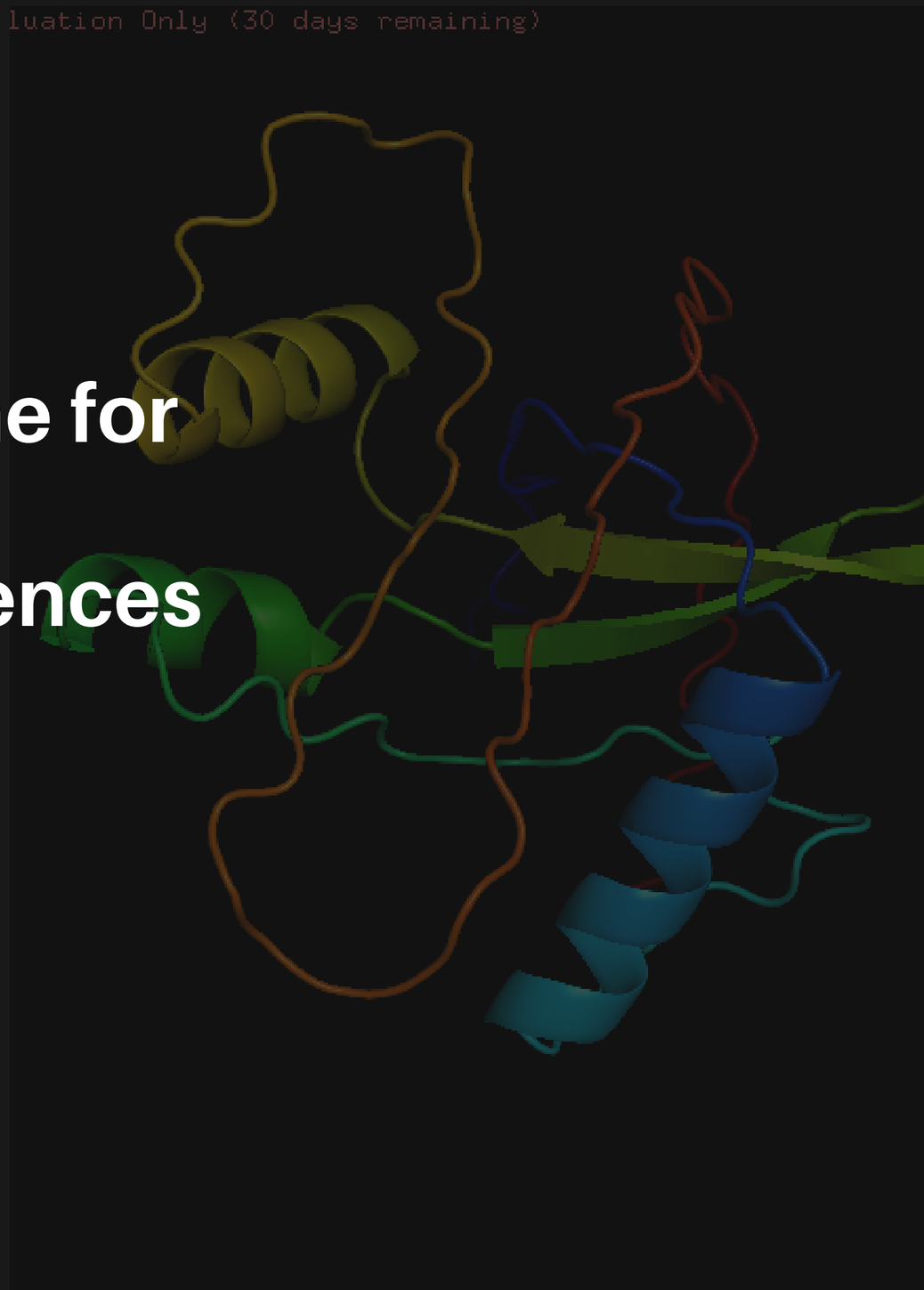

A Snakemake-Based Pipeline for Comprehensive Functional Annotation of Protein Sequences

Author: João Ferreira
Institution: University of Minho
Supervisors: Andreia Salvador, Maria Fernanda Vieira

Objectives

- Development of a protein annotation pipeline.
  - Tool assessment.
  - Selection, integration and implementation.
  - Pipeline evaluation.

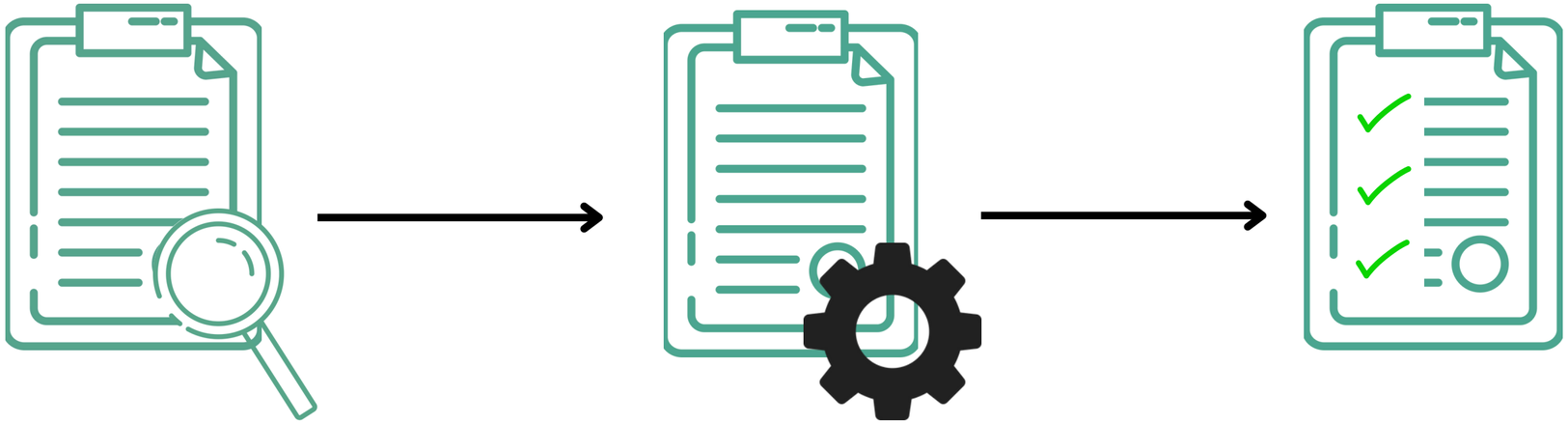

Figure 1 - Illustrative representation of the pipeline development process: tool assessment, selection, integration and implementation and evaluation of the pipeline.

Motivation

Why is this important?

- The explosion of genomic data has led to millions of uncharacterized proteins.
- Complementary methods improve coverage and confidence:
- Combining tools enhances annotation robustness, especially for hypothetical proteins.

Sequence homology tools

Tool	Annotation Strategy	Databases Used
ReCOGnizer	Matches conserved functional domains	CDD(Conserved Domains Database)
UPIMAPI	Compares full-length protein sequences to curated hits	UniProtKB
EggNOG-mapper	Transfers function via orthologs	eggNOG

Table 1 - Sequence homology tools included in the pipeline.

Protein structure tools

Tool	Annotation Strategy	Databases Used
FoldSeek	Finds structural homologs (3D match)	Protein Data Bank (PDB)
DeepFRI	Predicts GO from 3D graphs	Gene Ontology (trained), PDB

Table 2 - Protein structure tools included in the pipeline.

Machine Learning tools

Tool	Annotation Strategy	Databases Used
DeepGO-SE	Learns GO from sequence embeddings	
CLEAN	Predicts EC using contrastive learning	UniProtKB(training)
ProteInfer	Classifies function via CNN on sequence	

Table 3 - Machine learning tools included in the pipeline.

Workflow management system - snakemake

- Automates workflows using simple, readable rules
- Tracks inputs, outputs, and software versions
- Supports modular design
- Built-in support for Conda, Docker, Singularity
- Parallel and scalable: runs on local, cluster, or cloud
- Reproducible and portable for collaborative science

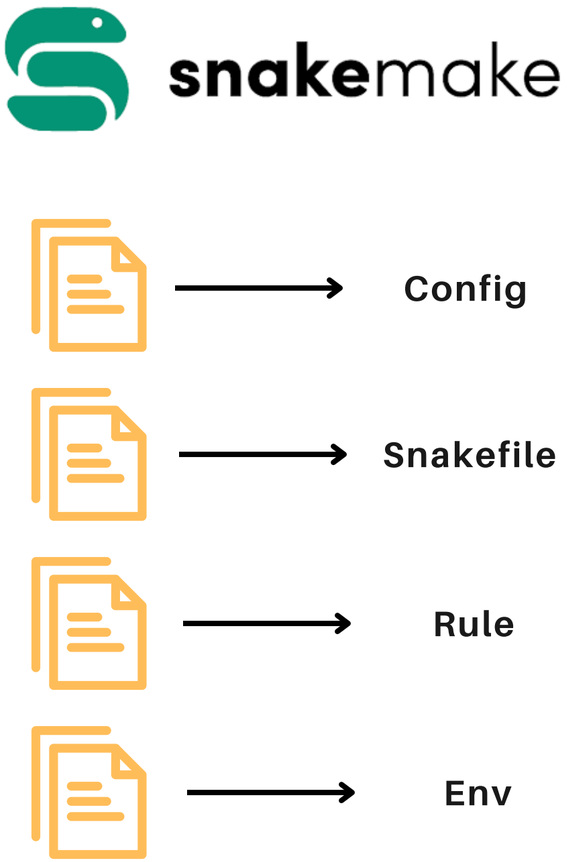

Figure 2 - Key components of a Snakemake project

Workflow management system - snakemake

Rule file

```
rule download_foldseek_pdb:
    output:
        "data/databases/foldseek_db/pb"
    conda:
        "../../envs/module_2/foldseek.yaml"
    shell:
        """
        mkdir -p data/databases/foldseek_db
        rm -rf data/databases/foldseek_db/pb
        cd data/databases/foldseek_db
        foldseek databases PDB pb tmp
        """
```

```
rule foldseek_easy_search:
    input:
        "results/colabfold",
        "data/databases/foldseek_db/pb"
    output:
        "results/foldseek/output_easy_search.m8"
    conda:
        "../../envs/module_2/foldseek.yaml"
    shell:
        """
        mkdir -p results/foldseek/tmp
        foldseek easy-search results/colabfold data/databases/foldseek_db/pb \
            results/foldseek/output_easy_search.m8 results/foldseek/tmp --threads 10
        """
```

Figure 3- Anatomy of a Snakemake rule.

Workflow management system - snakemake

Env file

```yaml
name: deepfri
channels:
  - conda-forge
  - bioconda
  - defaults
dependencies:
  - python=3.7
  - pip
  - pip:
      - tensorflow==2.3.1
      - scikit-learn==0.23.1
      - biopython==1.76
      - networkx==2.4
      - numpy==1.18.5
      - matplotlib
      - seaborn
```

Snakefile

```
# Load config
configfile: "config/config.yaml"
config["threads"] = workflow.cores
# Module_1
include: "workflow/rules/module_1/recognizer.smk"
include: "workflow/rules/module_1/eggnogmapper.smk"
#include: "workflow/rules/module_1/upimapi.smk"
# Module_2
include: "workflow/rules/module_2/colabfold.smk"
include: "workflow/rules/module_2/deepfri.smk"
include: "workflow/rules/module_2/foldseek.smk"
#module3
include: "workflow/rules/module_3/clean.smk"

rule all:
    input:
        # Module_1
        "results/recognizer_results/reCOGnizer_results.tsv",
        "results/eggnogmapper_results/eggnog_mapper_results.emapper.annotations",
        # "results/upimapi_results.tsv",

        # Module_2
        "results/colabfold",
        "data/deepfri_src/.models_downloaded",
        "data/deepfri_src/.requirements_done",
        "data/deepfri_src/.testte",
        "results/foldseek/output_easy_search.m8",
        "data/databases/foldseek_db/.pdb_ready",

        # Module_3
        "data/clean_src/.download_done",
        "data/clean_src/app/results/inputs/subset_f_maxsep.csv",
        "data/clean_src/app/.installed",
        "data/clean_src/app/.installed_esm",
        "data/clean_src/app/data/pretrained/.models"
```

Figure 4- Integration of environment and workflow logic in Snakemake.

Pipeline architecture

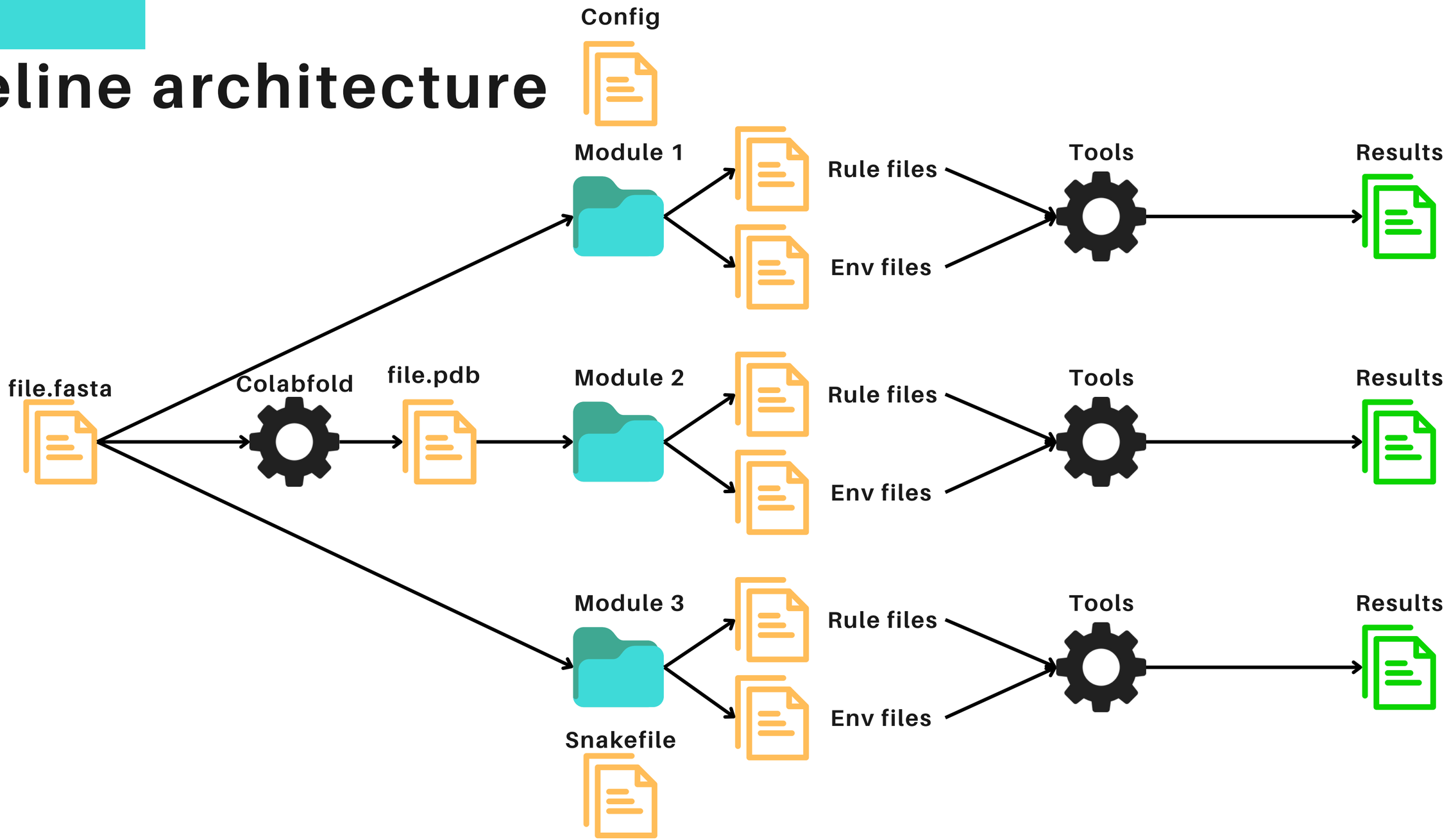

Figure 5- Illustrative representation of the pipeline architecture

Evaluation Strategy

- Assess on unannotated or poorly characterized proteins
- Compare outputs across methods
- Highlight strengths and weaknesses

Figure 6 - Workflow evaluation.

Results

Protein ID	Recognizer	eggNOG Description	Foldseek	Clean	DeepFRI
YP_010681523.1	Concanavalin A-like lectin/ glucanases superfamily	Concanavalin A-like lectin/ glucanases superfamily		EC:4.2.2.17 inulin class: Lyases	
YP_009217365.1	Dual specificity phosphatase, Predicted protein tyrosine phosphatase	COG1413 - (associated with regulatory domains) COG5350		EC:3.1.3.48 class: Hydrolases	
UYE95424.1	CDD:177403	Anti-CBASS protein Acb	7t27/7t26 - phage FBB1 anti-CBASS nuclease	EC:3.1.4.37 class: Hydrolases	

Table 4 - Comparison of results

- Consistent prediction: across reCOGnizer, CLEAN and eggNOGmapper.
- In most cases, FoldSeek did not return significant structural matches.
- DeepFRI returned low-confidence scores.

Conclusions

- The pipeline demonstrated that combining diverse annotation strategies increases reliability and functional coverage, especially for poorly characterized proteins.
- Tools converged on consistent annotations.
- In most cases CLEAN, eggNOGmapper and reCOGnizer presented significant results
- FoldSeek and DeepFRI were less effective probably due to limited databases.

Ongoing and Future Developments

- Add a general report with the results of all important tools and statistics.
- Integrate additional machine learning tools.
- Expand databases used by FoldSeek and DeepFRI.
- Test the pipeline with real experimental protein sequences obtained from raw data.