Cap-binding pocket?	-	-	-	MOE1	62.6	N	eIF3d	65.0	E	p66	66.7	?	p66	64.0	E
eIF3e	PCI	-	-	-	INT6	57.1	N	INT6	51.1	N	p48	51.8	E*	p48	52.2	E
eIF3f	MPN	-	-	-	CSN6	33.3	E	eIF3f	39.7	E	p32	31.9	E**	p47	37.6	E
eIF3g	RRM, Zn finger	TIF35	30.5	E	TIF35	31.5	E	p33	32.4	E	eIF3g	32,7/35,7	?	p44	35.6	E
eIF3h	MPN	-	-	-	p40	39.8	N	eIF3h	40.4	N	p38	38.4	E***	p40	39.9	N
eIF3i	WD40	TIF34	38.7	E	SUM1	36.8	E	TIF34	38.8	E	p36	36.4	?	p36	36.5	E
eIF3k	PCI	-	-	-	-	-	-	p25	26.8	N	p25	25.7	?	p28	25.1	N
eIF3l	PCI	-	-	-	-	-	-	eIF3l	54.5	N	eIF3l	60.2	?	p67	66.7	N
eIF3m	PCI	-	-	-	CSN7B	45.1	E	eIF3m	49.7	E.	eIF3m	46.8	?	GA17	42.5	E
associated factor eIF3j	-	HCR1	29.6	N	p35	30.5	N	HCR1	30.3	N	eIF3j	25.5	?	p35	29.1	N

Congratulations!

Answers & further resources

Question: What is the "Scaffold N50" and what does it mean?

Scaffold N50 is the length N such that 50% of the total assembly length is contained in scaffolds of length ≥ N. See: https://en.wikipedia.org/wiki/N50,_L50,_and_related_statistics

Question: What other "gene finding" approaches are possible to use with BUSCO?

Miniprot pipeline (default for eukaryota) - Miniprot is not a gene predictor, but a gene mapper, and uses a reference protein database to map proteins to the genome.
Metaeuk pipeline - Designed for eukaryotic metagenomes, Metaeuk is a fast and accurate gene predictor that uses a reference protein database to predict genes.
Augustus pipeline - Augustus is a widely used gene predictor for eukaryotic genomes. It is the default gene predictor for eukaryotic genomes in BUSCO v4.0.0 and earlier.

Question: Why would you want to use a more specific (Saccharomycetaceae, family level) or a less specific (fungi, kingdom level; or Eukaryota, domain level) lineage dataset for your BUSCO evaluations?

More specific levels have lineage datasets containing more BUSCO groups because the species included are more closely related, i.e. a shorter time to their last common ancestor. With a larger set of BUSCOs with which to perform the assessments the resolution of the results is much higher. However, larger datasets mean longer compute times, so if time is a key factor (e.g. you are planning to assess many genomes) then a less specific dataset could be a better choice for your analyses. Importantly, if you wish to compare results across species is is necessary to use the same lineage dataset, i.e. a lineage dataset that is as old as or older than the last common ancestor of the species you wish to compare.

Question: What is the hmmsearch step doing?

The hmmsearch step searches a profile (and HMM, or hidden Markov Model profile) against a sequence database. It is part of the HMMER suite of biosequence analysis using profile hidden Markov models. Here hmmsearch is comparing the predicted protein sequence to a library of HMMs for all BUSCO groups in the selected lineage dataset to score the match and determine if the protein sequence likely represents a true orthologue, and if it is long enough to be considered a complete orthologue.

Question: How many BUSCO groups in total are there in this eukaryota_odb12 lineage dataset? How many were found as complete? How many could not be found?

Total – 129 Total BUSCO groups searched
Complete – 110 Complete BUSCOs (C)
Missing – 11 Missing BUSCOs (M)

Question: Why might it be useful to have access to the sequences that have been predicted for the BUSCO genes found as part of your assessment?

One very practical example would be for use as part of a pipeline to build multiple sequence alignments to be used to infer the species phylogeny. Extracting all the sequences of the single-copy complete orthologues from in all the included species provides a useful dataset for phylogenomic reconstructions of species trees, either with consensus approaches using sets of gene trees or from concatenation approaches that combine all multiple sequence alignments into a single superalignment for phylogeny inference.

Question: What could this scenario suggest about the evolution of this Eukaryotic translation initiation factor 3 subunit M in Saccharomyces fungi?

It appears to suggest a complete loss of subunit M from both Saccharomycetaceae and Saccharomycodaceae. The evidence for Saccharomycetaceae is strong because it appears to be missing from all 163 species/assemblies included in OrthoDB. The evidence for Saccharomycodaceae is less strong as there are only a total of 7 species included in OrthoDB.

Question: What can you conclude from this investigation about the putative missing translation initiation factor 3 subunits compared to the putative missing splicing factor?

It appears that the translation initiation factor 3 subunits may be true gene losses (as supported by the literature too), while the splicing factor may be missing just from the *Saccharomyces jurei* genome assembly or BUSCO failed to find this gene in the assembly. This conclusion is supported by the fact that the translation initiation factor 3 subunits appear to be lost across the whole clade, while the splicing factor is found in all the Saccharomycetaceae present in OrthoDB.