Cov cross-validation benchmark how-to

Robert C. Edgar

robert@drive5.com

Root directory:

```
s3://serratus-public/rce/covx/
```

Directory	Description
`scripts/`	bash scripts for running things
`sim/`	Simulated reads of query genomes
`q/`	Query genomes, one genome per identity
`r/`	Reference pan-genomes, one per identity
`bench/`	tsv files with results
`bt2/`	bowtie2 indexes for reference pan-genomes
`bwa_index/`	bwa indexes for reference pan-genomes

There is at least one relevant python script in

```
s3://serratus-public/rce/py
```

Simulated reads

Reads are in `covx/sim`. FASTQ filenames look like this:

```
-rwxrwxrwx 1 bob bob 237K Apr 29 18:35 L100.id80.1.fq
-rwxrwxrwx 1 bob bob 237K Apr 29 18:35 L100.id80.2.fq
-rwxrwxrwx 1 bob bob 473K Apr 29 18:35 L100.id80.u.fq
```

100=read length (100 or 150)

80=%identity of query with most similar genome in the reference (80,81...99)

1,2=paired reads

u=unpaired reads (1 and 2 combined).

Testing a mapper

A mapper should be run on each (read length, identity) pair.

Existing scripts to run mappers include these found in `scripts/`:

```
bowtie2, bowtie2u, bwa, bwau, bwak14, bwak14u
```

Use these as-is, or as a starting point for writing your own mapper script.

The `scripts/forsim` script is handy for doing cycling over all the (length, identity) pairs, it takes one argument which could for example be the name of a script to run a mapper, and runs it with all combinations of read length and identity. For example,

```
cd covx/scripts
./forsim ./bowtie2
```

will run:

```
./bowtie2 100 80
./bowtie2 100 81
...
./bowtie2 150 99
```

The `forsim` script can also be useful for looping over output files to collect benchmark metrics.

Measuring sensitivity

To calculate sensitivity, count the number of SAM records which are mapped. The `sam_count_mapped.py` script in `s3://serratus-public/rce/py` does this. Command-line arguments are SAM filenames, it outputs the number of mapped reads and percentage of mapped reads for each filename, e.g.:

```
cd ../sam/bowtie2
python3 sam_mapped_count.py \
  L100.id80.sam \
  L100.id81.sam
L100.id80.sam   672     33.6
L100.id81.sam   615     30.8
```

The `bench1` script runs `sam_count_mapped.py` for one mapper and stores a summary in a `bench/` file. For example,

```
cd covx/scripts
./bench1 bowtie2
```

The `bench_table` script re-formats the files in `bench/` into a convenient tsv file in `covx/results`. Note that the list of methods is hard-coded into `bench_table` so you may need to edit this script.

Measuring false-positive rate

Measuring false-positives is not currently implemented in `rce/covx/`, but it is a straightforward extension. Pick a Cov-negative dataset, e.g. SRR10853354 and run the mapper on those reads. FP rate = fraction of reads mapped to a pan-genome reference.