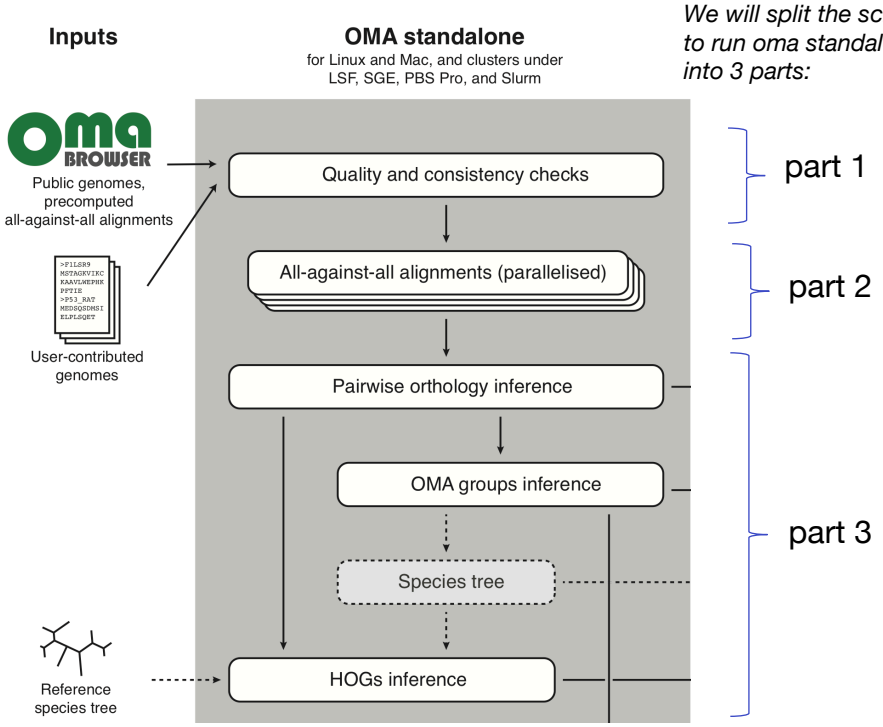
OMA Standalone CHEAT SHEET

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

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

Use this cheat sheet if:
-You have your own sequenced genomes
-You need to run it on the cluster (wally or axiom)

1. How it works



We will split the scripts to run oma standalone into 3 parts:

2. Get setup:

- Connect to cluster: ssh <user>@wally-front1.unil.ch
- 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 meeting 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:

- Mostly can leave defaults
- Mandatory: specify outgroup species
- Optional: specify species tree (if not, the estimated species tree **must** be verified in the Output*!)

5. Prepare scripts:

oma_part1.sh

```
#!/bin/bash
#SBATCH --partition=wally
#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
```

This is the database conversion part

oma_part2.sh

```
#!/bin/bash
#SBATCH --array=1-500
#SBATCH --partition=wally
#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/%A.%a.log
#SBATCH --export=None
#SBATCH --error=logs/%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
```

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

oma_part3.sh

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

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

This is the orthology inference part

6. Run scripts:

- Run 1 at a time!
- sbatch oma_part1.sh
- sacct to check status
- If failed, check in /logs

7. Output:

**Verify estimated species tree with phylo.io*

- EstimatedSpeciesTree.nwk
- EstimatedSpeciesTree.phyloxml

all HOGs at diff. taxonomic levels

- HierarchicalGroups.orthoxml
- HOGFasta/ — *Root HOGs i.e. gene families*

- Map-SeqNum-ID.txt
- 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*

Questions? natasha.glover@unil.ch