**Cognitive gene analysis**

Four Python scripts are presented for extracting and analyzing cognitive-related genes from the datasets containing all the protein-coding sequences of a species. It involves filtering isoforms, and identifying orthogroups associated with cognitive functions.

## **Overview**

This repository contains four Python scripts that form as part of the pipeline for cognitive gene analysis:

1. **extract\_genes.py** - Extract genes associated with cognitive GO terms from EggNOG-mapper annotations
2. **filter\_isoforms.py** - Filter FASTA sequences to retain one isoform per gene
3. **isoform\_check.py** - Check and analyze isoform counts in FASTA files
4. **make\_cognitive\_orthogroup.py** - Filter OrthoFinder results for cognitive gene orthogroups

## **Requirements**

### Python Version:

* Python 3.7 or higher

### Dependencies

* **BioPython** - Required for filter\_isoforms.py and isoform\_check.py  
  Can be installed by: pip install biopython

### **Input Data Requirements**

* EggNOG-mapper annotation files (.emapper.annotations)
* FASTA files (protein sequences)
* OrthoFinder output files (Orthogroups.tsv)

## **Scripts Description**

### **1. extract\_genes.py**

Extracts genes associated with cognitive-related Gene Ontology (GO) terms from EggNOG-mapper annotations.

**Features:**

* Minimal dependencies
* Customizable GO term list
* Generates per-GO-term gene lists and summary TSV

**Default Cognitive GO Terms:**

* GO:0007613 - Memory
* GO:0007611 - Learning
* GO:0099536 - Synaptic signaling
* GO:0007420 - Brain development
* GO:0007399 - Nervous system development
* GO:0050804 - Modulation of chemical synaptic transmission
* GO:0007268 - Synaptic transmission

**Usage:**

Bash command: python extract\_cognitive\_genes.py -a annotations.emapper.annotations -o output\_dir

*# With custom GO terms file*

python extract\_cognitive\_genes.py -a annotations.emapper.annotations \

-o output\_dir --go-file custom\_go\_terms.tsv

**Arguments:**

* -a, --annotations: EggNOG-mapper annotations file (TSV format)
* -o, --outdir: Output directory for results (default: cognitive\_go\_outputs)
* --go-file: Optional custom GO term mapping file (2-column: GO\_ID, label)
* --go-col: Column index for GO terms (default: 9)
* --gene-col: Column index for gene IDs (default: 0)

**Output:**

* Individual gene lists per GO term: cognitive\_genes\_GO:XXXXXXX\_label.txt
* Summary TSV: cognitive\_genes\_summary.tsv

### **2. filter\_isoforms.py**

Filters FASTA files to retain only one isoform per gene, supporting both Ensembl and NCBI RefSeq formats.

**Features:**

* Multiple filtering strategies (longest, first, curated)
* Supports gzipped input files
* Handles Ensembl and RefSeq header formats
* Optional isoform mapping output

**Usage:**

bash

*# Default strategy (longest sequence)*

python filter\_isoforms.py -i proteins.faa -o filtered.faa

*# Using curated strategy (prioritizes manually curated sequences)*

python filter\_isoforms.py -i proteins.faa.gz -o filtered.faa --strategy curated

*# With mapping file output*

python filter\_isoforms.py -i proteins.faa -o filtered.faa --write-map gene\_map.tsv

**Arguments:**

* -i, --input: Input FASTA file (can be gzipped)
* -o, --output: Output filtered FASTA file
* --strategy: Selection strategy (longest, first, curated)
* --write-map: Optional TSV output mapping genes to keep isoforms

**Strategies:**

* **longest**: Keep the isoform with the longest sequence
* **first**: Keep the first isoform encountered
* **curated**: Prioritize curated accessions (NP\_ > XP\_ for RefSeq, ENSP for Ensembl)

### **3. isoform\_check.py**

Analyzes FASTA files to count isoforms per gene and generate statistics.

**Features:**

* Counts isoforms per gene
* Reports genes with multiple isoforms
* Shows top genes by isoform count

**Usage:**

bash

python isoform\_check.py -i transcripts.fasta

*# Show top 20 genes with most isoforms*

python isoform\_check.py -i transcripts.fasta -n 20

**Arguments:**

* -i, --input: Input FASTA file
* -n, --top: Number of top genes to display (default: 10)

**Output:**

* Total gene count
* Total isoform count
* Number of genes with multiple isoforms
* List of top genes by isoform count

### **4. make\_cognitive\_orthogroup.py**

Filters OrthoFinder orthogroups to identify those containing cognitive genes.

**Features:**

* Filters orthogroups based on cognitive gene list
* Handles various gene ID formats (Ensembl, RefSeq)
* Preserves original orthogroup structure

**Usage:**

bash

*# Edit the script to set file paths, then run:*

python make\_cognitive\_orthogroup.py

**Required Files (edit paths in script):**

* cognitive\_genes.txt: List of cognitive genes (one per line)
* Orthogroups.tsv: OrthoFinder output file
* cognitive.tsv: Output file name

**Output:**

* Filtered TSV containing only orthogroups with cognitive genes

## **Typical Workflow**

1. **Extract cognitive genes from functional annotations:** bash  
   python extract\_cognitive\_genes.py -a species.emapper.annotations -o cognitive\_outputs
2. **Check isoform distribution in proteome:** bash  
   python isoform\_check.py -i proteome.fasta
3. **Filter to one isoform per gene:** bash  
   python filter\_isoforms.py -i proteome.fasta -o proteome\_filtered.fasta --strategy longest
4. **Run OrthoFinder on filtered proteomes** (not included in this repo):  
    bash  
   orthofinder -f proteomes\_directory/

**Filter orthogroups for cognitive genes:** bash  
*# Combine all cognitive gene lists*

cat cognitive\_outputs/cognitive\_genes\_\*.txt | sort -u > cognitive\_genes.txt

*# Edit make\_cognitive\_orthogroup.py with correct file paths*

1. python make\_cognitive\_orthogroup.py

## **Notes**

* The scripts are designed to handle various header formats from different databases
* Gene ID extraction uses pattern matching and may need adjustment for non-standard formats
* For Ensembl headers, look for patterns like gene:ENSG... or gene\_symbol:...
* For RefSeq headers, look for patterns like gene=... or GN=...

## **Troubleshooting**

### **Common Issues**

1. **No genes extracted from headers:**
   * Check FASTA header format
   * Modify regex patterns in extract\_gene() function if needed
2. **BioPython not found:** bash  
   pip install biopython
3. **Memory issues with large files:**
   * The scripts process files sequentially to minimize memory usage
   * For very large files, consider splitting input files
4. **Incorrect column indices for EggNOG:**
   * Check your EggNOG-mapper version
   * Adjust --go-col and --gene-col parameters as needed