Reproducing Analysis of MetaGeneMark2

Karl Gemayel

May 21 2020

Contents

1 Introduction

This document serves as a step-by-step instruction manual on how to replicate results from the MetaGeneMark2 paper.

2 Downloading and installing

2.1 Code

Downloading the code is fairly straightforward using `git`. The latest version can be downloaded from `WEBSITE`. To install, simply run

```
source config.sh
source install.sh
```

The first command loads all environment variables (including paths to data directories, binaries, etc...), and the second command creates an executable from all python files and stores them in `$bin` for easy access. For non-unix users, you can find the python driver files in `$driverpython`.

2.2 External Tools

MetaGeneMark2 and MetaGeneMark2+ and their analysis rely on a handful of external tools. The following need to be installed:

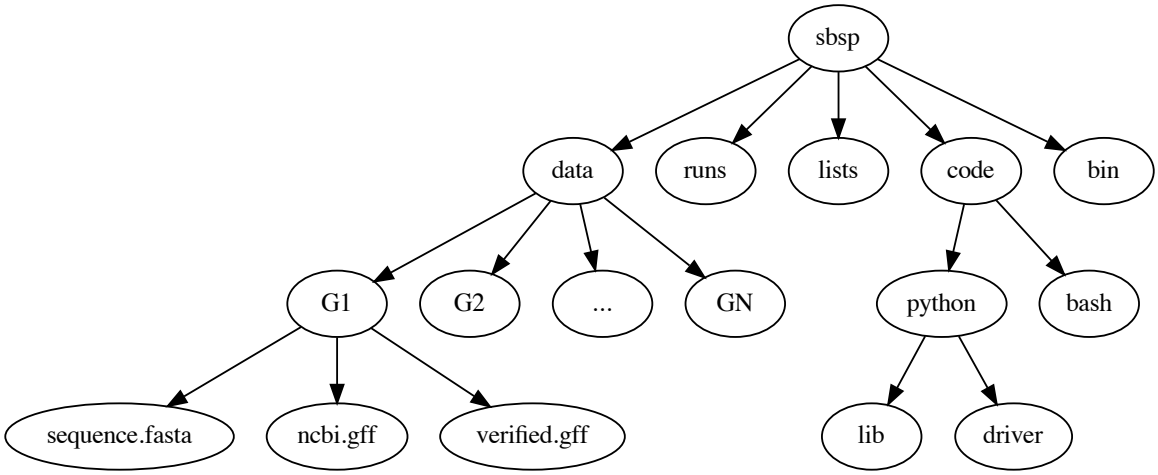
- GeneMarkS-2
  - Used for building MetaGeneMark2+ predictions, and for analysis
  - Link: `http://exon.gatech.edu/GeneMark/license_download.cgi`
- ClustalO:
  - Used for constructing multiple sequence alignments
  - Link: `http://www.clustal.org/omega/#Download`
- Prodigal:
  - Used for initial analysis of gene-start prediction status
  - Link: `https://github.com/hyattpd/Prodigal`

2.3 Data

TODO

3 Code and data structure

After installing MetaGeneMark2, you will have the following structure:

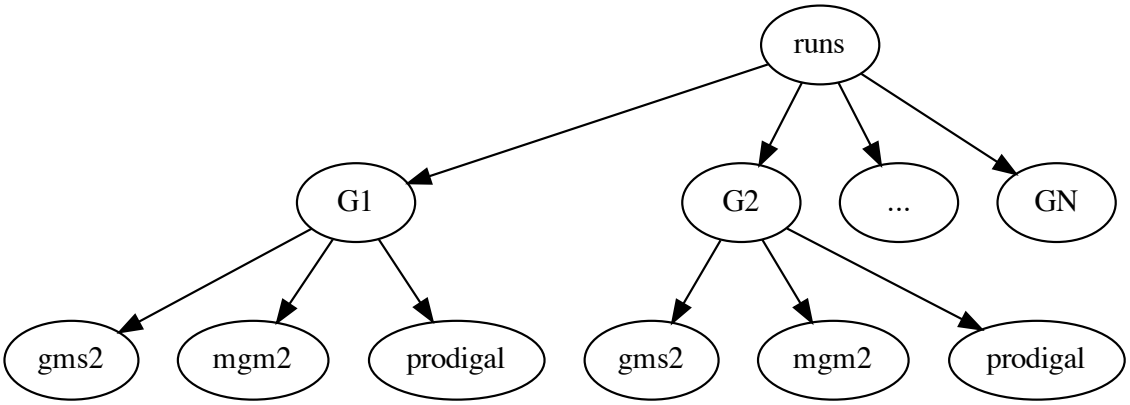


The `bin` directory contains all executables related to MetaGeneMark2, while the `bin_external` may contain external tools, such as GeneMarkS-2 or Prodigal.

The `data` directory will contain raw genome files (sequence and annotation labels) downloaded from NCBI. In particular, upon initial download of the code, it should contain the genomic sequences for the genomes with experimentally verified gene-starts.

The `list` directory has files that contain different lists of genomes (for example, those with verified genes, those selected as NCBI query genomes, etc. . .)

Finally the `runs` directory will contain runs of different tools, such as MetaGeneMark2, GeneMarkS-2, or Prodigal (as well as one for NCBI's `PGAP`). These will be placed in a subdirectory per genome, as shown below.



4 Setting up

4.1 Important: Python environment

Scripts to build and analyze results rely on a handful of python packages. The recommended way to install them is to use the `conda` package manager, and simply run

```
conda env create -f install/conda_mgm2.yaml
```

To activate this python environment, run

```
conda activate mg-starts
```

This automatically loads the correct python libraries and executables into `$PATH`.

5 Experiments

5.1 Building MGM2 start models

```
$bin/build_mgm_models_from_gms2_models_py.sh --
```

5.2 Extract NCBI Protein Homology Predictions

```
pf_gil=$lists/sbsp.list

awk -F "," '{if (NR > 1 && NF) print $1}' $pf_gil  | while read -r gcfid; do
  mkdir -p $runs/$gcfid/ncbi_ph;
  awk -F "\t" '{if ($2 == "Protein Homology") print }' $data/$gcfid/ncbi.gff >
      $runs/$gcfid/ncbi_ph/prediction.gff;
done
```

5.3 Complete Genomes

5.3.1 Verified Starts

```
# run tools on verified starts

dn_experiment=complete_verified
cd $tmp
mkdir -p $dn_experiment
cd $dn_experiment

pf_gil=$lists/verified.list
pf_mgm_mod=$bin_external/gms2/mgm_11.mod
pf_mgm2_mod=$bin_external/mgm2/mgm2_11.mod
pf_prl_options=$config/parallelization_pbs_2.conf

declare -a tools=(mgm mgm2 mprodigal fgs mga gms2 prodigal);

for t in "${tools[@]}"; do
  $bin/run_tool_on_genome_list_py.sh --pf-gil $pf_gil --type auto --tool $t --pf-mgm-mod
      $pf_mgm_mod --pf-mgm2-mod $pf_mgm2_mod --pf-parallelization-options $pf_prl_options
done


# collect statistics
pf_stats=$(pwd)/summary_complete_verified.csv
$bin/stats_per_gene_py.sh --pf-gil $pf_gil --tools ncbi_ph ncbi verified "${tools[@]}"
    --pf-parallelization-options $pf_prl_options --pf-output $pf_stats

# visualize statistics
mkdir -p figures
cd figures

$bin/viz_stats_per_gene_with_reference_py.sh --pf-data $pf_stats --reference verified
    --tools "${tools[@]}"

cd $base
```

5.3.2 StartLink+ Starts

```
# run tools on verified starts

dn_experiment=complete_startlink
cd $tmp
mkdir -p $dn_experiment
cd $dn_experiment

pf_gil=$lists/sbsp.list
pf_mgm_mod=$bin_external/gms2/mgm_11.mod
pf_mgm2_mod=$bin_external/mgm2/mgm2_11.mod
pf_prl_options=$config/parallelization_pbs_2.conf

declare -a tools=(mgm mgm2 mprodigal fgs mga gms2 prodigal);

for t in "${tools[@]}"; do
```

```
  $bin/run_tool_on_genome_list_py.sh --pf-gil $pf_gil --type auto --tool $t --pf-mgm-mod
      $pf_mgm_mod --pf-mgm2-mod $pf_mgm2_mod --pf-parallelization-options $pf_prl_options
done


# collect statistics
pf_stats=$(pwd)/summary_complete_startlink.csv
$bin/stats_per_gene_py.sh --pf-gil $pf_gil --tools sbsp ncbi ncbi_ph "${tools[@]}"
    --pf-parallelization-options $pf_prl_options --pf-output $pf_stats

# visualize statistics
mkdir -p figures
cd figures

$bin/viz_stats_per_gene_with_reference_py.sh --pf-data $pf_stats --reference sbsp ncbi_ph
    --tools "${tools[@]}"

cd $base
```

5.4 Genome Fragments

5.4.1 Verified Starts

```
dn_experiment=chunks_verified
cd $tmp
mkdir -p $dn_experiment
cd $dn_experiment

pf_gil=$lists/verified.list
pf_mgm_mod=$bin_external/gms2/mgm_11.mod
pf_mgm2_mod=$bin_external/mgm2/mgm2_11.mod
pf_prl_options=$config/parallelization_pbs_2.conf

declare -a tools=(mgm mgm2 mprodigal fgs mga gms2 prodigal);

pf_runs_summary=$(pwd)/runs_summary_chunks_verified.csv
for t in "${tools[@]}"; do
  $bin/run_tools_on_chunks_py.sh --pf-gil $pf_gil --tools "${tools[@]}" --pf-mgm2-mod
      $pf_mgm2_mod --pf-mgm-mod $pf_mgm_mod --pd-work $runs --pf-summary $pf_runs_summary
      --pf-parallelization-options $pf_prl_options
done


# collect statistics
pf_stats=$(pwd)/summary_chunks_verified.csv
$bin/stats_per_gene_on_chunk_py.sh --pf-summary $pf_runs_summary  --reference-tools
    verified ncbi ncbi_ph  --parallelization-options $pf_prl_options --pf-output $pf_stats

# visualize statistics
mkdir -p figures
cd figures
```

```
$bin/viz_stats_per_gene_with_reference_py.sh --pf-data $pf_stats --reference verified
    --tools "${tools[@]}"

cd $base
```

5.4.2 StartLink Starts

```
dn_experiment=chunks_startlink
cd $tmp
mkdir -p $dn_experiment
cd $dn_experiment

pf_gil=$lists/sbsp.list
pf_mgm_mod=$bin_external/gms2/mgm_11.mod
pf_mgm2_mod=$bin_external/mgm2/mgm2_11.mod
pf_prl_options=$config/parallelization_pbs_2.conf

declare -a tools=(mgm mgm2 mprodigal fgs mga gms2 prodigal);

pf_runs_summary=$(pwd)/runs_summary_chunks_startlink.csv
for t in "${tools[@]}"; do
  $bin/run_tools_on_chunks_py.sh --pf-gil $pf_gil --tools "${tools[@]}" --pf-mgm2-mod
      $pf_mgm2_mod --pf-mgm-mod $pf_mgm_mod --pd-work $runs --pf-summary $pf_runs_summary
      --pf-parallelization-options $pf_prl_options
done


# collect statistics
pf_stats=$(pwd)/summary_chunks_startlink.csv
$bin/stats_per_gene_on_chunk_py.sh --pf-summary $pf_runs_summary  --reference-tools ncbi
    sbsp sbsp_plus ncbi_ph  --parallelization-options $pf_prl_options --pf-output
    $pf_stats

# visualize statistics
mkdir -p figures
cd figures

$bin/viz_stats_per_gene_with_reference_py.sh --pf-data $pf_stats --reference sbsp ncbi_ph
    --tools "${tools[@]}"

cd $base
```