# Annotate your genome with PGAP

## Requirements

To run the PGAP pipeline you will need:

- Python (version 3.6 or higher),
- the ability to run Docker (see https://docs.docker.com/install/ if it is not already installed), Singularity, or Podman
- about 100GB of storage for the supplemental data and working space,
- and 2GB-4GB of memory available per CPU used by your container.
- The CPU must have SSE 4.2 support (released in 2008).

## Quick Start

Download the file using either

```
$ curl -OL https://github.com/ncbi/pgap/raw/prod/scripts/pgap.py
```

or

```
$ wget -O pgap.py https://github.com/ncbi/pgap/raw/prod/scripts/pgap.py
```

depending upon which utility your system has installed. If one does not work, try the other.

Install the pipeline. By default it will install in `$HOME/.pgap`, but this location can be changed by setting environmental variable `PGAP_INPUT_DIR`.

```
$ chmod +x pgap.py
$ ./pgap.py --update # required files are downloaded and ex
```

```
tracted
```

Run the pipeline on the Mycoplasmoides genitalium genome provided with the installation:

```
$ ./pgap.py -r -o mg37_results -g $HOME/.pgap/test_genomes/
MG37/ASM2732v1.annotation.nucleotide.1.fasta -s 'Mycoplasmo
ides genitalium'
```

Output will be located in the `mg37_results` subdirectory as specified by the `-o` flag.

## Bring Your Own Data

To run this pipeline using your own genomes, you will need, at a minimum, the multifasta file for the genome, and the associated organism name (genus or genus species).

```
$ ./pgap.py -r -o <results> -g <fasta> -s '<organism_name>'
```

After successful completion the output directory will contain the following files:

- .fasta - nucleotide FASTA file that you supplied
- ani-tax-report.txt - ANI report in text format
- ani-tax-report.xml - ANI report in XML format for machine processing
- annot-gb.ent - ASN.1 file in Seq-entry genbank format
- annot.faa - FASTA file with all proteins
- annot.fna - FASTA file with all nucleotides (note the file might be slightly normalized compared to your input nucleotide FASTA file)
- annot.gbk - annotations in flatfile format
- annot.gff - annotations in GFF3 format
- annot.sqn - ASN.1 file in Seq-submit format

- annot_cds_from_genomic.fna - FASTA file with nucleotide sequences of all coding regions
- annot_translated_cds.faa - FASTA file with translated sequences of all coding regions
- annot_with_genomic_fasta.gff - file combining annotations in GFF format and nucleotide sequence in FASTA format used in some third party applications, like Roary
- checkm.txt - CheckM output for this genome
- cwltool.log - CWL tool log that could be instrumental for post mortem analysis of failures
- fastaval.xml - XML file with validation results for input FASTA file

## Useful options

| Command | Description |
| --- | --- |
| `-g <path>, --genome <path>` | Path to genomic fasta |
| `-s 'organism', --organism 'organism'` | Genus, or genus species |
| `-r, --report-usage-true` | Report anonymized usage metadata to NCBI |
| `-n, --report-usage-false` | Do not report anonymized usage metadata to NCBI |
| `-o <path>, --output <path>` | Output directory to be created, which may include a full path |
| `--ignore-all-errors` | Ignore errors from quality control analysis, in order to obtain a draft annotation |
| `--no-internet` | Disable internet access for all programs in pipeline |
| `-D <path>, --docker <path>` | Docker-compatible executable (e.g. docker, podman, singularity), which may include a full path like /usr/bin/docker |
| `--taxcheck` | Also calculate the Average Nucleotide Identity to type assemblies |
| `--taxcheck-only` | Only calculate the Average Nucleotide Identity to type assemblies, do not run |

| Command | Description |
| --- | --- |
| | PGAP |
| `--auto-correct-tax` | Override the organism provided in the input YAML file, if the taxcheck predicts a different organism with high confidence. Use in combination with the `--taxcheck` flag |
| `-d, --debug` | Debug mode. Retain intermediate files needed for investigating failures |