Cov assembler benchmark notes

Robert C. Edgar
robert@drive5.com

See also:

notebook/200430_covx_benchmark_howto.pdf
`s3://serratus-public/rce/covx/`

Reference-based assembly

The sequence identity of a novel virus with its closest known reference could be anywhere from ~100% to 80% or perhaps even less.

Reference model with simulated reads

The CovX benchmark was developed to test mappers, but it can also be used to test reference-based assembly. CovX has 20 test-reference genome pairs with identities 80, 81 ... 99, 99%. Complete genomes are provided for each pair. At each identity, reads of length 100 and 150nt are simulated. Sequencing error is not simulated, but this is ok because reads of the test genome have biological differences with the reference which pose a similar challenge to the assembler.

At a given identity, reads should be assembled against the reference genome at that identity. The accession of the reference can be found in `s3://serratus-public/rce/covx/split`, e.g. for 80% identity:

`covx/split/80.txt`

The first two lines of the file give the test genome accession ('S') and the reference genome accession ('T'):

```
head -n2 covx/split/80.txt
S       LC469308.1      1.0000
T       MG987421.1      0.1960
```

The 'S' and 'T' stand for Source and Target in my personal terminology, please note that T is the reference, not the test genome in the usual terminology of cross-validation. The floating-point number after the accession is the distance (0=identical, 1=zero similarity) from S measured from `blastn` alignments. In this example, distance 0.196 = (1 - 0.804) = 80.4% identity.

Testing low read depth

There are 1,000 read pairs = 2,000 reads in each set. For a genome of length 30,000, the read depth for reads of length L is 2,000 × L / 30,000 which is depth=7 for L=100 and depth=10 for L=150. To test more challenging cases where read depth is lower, take a subset of the reads. You can do this using the `fastx_subsample` command in usearch:

https://drive5.com/usearch/manual/cmd_fastx_subsample.html

This test may not be necessary; I guess we'll have to see if we find any datasets with smaller numbers of virus reads.

Reference model with real data

Real data will be noisier than the simulated reads for two main reasons: 1. sequencing error, which is not modeled in the CovX benchmark, and 2. reads of the host transcriptome, which will be a large fraction of the data.

To test reference-based assemblers on real data, reads of samples with SARS-Cov-2 can be assembled using a known Cov reference genome with <100% identity. The table below lists suitable genomes with their identity to SARS-Cov-2 (MT121215.1).

Acc	%Id	Description
KY417150.1	80.0	Bat SARS-like coronavirus Rs4874
AY395002.1	80.2	SARS coronavirus LC5
MG772933.1	88.2	Bat SARS-like coronavirus bat-SL-CoVZC45
MN996532.1	96.1	Bat coronavirus RaTG13

Reference genomes for assembling real Cov-2 data

It would be nice to have test cases with more identities, but few close relatives of SARS-Cov-2 are currently known. Hopefully, we will discover more of them.

Datasets containing SARS viruses are given in the table below.

Acc	Description
SRR11454606	hCov-19 infected patients Throat swab
SRR11454607	hCov-19 infected patients Faeces
SRR1942929	SARS infection at 0 hour

Datasets for testing assembly

De-novo assembly

De novo assemblers can be tested on the real datasets above.

Host reads

With reference-based assembly, host reads will not align to the reference, except perhaps in pathological cases which must be identified and handled as special cases. Otherwise, host reads should not be a problem with reference-based assembly; testing will confirm if this is correct.

With de-novo assembly, there are at two primary strategies for dealing with host reads:

(A) Remove host reads as pre-processing step before assembly.

(B) Assemble all reads and remove host contigs as a post-processing step.

A mixed strategy could be optimal:

1. Mask or filter out pathological regions from the host as a pre-processing step.
2. Assemble.
3. Remove host contigs.

Removing host contigs as a post-processing step may be more accurate, because longer sequences will align better to the host genome, which may not be easy with short reads because the host genome is not spliced and the closest available host reference genome may be quite diverged from the actual host. Alternatively, virus contigs may be identifiable as sequences which do not align to the host, in which case we don't need to explicitly identify the host contigs.

Keep in mind that different strategies may work in different situations. E.g., discarding human contigs by aligning to a reference may be easier than discarding bat contigs because humans have very similar genomes while bats are very diverse and we usually won't have a close reference genome -- there are 1,300+ species of bats!

Measuring assembly accuracy and quality

See this article for introduction to assembly quality metrics such as N50 and N90 and links to tools which report them:

https://bioinformaticshome.com/bioinformatics_tutorials/sequence-assembly/assembly-quality-metrics.html

I have no experience with virus assembly, but I would guess that because the genomes are so short, we will typically get a complete genome or a very small number of contigs, say two or three. If so, the best summary of the assembly is the length and identity of each contig to the closest known Cov reference. A reference-based assembler will probably

report these metrics. With a de-novo assembler, we could use the `dnadiff` command in mummer4 or `blastn` to generate the report.