# Gene family evolution

**By Chema Martin**

This pipeline describes the steps to conduct orthogroup construction and explore protein domain evolution with a given set of proteomes, as it was done for the analysis of *Owenia* genome.

# Step 1. Data collection

The first step is to choose the species to include in the analysis and download the genome and annotation. It is important to write down which version of the genome/annotation is used, as it needs to be included as Materials & Methods in any manuscript and ensures future reproducibility of the analysis.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Species** | **Clade** | **Genome/Annotation** | **Source** | **No NR proteins** |
| *N. vectensis* | Cnidaria | Nematostella\_vectensis.ASM20922v1.46.gff3 | Ensembl | 24688 |
| *S. kowalevskii* | Hemichordata | SkowalevskiiJGIv3.0.longestTrs.gff3 | OIST | 34238 |
| *S. purpuratus* | Echinodermata | Strongylocentrotus\_purpuratus.Spur\_3.1.46.gff3 | Ensembl | 21231 |
| *B. lanceolatum* | Chordata | Branchiostoma\_lanceolatum.BraLan2.46.gff3 | Ensembl | 35732 |
| *L. oculatus* | Chordata | Lepisosteus\_oculatus.LepOcu1.dna.toplevel.fa | Ensembl | 18328 |
| *H. sapiens* | Chordata | Homo\_sapiens.GRCh38.dna.toplevel.fa | Ensembl | 19501 |
| *S. maritima* | Ecdysozoa | Strigamia\_maritima.Smar1.46.gff3 | Ensembl | 14977 |
| *T. castaneum* | Ecdysozoa | Tribolium\_castaneum.Tcas5.2.46.gff3 | Ensembl | 16557 |
| *O. fusiformis* | Annelida | Owenia\_annotation\_v250920.gff3 | Own | 30840 |
| *C. teleta* | Annelida | Capitella\_teleta.Capitella\_teleta\_v1.0.46.gff3 | Ensembl | 32115 |
| *D. gyrociliatus* | Annelida | Dgy\_annot.gtf (converted to gff with gffread) | Own | 14182 |
| *L. luymesi* | Annelida | Lamellibrachia\_gitRepo.gff3 | GitHub repo of paper | 39303 |
| *H. robusta* | Annelida | Helobdella\_robusta.Helro1.46.gff3 | Ensembl | 23359 |
| *Eisenia andrei* | Annelida |  |  |  |
| *S. mediterranea* | Platyhelminthes | smes\_v2\_hconf\_SMESG.gff3 | PlanMine | 30885 |
| *N. geniculatus* | Nemertea | nge\_genome\_v2.0.gff | OIST | 43054 |
| *L. anatina* | Brachiopoda | lan\_genome\_v2.0.gff | OIST | 29771 |
| *P. australis* | Phoronida | pau\_genome\_v2.0.gff | OIST | 20393 |
| *C. gigas* | Mollusca | Crassostrea\_gigas.oyster\_v9.46.gff3 | Ensembl | 26089 |
| *L. gigantea* | Mollusca | Lottia\_gigantea.Lotgi1.46.gff3 | Ensembl | 23256 |
| *M. yessoensis* | Mollusca | GCF\_002113885.1\_ASM211388v2\_genomic.gff | NCBI | 24434 |
| *H. miamia* | Acoela | Hofstenia\_miamia.HmiaM1.46.gff3 | Ensembl | 22454 |

# Step 2: Generate Non-Redundant proteomes

To generate a non-redundant proteome, we can use [AGAT](https://github.com/NBISweden/AGAT), a suite of scripts to manipulate GFF files. This suite will allow us to merge the different transcripts into one, retaining the longest isoform. We can then use gffread to extract the protein sequences with the -V flag to remove all those with premature STOP codons. AGAT might be a bit difficult to install and I only managed to make it work on my desktop iMac (not Apocrita). Try the conda approach first (recommended by developer):

conda install -c bioconda agat

#Merge loci

agat\_sp\_gxf\_to\_gff3.pl --gff Owenia\_annotation\_v102519.gff3 --merge\_loci -o Owenia\_lociMerged.gff

#Generate the non-redundant gff picking the longest isoform

agat\_sp\_keep\_longest\_isoform.pl --gff Owenia\_lociMerged.gff -o Owenia\_lociMerged\_longestIsoform.gff

#Extract the proteins

gffread -g Owenia\_softmasked\_v072019.fa -V -y Owenia\_lociMerged\_longestIsoform.fa Owenia\_lociMerged\_longestIsoform.gff

* agat\_sp\_gxf\_to\_gff3.pl replaced by [agat\_convert\_sp\_gxf2gxf](https://github.com/NBISweden/AGAT/wiki/agat_convert_sp_gxf2gxf) since AGAT v0.3.0

# Step 3: Prepare your proteome

Once you have your non-redundant proteomes, and in order to handle better the data once orthogroups are inferred, rename each proteome to have an easy name related to the species it comes from (e.g. the *Owenia* proteome could be Ofus [*Owenia fusiformis*]) and add that tag to the beginning of each sequence (e.g. >Ofus\_GENEID). You can do that easily with a text editor. It will make finding genes of different species much easier.

# Step 4: OrthoFinder – Orthogroup inference

We use [OrthoFinder](https://github.com/davidemms/OrthoFinder) to infer orthology groups (~gene families). OrthoFinder uses sequence similarity (e.g. BLAST) and MCL clustering to group related genes from different species that might share a common origin. OrthoFinder is actively developed and it is always good to check and update to the latest version.

## 4.1 Finding best parameters to OrthoFinder

We can tweak two parameters to improve orthogroup inference: the inflation value (granularity of the inference, high inflation value means more orthogroups, smaller, as you restrict similarity more [e.g. you differentiate between WNT1 and WNT2], while lower inflation values mean less and larger orthogroups [e.g. all WNT genes go together, as a WNT orthogroup]). The other parameter is the sequence similarity engine OrthoFinder uses: BLASTP, DIAMONDBLAST, MMseqs2. Blast is very slow, and with such large proteomes and number of species, it will take days, so I discourage its use. Diamond is faster, but it could be slightly less accurate. [MMseqs2](https://www.nature.com/articles/nbt.3988) uses a different approach (Smith-Waterman alignments) and is also faster and more sensitive.

MMseqs2 is not in the cluster and requires installation:

conda create --yes --name MMseqs2 mmseqs2

Diamond is used by default, but the -S flag of OrthoFinder can change the behaviour to use MMseqs2 (loading the MMseqs2 environment) and the -I flag can change the inflation value. Include many cores and RAM to speed up the process:

#!/bin/bash

#$ -pe smp 12

#$ -l h\_vmem=1G

#$ -l h\_rt=48:0:0

#$ -cwd

#$ -j y

#This is the default with diamond and inflation 1.5

orthofinder -f 00-NonRed\_Proteomes -t 12

This one changes to MMseq2 and inflation 2:

#!/bin/bash

#$ -pe smp 12

#$ -l h\_vmem=2G

#$ -l h\_rt=120:0:0

#$ -cwd

#$ -j y

#Load anaconda and activate MMseqs2 environment

module load anaconda3/

source activate MMseqs2

#Run orthofinder with mmseqs and inflation of 2

orthofinder -f 00-NonRed\_Proteomes -t 12 -S mmseqs -I 2

source deactivate MMseqs2

Check how many genes are included in orthogroups:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Total % genes in OG | *Owenia* (%genes in OG) | *Capitella* (%genes in OG) | *Helobdella* (%genes in OG) |
| Diamond, I=1.5 | 470189 genes (86.2% of total) to 37197 OG | 84.7 (11203 OGs) | 88.7 (11243 OGs) | 79.5 (7740 OGs) |
| Diamond, I=2 | 470058 genes (86.2% of total) to 46819 OG | 84.7 (12318 OGs) | 88.6 (12577 OGs) | 79.5 (8413 OGs) |
| MMseqs2, I=1.5 | 480208 genes (88.0% of total) to 37870 OG | 86.4 (11140 OGs) | 90.3 (11219 OGs) | 80.5 (7723 OGs) |
| **MMseqs2, I=2** | **480086 genes (88.0% of total) to 46863 OG** | **86.4 (12223 OGs)** | **90.3 (12439 OGs)** | **80.4 (8299 OGs)** |

Using MMseqs2 and an inflation value of 2, we generate more orthogroups and include more genes.

## 4.2 Running OrthoFinder with a defined species tree

By default, OrthoFinder will estimate a species tree based on the data. However, we do know the phylogenetic relationship of these species, and thus we can force OrthoFinder to use the one we know is real:

This is the one used for Owenia:

(Nvec,(Hmia,(((Skow,Spur),(Blan,(Locu,Hsap))),((Smar,Tcas),(Smed,((Lgig,(Cgig,Myes)),((Ofus(Dgyr,(Lluy,(Ctel,Hrob)))),(Ngen,(Paus,Lana)))))))));

We can save this newick tree in a file named SpeciesTree.nwk, re-use the MMseq2 pre-compiled searches, and re-run the analysis using the inflation of 2:

#!/bin/bash

#$ -pe smp 1

#$ -l h\_vmem=40G

#$ -l h\_rt=10:0:0

#$ -cwd

#$ -j y

orthofinder -b ./00-NonRed\_Proteomes/OrthoFinder/Results\_Mar13\_2/WorkingDirectory/ -I 2 -s SpeciesTree.nwk

OrthoFinder generates lots of files and stats and it is worth taking some time exploring it. The key files are the Orthogroups itself (and the Orthogroups.GeneCount.tsv file) and the Comparative\_Genomics\_Statistics folder, with stats of orthogroup size, % of incorporated genes, species-specific orthogroups, etc.

# Step 5: Gene family gain/loss/expansions

Run Ferdi’s script and analyse it according to Martin-Duran et al Nature Eco Evo (DOI: <https://doi.org/10.1038/s41559-020-01327-6>) and a script available in <https://github.com/fmarletaz/comp_genomics>

conda install -c etetoolkit ete3

conda install numpy scipy statsmodels

conda install -c conda-forge tqdm

conda install -c anaconda ipykernel

python -m ipykernel install --user

conda install -c conda-forge notebook

conda install -c conda-forge ipywidgets

jupyter notebook # this will launch the browser app to actually run the script