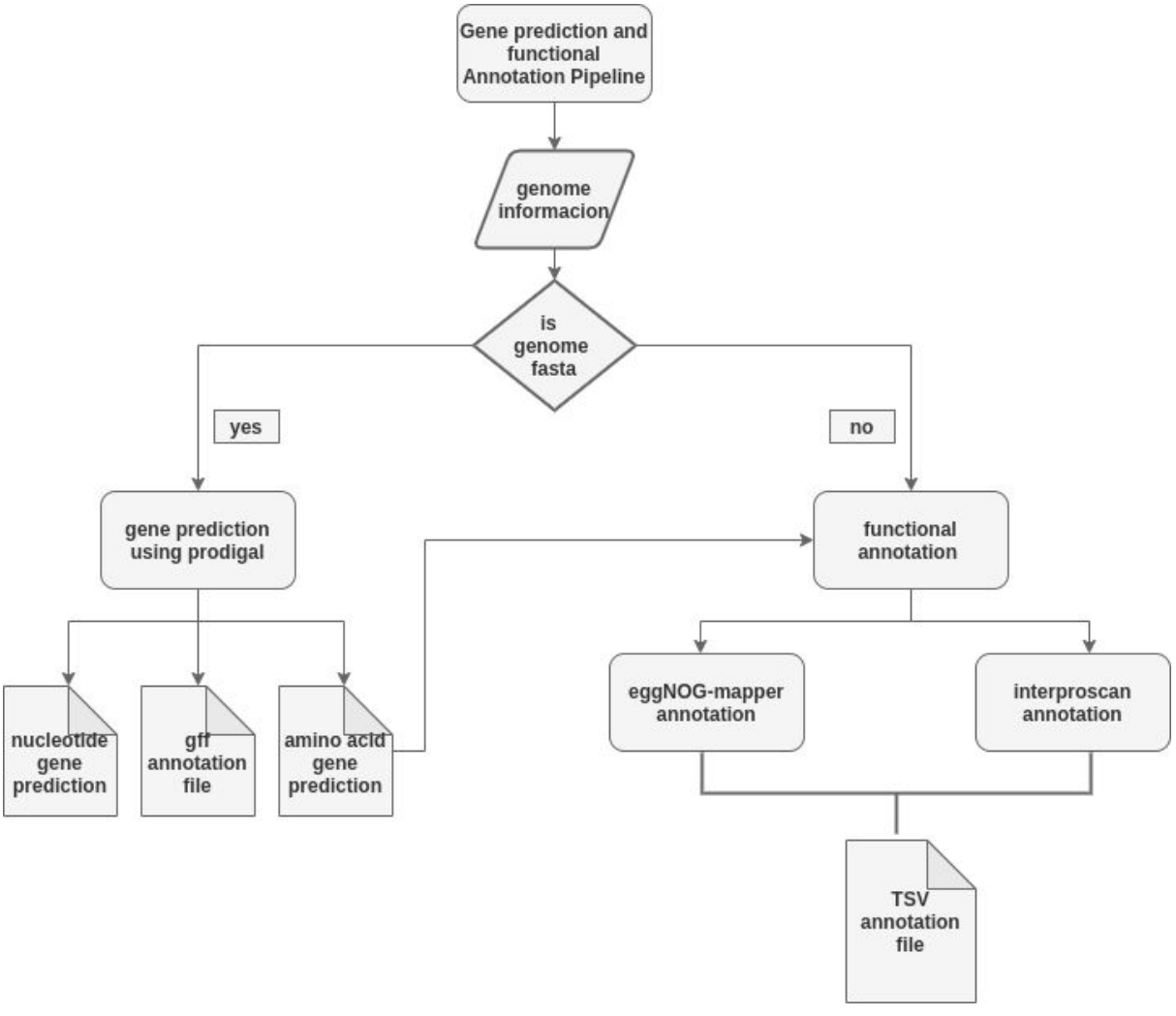
Gene prediction and functional Annotation Pipeline (GAP)

1. **Introduction**
2. **Dependencies**
3. **Installation**
4. **How to Run**
5. **Outputs**
6. **Contact and citation**
7. **Reference**

1. Introduction

The **G**ene prediction and functional **A**nnotation **P**ipeline (**GAP**) is a workflow who have as main aim the automatization of prediction and functional assignation of genes for microbial genomes.

GAP is divided into two modules. The first module for gene prediction, using prodigal[1] and the second module uses EggNOG 4.5[2] database to assign function by orthologs search using the EggNOG-mapper[3] and Interproscan[4] to detect homology based on domains. In figure 1, we schematize the pipeline workflow.



2. Dependencies

GAP is implemented as a multilanguage software, so it requires the prior installation of these languages on your computer and also requires the installation of several other tools for its correct operation. Below we list the previous requirements that must be installed in order to have an optimal functioning of this tool.

Perl installation:

Perl comes in the default repository of Ubuntu, thus no need to add any third-party repo. But, if it is not installed, carry out the following instructions

```
$ sudo apt-get update
$ sudo apt-get install perl
```

Python 3 installation:

Most factory versions of Ubuntu 18.04 or Ubuntu 20.04 come with Python pre-installed. Check your version of Python by entering the following:

```
$ python --version
```

If the revision level is lower than 3.7.x, or if Python is not installed, continue to the next steps.

```
$ sudo apt install software-properties-common
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt update
$ sudo apt install python3.8
```

R installation:

To install the latest stable version of R on Ubuntu 18.04, follow these steps:

```
$ sudo apt install apt-transport-https software-properties-common
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
E298A3A825C0D65DFD57CBB651716619E084DAB9
$ sudo add-apt-repository 'deb https://cloud.r-project.org/bin/linux/ubuntu
bionic-cran35/'
$ sudo apt update
$ sudo apt install r-base
$ R --version
```

Prodigal installation:

Prodigal (Prokaryotic Dynamic Programming Genefinding Algorithm) is a microbial (bacterial and archaeal) gene finding program, for his installation follow the next steps:

```
$ sudo apt-get update -y
$ sudo apt-get install -y prodigal
```

EggNOG-mapper installation:
eggNOG-mapper is a tool for fast functional annotation of novel sequences. It uses precomputed orthologous groups and phylogenies from the eggNOG database to transfer functional information from fine-grained orthologs only.
For installation of eggNOG-mapper follow the next instructions:

```
$ git clone https://github.com/eggnogdb/eggnog-mapper.git
$ cd eggnog-mapper
$ pwd  (copy the entry path)
```

Next go to home directory and type

```
$ nano .bashrc
```

Next type the following line in .bashrc file:

export PATH="$PATH:/directory/eggnog-mapper" (change “/directory/eggnog-mapper” for the path copied from pwd output)

Then save the file
Finally type

```
$  emapper.py --version
```

to check the version.

Interproscan installation:
InterPro is a database which integrates together predictive information about proteins' function from a number of partner resources, giving an overview of the families that a protein belongs to and the domains and sites it contains.
For the installation of interproscan follow the next steps:

```
$ git clone https://github.com/ebi-pf-team/interproscan.git
$ cd interproscan
$ pwd (copy the entry path)
```

Next go to home directory and type

```
$ nano .bashrc
```

Next type the following line in .bashrc file:

export PATH="$PATH:/directory/interproscan" (change “/directory/interproscan” for the path copied from pwd output)

Then save the file
Finally type

```
$ interproscan.sh  --version
```

to check the version.

NOTE: interproscan requires Java 11 to run. Before installing this tool verify your java version or install java

To add Panther model data to interproscan follow the instructions below:
Go to interproscan data folder

```
$ cd interproscan/data
```

Next download the Panther data using wget

```
$ wget ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/data/panther-data-14.1.tar.gz
$ wget \
ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/data/panther-data-14.1.tar.gz.md5
```

After both files were downloaded decompress panther data and it will be ready for use

```
$ tar -pxvzf panther-data-14.1.tar.gz
```

To verify that data is ready to be used type:

```
$ interproscan.sh
```

and go to the Available analyses section and check if the following message appear:
" *PANTHER (14.1) : The PANTHER (Protein ANalysis THrough Evolutionary Relationships) Classification System is a unique resource that classifies genes by their functions, using published scientific experimental evidence and evolutionary relationships to predict function even in the absence of direct experimental evidence.* "

If this message is not available, retry the installation of Panther data.

3. Installation

To install GAP you need follow the below instructions:

```
$ git clone https://github.com/lalomartinez/GAP.git
```

Next move to GAP bin folder and copy the path

```
$ cd GAP/bin
```

$ pwd bin (copy the entry path)

Next go to home directory and type

$ nano .bashrc

Next type the following line in .bashrc file:

export PATH="$PATH:/directory/GAP/bin" (change “/directory/GAP/bin” for the path copied from pwd output)

Then save the file.

Finally before to run GAP you need to change the path of required scripts in the main script for this, go to GenePredAnnotation_pipeline.pl and modify the following lines with the repository path.

---------- SCRIPTS REQUIRED

my $pscript = "/path_to/GenePredAnnot/bin/parse_interproannotation.py";
my $rscript = "/path_to/GenePredAnnot/bin/merge_annotations.R";

Save the file.

4. **How to run**

As we describe before GAP is a modular tool, this means that you have the option to run all analyzes or only run some steps of this workflow. In the following example we describe how to run the different analyzes of GAP.

Usage: perl GenePredAnnotation_pipeline.pl -i <FILE_RUTE_FASTA> -o <OUTPUT FOLDER> <OPTIONS>

-i option is a list of genomes
-o output folder

If you run:

$ perl GenePredAnnotation_pipeline.pl -h

display the following text :

---------- You selected help (-h) ----------

Usage: perl GenePredAnnotation_pipeline.pl -i <FILE_RUTE_FASTA> -o <OUTPUT FOLDER> <OPTIONS>

Options:

Main analyzes

-A Gene annotation (EggNOG mapper and InterproSCAN)
-P Gene prediction (Prodigal)

OTHER

-i Input list of genomes/proteomes
-o Output folder
-t|--threads Number of Threads
-h Print help
-v version

NOTES

1) If you only use annotation step, provide gene prediction in amino acid sequence
2) To run all analizes please select options -A -P
3) Always select threads when run annotation steps
4) To run this pipeline need to install Prodigal, eggNOG mapper and InterproSCAN in your computer

If you type:

$ perl GenePredAnnotation_pipeline.pl -v

the current version of GAP was shown.

If you run:

$ perl GenePredAnnotation_pipeline.pl -i genomes.list -o output_folder -A -P -t (number of threads)

you run all GAP workflow.
But if you run :

$ perl GenePredAnnotation_pipeline.pl -i genomes.list -o output_folder -P

you only run the gene prediction module.

On the other hand, if you run:

$ perl GenePredAnnotation_pipeline.pl -i genomes.list -o output_folder -A -t (number of threads)

you run the functional annotation module.

To run this module you need to provide a list of predicted proteomes (amino acid sequence) for your interest organism. This module has the facility to functionally annotate both proteins of bacteria and archaea as well as eukaryotes.
As you noted, to run functional annotation is required that you provide how many threads will be used.

5. Outputs

GAP outputs were deposited into several folders inside the main folder that you name. In the following lines we will explore the outputs obtained when running the complete workflow.

Prodigal outputs:

For the prediction module we obtain a folder named *prodigal_outputs* that contain other three subfolders named *aa_fasta*, *nt_fasta*, *gff_files*.
aa_fasta subfolder contains amino acid sequences of predicted genes in fasta format.
nt_fasta subfolder contains nucleotide sequences of predicted genes in fasta format.
gff_files subfolder contains gff format annotation files for all genomes. To know more about the gff file format go to https://www.ensembl.org/info/website/upload/gff.html .

Functional annotation outputs:

For the functional annotation module we obtain three folders *eggNOG_outputs*, *interproSCAN_outputs* and *Final_annotation*.
eggNOG_outputs folder contains two subfolders *raw* and *parsed*. Subfolder raw contains other two subfolders *annotations* and *seed_orthologs*. The annotation subfolder contains raw functional annotations files for all genomes and seed_orthologs subfolder tab separated files with orthologs assignation outcomes. The parsed subfolder contains tab separated files that with functional annotations for all genomes, these files contain the following information:
gene_id = gene name assigned by prediction step
seed_eggNOG_ortholog = best ortholog id
seed_ortholog_evalue = E Value for alignment
seed_ortholog_score = alignment score
best_tax_level = best taxonomic level assignation
Preferred_name = putative product name
GOs = Gene Ontology terms
EC = EC number
KEGG_ko = KEGG ortholog id
KEGG_Pathway = KEGG pathway id
KEGG_Module = KEGG module id
KEGG_Reaction = KEGG reaction id
KEGG_rclass = KEGG reaction class
BRITE = KEGG brite id
KEGG_TC =
CAZy = structural and biochemical information on Carbohydrate-Active Enzymes
BiGG_Reaction = BiGG models id
Tax.scope = taxonomic level used for annotation
eggNOG.Ogs = eggNOG OGs

Best.Ogs =
COGs = Cluster Ortholog Group categories
eggNOG.HMM.Description = eggNOG free description

interproSCAN_outputs folder contains *raw* and *parsed* subfolders. The raw subfolder contains all raw data obtained from interproscan annotation. On the other hand, the parsed subfolder contains tab separated files with the following information:

gene_id = gene name assigned by prediction step
Analysis = all tests performed for that gene
AnAccess = accession numbers for all test
Description = description of all test
InterPro_ann = interpro accession
Interpro_description = intepro description
GO = GO terms
Pathways = KEGG, BiGG, Reactome or other pathway/reaction id

Finally the Final_annotation folder contains tsv files (easy to read in excel or libreoffice) that contain all information from parsed files.
In future versions of GAP we implement several new output files such GAF or GMT files. To know more about GAP or GMT files see following links:
GAF : http://geneontology.org/docs/go-annotation-file-gaf-format-2.1/
GMT : https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats

6. Contact and citation

If you have any comments about GAP we would like to hear them. You can either enter them into the github bug tracker at:

https://github.com/lalomartinez/GAP/issues

..or send them directly to jesus.martinez@mayor.cl.

This pipeline was designed by J. Eduardo Martinez-Hernandez and members of the Laboratory of Integrative Bioinformatics (Universidad de Chile) and Network Biology Laboratory (Universidad Mayor). All reproduction must be cited.

7. Reference

1. Hyatt, D. *et al.* Prodigal: prokaryotic gene recognition and translation initiation site

identification. *BMC Bioinformatics* **11**, 119 (2010).

2. Huerta-Cepas, J. *et al.* eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. *Nucleic Acids Res.* **44**, D286–93 (2016).
3. Huerta-Cepas, J. *et al.* Fast Genome-Wide Functional Annotation through Orthology Assignment by eggNOG-Mapper. *Mol. Biol. Evol.* **34**, 2115–2122 (2017).
4. Jones, P. *et al.* InterProScan 5: genome-scale protein function classification. *Bioinformatics* **30**, 1236–1240 (2014).