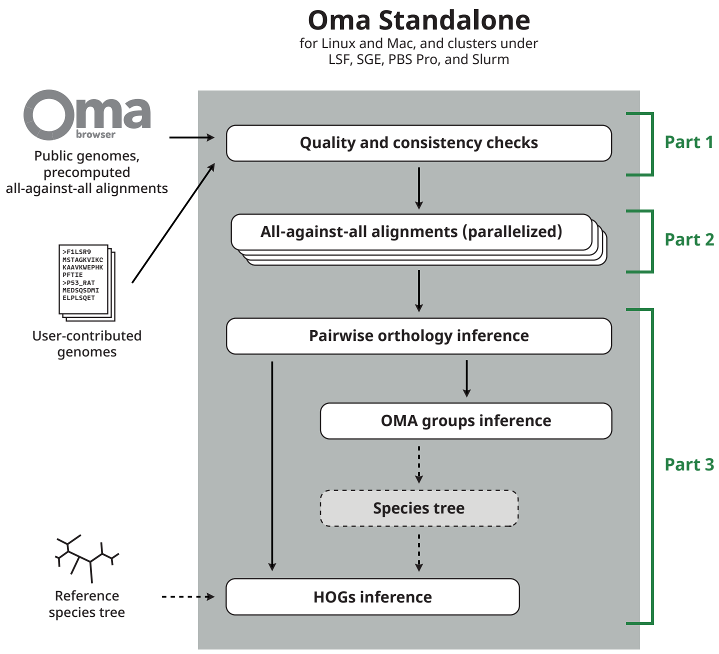
OMA STANDALONE CHEAT SHEET

OMA Standalone is a software for identifying homologs from complete genomes.

Use this cheat sheet if:
- You have your own annotated genomes
- You need to run OMA Standalone on an HPC cluster
- Your cluster uses the SLURM job scheduler

See https://omabrowser.org/standalone/ for many more details!

1. How it works:



Oma Standalone

for Linux and Mac, and clusters under LSF, SGE, PBS Pro, and Slurm

Public genomes, precomputed all-against-all alignments

User-contributed genomes

Reference species tree

Quality and consistency checks — **Part 1**

All-against-all alignments (parallelized) — **Part 2**

Pairwise orthology inference

OMA groups inference

Species tree

HOGs inference — **Part 3**

We will split the scripts to run OMA Standalone in 3 parts

2. Get setup:

- Connect to cluster: `ssh <user>@<login_node>`
- Download software:
  `wget https://omabrowser.org/standalone/OMA.latest.tgz`
- Alternatively, install with homebrew:
  `brew tap brewsci/bio; brew install oma`
- Untar: tar `xvzf OMA.latest.tgz`
- Change directory: `cd OMA.<latest>`
- `mkdir logs`

3. Prepare genomes:

Make sure all of your genomes meet the following requirements:
- fasta files (1 for each genome)
- protein sequences
- the name of each fasta file is the name of the genome
- all files must end in "`.fa`"
- copy all genome fasta files into `OMA.<latest>/DB/`

4. Edit parameters.drw:

- Can generally leave default parameters
- Optional, yet recommended: specify species tree
- If you choose to estimate species tree: must specify outgroup

*If you estimate the species tree, verify the results in the Output!**

5. Prepare scripts:

`oma_part1.sh`

This is the database conversion part

```
#!/bin/bash
#SBATCH --partition=<your_partition>
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=2GB
#SBATCH --job-name=oma1
#SBATCH --output=logs/oma1-%J.log
#SBATCH --export=None
#SBATCH --error=logs/oma1-%J.err

cd <full_path_to_OMA.latest>
./bin/oma -c
```

`oma_part2.sh`

This is the all-against-all part,
it is split into 500 parallelized jobs

```
#!/bin/bash
#SBATCH --array=1-500
#SBATCH --partition=<your_partition>
#SBATCH --time=2:00:00
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=2GB
#SBATCH --job-name=oma2
#SBATCH --output=logs/oma2-%A.%a.log
#SBATCH --export=None
#SBATCH --error=logs/oma2-%A.%a.err

cd <full_path_to_OMA.latest>
export NR_PROCESSES=500
./bin/oma -s -W 7000
if [[ "$?" == "99" ]] ; then
scontrol requeue \
${SLURM_ARRAY_JOB_ID}_${SLURM_ARRAY_TASK_ID}
fi
exit 0
```

`oma_part3.sh`

This is the orthology inference part

```
#!/bin/bash
#SBATCH --partition=<your_partition>
#SBATCH --time=1:00:00
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=50GB
#SBATCH --job-name=oma3
#SBATCH --output=logs/oma3-%J.log
#SBATCH --export=None
#SBATCH --error=logs/oma3-%J.err

cd <full_path_to_OMA.latest>
./bin/oma
```

6. Run scripts:

- Run 1 at a time. *For example:*
  `sbatch oma_part1.sh`
- `sacct` to check status
- If failed, check in `/logs`
  Additionally, check special characters (like quotes) to make sure they copied from pdf correctly
- Use `bin/oma-cleanup` if there are prematurly terminated all-against-all processes

7. Output:

`EstimatedSpeciesTree.nwk`
`EstimatedSpeciesTree.phyloxml`
**Verify estimated species tree with phylo.io*

`HierarchicalGroups.orthoxml`
All HOGs at different taxonomic levels

`HOGFasta/`
Root HOGs i.e. gene families at their most deep taxonomic level

`Map-SeqNum-ID.txt`
List of all genes

`OrthologousMatrix.txt`
`OrthologousGroupsFasta`
`OrthologousGroups.orthoxml`
`OrthologousGroups.txt`
Orthologous Groups, i.e. OMA Groups: useful for phylogenetic trees

`OrthologousPairs.orthoxml`
`PairwiseOrthologs/`
Pairwise orthologs; useful for comparing 2 genomes

`PhyleticProfileHOGs.txt`
`PhyleticProfileOMAGroups.txt`
Useful for phylogenetic profiling

Last updated 29 Nov. 2021

Questions? Contact us at:
https://biostars.org/tag/oma/
or contact@omabrowser.org