Genome Annotation Pipeline

Description

Annotate one or more genome assemblies using BRAKER2. See the Snakefile for exact commands, the main steps implemented here are:

- Mask repeats in genome using RepeatMasker
- Run braker in mode proteins of any evolutionary distance
- Annotate proteins predicted by braker/augustus with Pfam domains using hmmer (TODO: Should we use InterProScan instead?)

As always: Ensure settings are appropriate to your case, read working program documentation, check reliability of results - This is just a pipeline.

Set up conda env

Assuming conda, bioconda, and optionally mamba have been already installed:

```
conda create --yes -n genomeAnnotationPipeline
conda activate genomeAnnotationPipeline
mamba install -n genomeAnnotationPipeline --yes --file requirements.txt
```

Additional software

Download the appropriate distribution of GeneMark-ES/ET/EP program. The downloaded file should be named something like `gmes_linux_64.tar.gz`).

GeneMark is free to use but its license prevents from distributing it so it has to be downloaded manually as a `tar.gz` bundle. The pipeline will take care of its installation. To check the kernel version of your system and download the appropriate distribution execute `uname -r`.

Run

```
snakemake -p -n -j 5 \
    --config sample_sheet=$PWD/sample_sheet.tsv \
             genemark_tar_gz=$PWD/gmes_linux_64.tar.gz \
    -d output
```

Input option

- `sample_sheet=`

Tab-separated file indicating the genome files to be annotated and other input options. You can annotate different genome files and/or the same genome with different settings. Column are:

Column	Description
genome_id	A unique identifier for the output sub-directory of this annotation (no spaces or metacharacters)
genome_fasta	Fasta file of the genome to be annotated
protein_database	Local fasta file or URL (e.g. from plasmodb) of the proteins to use for training. Alternatively, a taxonomy identifier (e.g. Apicomplexa, see below)
repeatmasker_species	NCBI taxonomy for option `-species` for repeat masker. E.g. *plasmodium*. Use NA if the genome is already masked

If `protein_database` is a taxonomy, the pipeline downloads OrthoDB and extracts the proteins belonging to this taxonomy. See OrthoDB for available taxonomies.

- `genemark_tar_gz=` Full path to `genemark tar.gz` download as explained above
- `-d/--directory` Output directory
- `-n` Dry-run mode - omit to actually execute the workflow
- `-j` Number of jobs to run in parallel

Output

Most relevant files probably are:

- `'{genome_id}/hmmer/augustus.hints.gff3'`: Predicted genes and gene features from braker/augustus annotated with Pfam domains
- `{genome_id}/braker/augustus.hints.aa`: Fasta file of aminoacid sequences of the mRNAs in `augustus.hints.gff3`
- `{genome_id}/braker/augustus.hints.codingseq`: Fasta file of nucleotide sequences of the mRNAs in `augustus.hints.gff3`

Misc (Ignore me)

Example of mapping proteins to genome using spaln:

```
spaln -W -KP ToxoDB-56_TgondiiRH.fasta
spaln -O:0 -Q7 -dToxoDB-56_TgondiiRH ToxoDB-56_TgondiiRH88_AnnotatedProteins.fasta \
> ToxoDB-56_TgondiiRH.splan.gff
```

For testing annotation of *P. berghei* ANKA against Apicomplexa without Pb proteins:

```
pigz -cd orthodb/odb10v1_all_fasta.tab.gz \
| ./scripts/getOrthodbProteinsForTaxonomy.py -f - \
    -l2s odb10v1_level2species.tab.gz \
    -l odb10v1_levels.tab.gz \
    -s odb10v1_species.tab.gz \
    -i 'Apicomplexa' -x 'Plasmodium berghei ANKA' > tmp/apicomplexa_wo_pbanka.fasta
```

Download *T. gondii* assemblies from Xia *et al.*, 2021, NCBI accession PRJNA638608:

```
python -c "
import urllib.request
import os

NCBI = 'https://ftp.ncbi.nlm.nih.gov/genomes/all'

assembly = {'RH88': 'GCA_019455545.1_UPITT_Tgon_RH88_1.0',
            'ME49': 'GCA_019455585.1_ASM1945558v1',
            'CTG': 'GCA_016808245.1_ASM1680824v1'}

for strain in assembly:
    id = assembly[strain]
    url = f'{NCBI}/{id[0:3]}/{id[4:7]}/{id[7:10]}/{id[10:13]}/{id}/{id}_genomic.fna.gz'
    name = strain + '_' + '_'.join(id.split('_')[0:2])
    urllib.request.urlretrieve(url, strain + '_' + os.path.basename(url))
"
```

Compile this document to pdf

```
pandoc -V colorlinks=true -V geometry:margin=1in README.md -o README.pdf
```