Benchmark

We benchmark the result against several state-of-art assembly tools, both de novo and reference based methods, as follows

assemblers

1. Megahit
2. SPAdes - Careful
3. SAVAGE

reference-free (de novo)

1. virus-VG + SAVAGE (native)
2. VG-flow + SAVAGE (native)
3. VG-flow + SPAdes (secondary)
4. vsAware + SPAdes (native)
5. vsAware + Megahit (secondary)
6. ViQUF

reference-based

1. PredictHaplo
2. ShoRAH

Datasets

	simulated dataset(20000x)	vsAware+SPAdes	VG-Flow+SPAdes	VG-Flow+SAVAGE
5	HIV	work	work	work
6	POLIO	work	bug (vg-flow side)	work
10	HCV	work	bug (vg-flow side)	work
15	ZIKV	work	work	work

	real dataset	vsAware+SPAdes	VG-Flow+SPAdes	VG-Flow+SAVAGE
2	SARS-COV2 (400x, pairend), BA.2, Delta	work	bug (vg toolkit side)	fail (savage incompactible)
2	SARS-COV2 (5000x, pairend) BA.1, B.1.1	work	bug (vg toolkit side)	fail (savage incompactible)
5	HIV	work	work	work

Command lines

For simulated datasets, all experiments are ran under default settings.

For SARS-COV2 dataset (accession number: SRP359650), we estimate the sequencing depth for each sample, then paired the samples into different dataset with even coverage. We use the `fastp` as a preprocessing tool to clean the reads. For each sample, we run SPAdes (under `--careful` option) and select the longest contig as the candidate strain from SPAdes output, we use `samtools` as the alignment tools to further filter out the reads that does not map to the candidate strain or with low quality, we mix the processed reads within each dataset and run SPAdes (under `--careful` option) to generate the assembly graph and contigs required by vsAware, we then use vsAware to produce the full-length strain.

For HIV lab mix dataset (accession number: SRR961514), we use `fastp` to preprocess the reads and clean the adapters, then use assembly tools to complete the experiment.

We provide the following commands used for benchmarking and reproduce the experimental results, all tools were run under default setting, unless specified otherwise.

Tools version

fastp

fastp 0.23.2

bwa

Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.17-r1198-dirty
Contact: Heng Li hli@ds.dfci.harvard.edu

samtools

Program: samtools (Tools for alignments in the SAM format)
Version: 1.14-28-g84dfab2 (using htslib 1.14)

SPAdes

SPAdes genome assembler v3.15.4

SAVAGE (embedded in HaploConduct since 2019)

Program: HaploConduct
Version: 0.2
Release date: December 23, 2019
GitHub: https://github.com/HaploConduct/HaploConduct

Program: SAVAGE - Strain Aware VirAl GEnome assembly Version: 0.4.2 Release date: December 23, 2019
Contact: Jasmijn Baaijens

MEGAHIT

megahit: MEGAHIT v1.2.9 contact: Dinghua Li voutcn@gmail.com

virus-VG

still testing

vg-flow

last commit caeafa07e4a8c2de06d12accac6c7d590c93f722
Author: jbaaijens jasmijn_baaijens@hotmail.com
Date: Sat May 15 13:49:52 2021 -0400

ViQUF

still testing

ShoRAH

still testing

PredictHaplo

still testing

vg

vg version v1.40.0 "Suardi" Compiled with g++ (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0 on Linux Linked against libstd++ 20210601 Built by stephen@lubuntu

Gurobi

Gurobi 9.5.0

minimap2

2.23-r1111

QUAST

QUAST v5.1.0rc1 (MetaQUAST mode)

Read trimming, adapter removal

1. For SARS-COV2 (accession number: SRP359650): `fastp -i 1.fastq -o 1.fp.fastq -I 2.fastq -O 2.fp.fastq -e 36`
2. For HIV (accession number: SRR961514) `fastp -i 1.fastq -o 1.fp.fastq -I 2.fastq -O 2.fp.fastq -e 36`

Read alignment, preprocess

```
# generate sam file
bwa index contigs.fasta
bwa mem contigs.fasta 1.fq 2.fq > <prefix>.sam
```

```
# generate sorted bam file
samtools view -S -b <prefix>.sam > <prefix>.bam
samtools sort <prefix>.bam > <prefix>.sorted.bam
samtools index <prefix>.sorted.bam

# filter reads
samtools view -f 0x2 <prefix>.sorted.bam -o <prefix>.sorted.fil.bam -h
<best_contig_name>
samtools sort -n <prefix>.sorted.fil.bam -o <prefix>.2sorted.fil.bam

# convert back to pair-end fastq format
samtools fastq -@ 8 <prefix>.2sorted.fil.bam -1
<output_name>.forward.fastq -2 <output_name>.reverse.fastq -N
```

the preprocess section also embedded in the python script `filt_fastq.py`

```
python filt_fastq.py -1 1.fp.fastq -2 2.fp.fastq -r contigs.fasta -o
<output_name> -n <best_contig_name>
```

De novo sssembly tools

```
SAVAGE: haploconduct savage -p1 forward.fastq -p2 reverse.fastq --revcomp
--split 30 -t 8

SPAdes: spades.py -1 forward.fastq -2 reverse.fastq --careful -t 8 -o
<output_dir>

MEGAHIT: megahit -1 forward.fastq -2 reverse.fastq --k-list
21,33,55,77,99,127 --bubble-level 0 --keep-tmp-files -t 8 -o <output_dir>

Virus-VG:
* python build_graph_msga.py -f forward.fastq -r reverse.fastq \
               -c contigs_stage_c.fasta -t 8 -vg vg-v1.7.0
* python optimize_strains.py -m 100 -c 200 node_abundance.txt \
contig_graph.final.gfa

VG-flow:
* python build_graph_msga.py -f forward.fastq -r reverse.fastq \
               -c contigs_stage_c.fasta -t 8 -vg vg

* python vg-flow.py -m 100 -c 200 --greedy_mode=all \
        node_abundance.txt contig_graph.final.gfa

vsAware: python vsAware.py -a spades -g
assembly_graph_after_simplification.gfa -p contigs.paths -o <output_dir>

ViQUF:
still testing, k=121 as mentioned by ViQUF paper
```

Reference-guided quasispecies reconstruction tools

```
ShoRAH: python shorah.py -b paired.sorted.bam -f reference.fasta

PredictHaplo: PredictHaplo-Paired config.txt
```

Assembly evaluation

```
python2 metaquast.py --unique-mapping -m 250 -t 8 <f1.fasta> <f2.fasta>
... -o <output_dir> -R <ref1.fasta>,<ref2.fasta>,...,<refn.fasta>"
```

the evaluation section also embedded in a python script `quast_evaluation.py`

```
python eval_script/quast_evaluation.py -quast <path_to_MetaQUAST> -cs
<file1.fasta> <file2.fasta> ... -ref <refs.fasta> -o <output_dir>
```

Reference

1. virus-VG: J.A. Baaijens, B. van der Roest, J. K¨oster, L. Stougie, and A. Sch¨onhuth. Full-length de novo viral quasispecies assembly through variation graph construction. Bioinformatics, to appear, 2019.
2. VG-flow: Baaijens, Jasmijn A., Leen Stougie, and Alexander Schönhuth. "Strain-aware assembly of genomes from mixed samples using flow variation graphs." International Conference on Research in Computational Molecular Biology. Springer, Cham, 2020.
3. SAVAGE: J.A. Baaijens, A. Zine El Aabidine, E. Rivals, and A Scho¨nhuth. De novo assembly of viral quasispecies using overlap graphs. Genome Research, 27(5):835–848, 2017.
4. SPAdes: A. Bankevich, S. Nurk, D. Antipov, A.A. Gurevich, M. Dvorkin, A.S. Kulikov, V.M. Lesin, S.I. Nikolenko, S. Pham, A.D. Prijbelski, A.V. Pyshkin, A.V. Sirotkin, N. Vyahni, G. Tesler, P.A. Pevzner, and M.A. Alekseyev. SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. Journal of Computational Biology, 19(5):455–477, 2012.
5. ViQUF: Freire, Borja, et al. "ViQUF: de novo Viral Quasispecies reconstruction using Unitig-based Flow networks." IEEE/ACM Transactions on Computational Biology and Bioinformatics (2022).
6. predictHaplo: S. Prabhakaran, M. Rey, O. Zagordi, N. Beerenwinkel, and V. Roth. HIV haplotype inference using a propagating dirichlet process mixture model. IEEE Transactions on Computational Biology and Bioinformatics, 11(1):182–191, 2014.
7. ShoRAH: O. Zagordi, A. Bhattacharya, N. Eriksson, and N. Beerenwinkel. ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. BMC Bioinformatics, 12(1):119, 2011.
8. Megahit: Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn

graph. Bioinformatics, 31(10):1674–1676, 01 2015.