Reproducing SBSP

The commands used to set up, reproduce, and graph results from the SBSP paper

Karl Gemayel

Mon 27 Apr 2020 10:37:05 EDT

Contents

1 Downloading and installing

1.1 Code

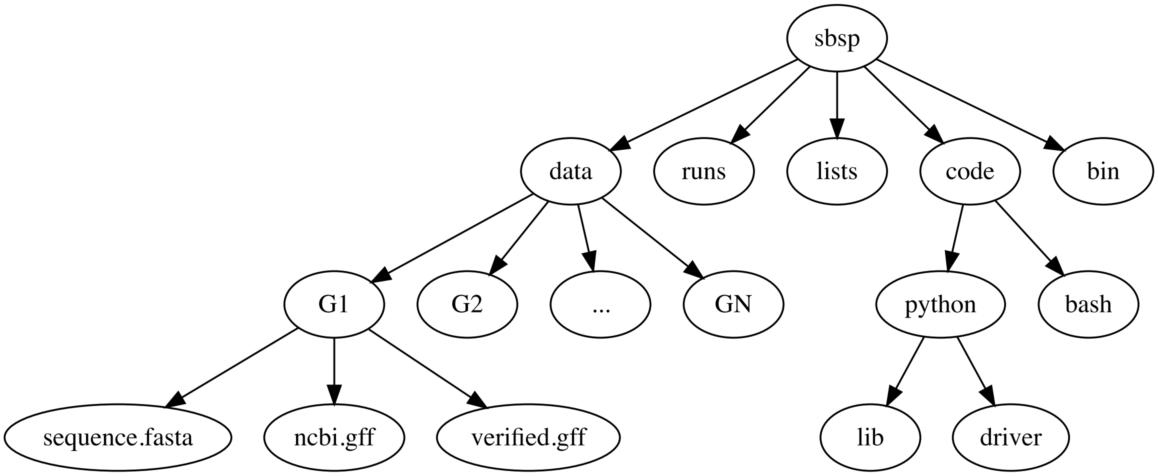
Downloading the code is fairly straightforward using `git`.

1.2 Data

We provide the databases for *Enterobacterales*, *Actinobacteria*, *Archaea*, and *FCB group*, and the sequence and label files for the genomes with verified starts: *E. coli*, *H. salinarum*, *N. pharaonis*, *M. tuberculosis*. We also provide the steps to create a data base with for any ancestor using data that can be downloaded from NCBI's website.

2 Code and data structure

After installing SBSP, you will have the following structure



2.1 Bin

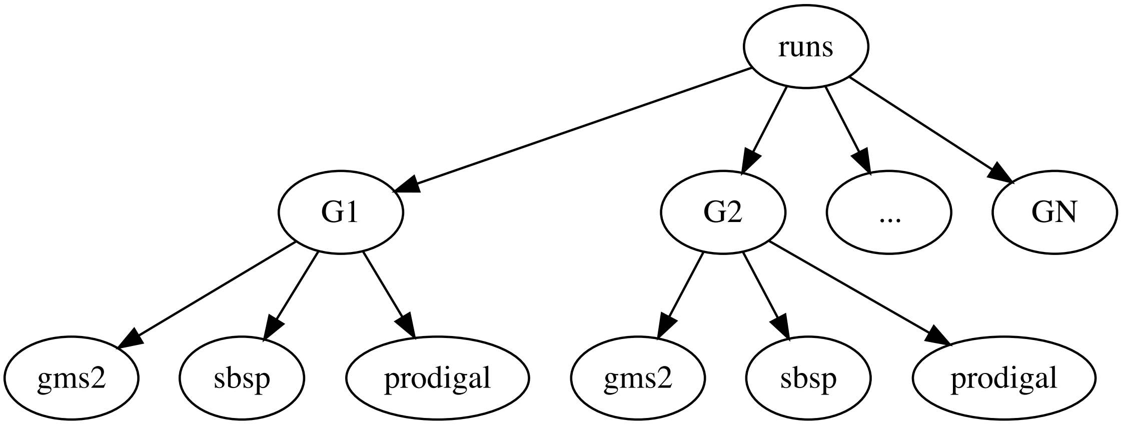
The python scripts can be located at

2.2 Data

The data directory contains all genomic raw information: mainly the sequence and labels files. If constructing databases from scratch, this directory will also include all genomes downloaded from NCBI.

2.3 Runs

For this analysis, all runs executed by SBSP, GMS2, and Prodigal will be put in subdirectories for each genome.



3 Running on verified genomes

SBSP takes as input:

- Query proteins: FASTA file

- Target protein database: Diamond database

It outputs:

- GFF file containing labels
- Multiple sequence alignment files for all queries
- details.csv: output file containing details of predictions

```
# List of genomes with verified genes
pf_list_verified=$lists/verified.list  # verified genomes
pf_db_index=$db/index.csv  # database location files
pf_sbsp_conf=$config/sbsp_defaults.conf # sbsp config file

toggle_pbs="--pf-conf-pbs $config/pbs_defaults.conf"  # if PBS not installed, set this
    option to empty: ""
sg=8   # number of genomes to run simutaneously (low number recommended)
opt_verif="--fn-q-labels verified.gff --fn-q-labels-true verified.gff"

$bin/sbsp_on_genome_list_py.sh --pf-q-list $pf_list_verified --simultaneous-genomes $sg
    --pd-work $pd_run --pf-sbsp-options $pf_sbsp_options  --pf-db-index $pf_db_index
    $opt_verif $toggle_pbs
```

4 GMS2 on metagenomes

4.1 Run GMS2 on genome fragments

```
$bin/run_tools_on_genome_fragments_py.sh --pf-genome-list $lists/verified.list --tools
    gms2 prodigal
```

5 Collecting Data

6 Tables and Graphs

6.1

7 Experiments

7.1 Difference in 5' predictions on Representative Genomes

7.1.1 Data download

```
pf_rep_bac=$lists/refseq_representative_bacteria.list
pf_rep_arc=$lists/refseq_representative_archaea.list
pf_assembly_bac=$metadata/assembly_summary.txt
$bin/download_from_ncbi_py.sh --pf-assembly-summary $pf_assembly_bac --pf-data $data
    --pf-output-list

# link ncbi as "tool" (for easy comparison wwith other tools)
cat $pf_rep_bac $pf_rep_arc | grep -v gcfid | cut -f1 -d, | while read -r line; do
  mkdir -p $runs/$line; mkdir -p $runs/$line/ncbi;
  ln -s $data/$line/ncbi.gff $runs/$line/ncbi/ncbi.gff ;
done
```

7.1.2 Run GMS2 and Prodigal

```
# Run on GMS2
$bin/run_tool_on_genome_list_py.sh --tool gms2 --pf-genome-list $pf_rep_bac --type
    bacteria --dn-run gms2
$bin/run_tool_on_genome_list_py.sh --tool gms2 --pf-genome-list $pf_rep_arc --type
    archaea --dn-run gms2

# Run on Prodigal
$bin/run_tool_on_genome_list_py.sh --tool prodigal --pf-genome-list $pf_rep_bac --type
    bacteria --dn-run prodigal
$bin/run_tool_on_genome_list_py.sh --tool prodigal --pf-genome-list $pf_rep_arc --type
    archaea --dn-run prodigal
```

7.1.3 Collect statistics

We can now collect the statistics and create the figures to compare GMS2, Prodigal, and NCBI.

```
$bin/compare_tools_5prime_py.sh --pf-genome-lists $pf_rep_bac $pf_rep_arc --list-names
    Bacteria Archaea --dn-tools gms2 prodigal ncbi --tool-names GMS2 Prodigal NCBI
```

-
- Prodigal vs NCBI
- GMS2 vs Prodigal