ZGA: a flexible pipeline for read processing, *de novo* assembly and annotation of prokaryotic genomes

Korzhenkov A.A., oscypek@ya.ru

NRC Kurchatov Institute, Moscow, Russia

Abstract

Whole genome sequencing (WGS) became a routine method in modern days and may be applied to study a wide spectrum of scientific problems. Despite increasing availability of genome sequencing by itself, genome assembly and annotation could be a challenge for an inexperienced researcher. To solve this problem, a bioinformatic pipeline was developed to conduct a user from raw sequencing reads to annotated bacterial or archaeal genome ready for deposition to any INSDC database as NCBI, ENA or DDBJ. The pipeline is fully automated and doesn't require internet connection after installation which prevents data leakage and premature publication of genome sequences. The source code of the pipeline is freely available at https://github.com/laxeye/zga/. The software may be installed from popular repositories: Anaconda Cloud (https://anaconda.org/bioconda/zga/) and PyPI (https://pypi.org/project/zga/).

Keywords

Microbial genomics, genome assembly, genome annotation, next generation sequencing

Introduction

26 years have passed since the Haemophilus influenzae genome was sequenced in 1995 (Fleischmann, 1995) and high-throughput sequencing became trivial. To date (April 2021) NCBI RefSeq includes more than 200 000 genome assemblies of Bacteria and more than 1000 of Archaea. Genome Taxonomy Database (Release 06-RS202, 27th April 2021), which includes genomes of both cultivated and uncultivated prokaryotes, consists of 254 090 bacterial genomes representing 45 555 species of 12 037 genera and 4 316 archaeal genomes representing 2 339 species of 851 genera (Parks et al., 2020).

These numbers seem to be high enough, but in comparison with global prokaryotic diversity estimated to one million OTUs (Louca et al., 2019) or 200 000 bacterial genera and 5 000 that of Archaea (Amann and Rosselló-Móra, 2016) they seem to be not so impressive. The scientific community should be prepared for ten-fold growth genomic data provided only from representatives of novel genera and species.

To date several instruments were developed for prokaryotic genome assembly and annotation (Supplementary Table S1), but most of them are abandoned. Old versions of assemblers integrated in the pipelines cannot handle data from modern sequencers (Nanopore, PacBio, BGI). Growth of existing databases, emergence of novel databases, development and modernisation of basic bioinformatic tools force use of updating annotation instruments. From pipelines specialized on full cycle processing of prokaryotic genomic data only **tormes** (Quijada et al., 2019) and **nullarbor** (Seeman et al., 2020) are still in active development. Both tormes and nullarbor use only Illumina sequencing data which prevents their application to hybrid and long read-based genome assembling.

Methods

Successful genome assembly and annotation requires several steps: read quality control, read pre-processing, genome assembling, genome quality assessment, genome annotation (Toshchakov et al., 2017). There are many tools which perform above mentioned operations so following criteria were used for the selection:

- Speed and multi-threading
- Accuracy and reproducibility
- Free use, availability of source code and permissive licensing
- Moderate system requirements not exceeding modern consumer PC
- Simplicity of integration
- Continuation of development
- Availability in software repositories such as Anaconda Cloud, Python Packaging Index (PyPI), brew etc.

Input data

On 21 of April 2021 NCBI SRA database included 1,424,935 bacterial genome sequencing experiments conducted on different high-throughput platforms: Illumina (1,385,704), Pacific Biosciences SMRT (16,989), Oxford Nanopore (8,064), 454 Life Sciences (7,513), Ion Torrent (4,517), BGISEQ (1106), ABI SOLiD (724). This data shows that the developing pipeline should support popular modern platforms such as Illumina, PacBio and Oxford Nanopore as well as emerging and prospective BGISEQ technology.

Despite different technologies, sequencing data from Illumina and BGISEQ platforms are very similar in terms of read length, base quality distributions and overall sequencing quality (Kim, H.-M. et al., 2021). These two platforms use different library-prep adapters which should be trimmed during read processing. This issue was solved by addition of BGISEQ-specific adapter sequences to Illumina-oriented read processing workflow.

Read quality control

Read quality control (QC) performed with **fastp** (Chen et al., 2018) which supports both short and long sequencing reads and writes a report in HTML (user-friendly) and JSON (software-friendly) formats.

Read processing

Short read processing includes quality trimming, adapter trimming, filtering by length and complexity, merging overlapped reads. **BBTools** package (Bushnell, 2017) includes specialized tools for above mentioned actions: **BBDuk** for quality trimming, adapter removal and filtering by length, **BBMerge** for overlapping paired end reads merging and **filterbytile.sh** script for optional filtering of Illumina reads. Short reads may be corrected with **Tadpole** from the same package. It may be useful to speed-up the assembly process, SPAdes read-correction may be skipped in this way or even if it will be performed time and memory consumption decrease. BBtols itself contains a script for read processing and genome assembly named “assemblyPipeline.sh”. Despite it’s dignity the script should be manually edited if any change in the workflow is desired and it’s compatible only with Illumina reads preventing its usage in the universal pipeline.

Illumina mate-pair reads help in assembling larger scaffolds or even circular genomes. Mate-pair reads are processed with **NxTrim** (O'Connell et al., 2015) to filter out suspecting reads and trim adapter sequences. Long reads from ONT or PacBio platforms currently aren't processed due to error correction stage in Flye assembler.

Genome size estimation may be useful for QC of assembled genome and is essential for genome assembling with Flye (Kolmogorov et al., 2019) and may be useful in QC. Genome size estimation in the pipeline is performing with **Mash** (Ondov et al., 2016).

Assembly

Three genome assemblers covering all possible variants of input files: short (Illumina, BGI) reads, long (PacBio, Nanopore) reads or their combination were selected: SPAdes, Unicycler and Flye.

SPAdes is the short-read genome assembler, which is able to use long reads for genome scaffolding (Bankevich et al., 2012). **Unicycler** assembler was intended for hybrid genome assembly, it uses SPAdes for short-read based assembly and **miniasm** for long-read based assembly (Wick et al., 2017). Unicycler performs extensive bridging that generally results in better assemblies than just using SPAdes at the cost of time. **Flye** assembler (Kolmogorov et al., 2019) works only with long reads, but resulting assembly may be then polished using short reads in case they are provided. Flye needs to be provided with estimated genome size that may be specified by user or calculated with Mash during the previous step of the pipeline.

Assembly polishing

Assembly polishing is a process of correcting bases, fixing mis-assemblies and filling gaps to improve quality of resulting genome assembly using sequencing reads. SPAdes includes MismatchCorrector, a module improving mismatch and short indel rates in the assembly which uses the **BWA** (Li H. and Durbin R., 2009) and requires no additional polishing. Unicycler uses **Pilon** (Walker et al., 2014) for assembly improvement and also doesn't require external polishing. Flye assembler uses an internal tool to polish assembly with long reads. For a hybrid assembly with Flye there is a possibility to skip the internal stage of genome polishing with long reads and perform polishing with short reads using Racon (Vaser et al., 2017) which improves quality of the genome assembly.

Assembly quality control

PhiX is a sequencing control frequently added during Illumina sequencing and the study of Integrated Microbial Genomes database showed that more than 5% of genomes had PhiX contamination (Mukherjee et al., 2015). To detect PhiX contamination, a reference nucleotide sequence was aligned against genome assembly with BLASTn from NCBI BLAST+ (Camacho et al., 2009). Contigs which have alignment coverage of PhiX sequence more than 50% being removed from assembly using **Biopython** (Cock et al., 2009).

CheckM (Parks et al., 2015) is used to assess the quality of genome assembly was selected in terms of genome completeness, contamination and heterogeneity. Analysis is possible in taxonomy mode for which rank and taxon name should be provided by the user, by default a bacterial marker set is used. Another option is the lineage mode for which the marker set will be defined by CheckM itself.

Annotation

To date several pipelines for prokaryotic genomes annotation are developed, but only two autonomous offline open-source tools are in active development: **Prokka** (Seeman, 2014) written in Perl and **DFAST** (Tanizawa et al., 2018) written in Python and supporting modern versions of the interpreter (ver. 3.6 and higher). To reduce dependencies count, decrease language heterogeneity and accounting to announce of Perl7 (https://www.perl.com/article/announcing-perl-7/) which may lead to broken dependencies of many Perl modules, including such used by Prokka, DFAST was implemented in ZGA pipeline.

Logging

For user convenience logging during the run of the pipeline was implemented. Information, warnings and errors are displayed on the screen and are written to the log file in the output directory, while debug information is only written to the log file to not overwhelm the user.

Output data

After successful completion of the pipeline users obtain an annotated genome ready for submission to INSDC databases. Output directory also includes read quality statistics, processed reads, raw genome assembly and genome completeness and contamination report.

Genome assembly quality assessment

QUAST (Gurevich et al., 2013) was used to collect genome assembly metrics from available reference genomes and genome assemblies resulting from the pipeline.

Computational resources

The pipeline was tested on several devices including laptop, PC and workstation. To emphasize the accessibility of the pipeline and it's modest requirements I present results of pipeline run on a consumer laptop, specifications given in Supplementary Table S2. During the runs the memory limit was set to 8 gigabytes to be close to modest volume of RAM in some laptops. The software versions are shown in Supplementary table S3.

Results

ZGA pipeline was written in Python and is compatible with interpreter versions from 3.6 and newer. The pipeline starts from long or short (paired or unpaired) sequencing reads and results in an annotated genome assembly. The pipeline may be stopped after any step to skip annotation, genome QC or even genome assembly if only read-processing is desired. Currently processing of long reads is not implemented due to an internal correction algorithm of Flye. The pipeline workflow is shown on Figure 1.

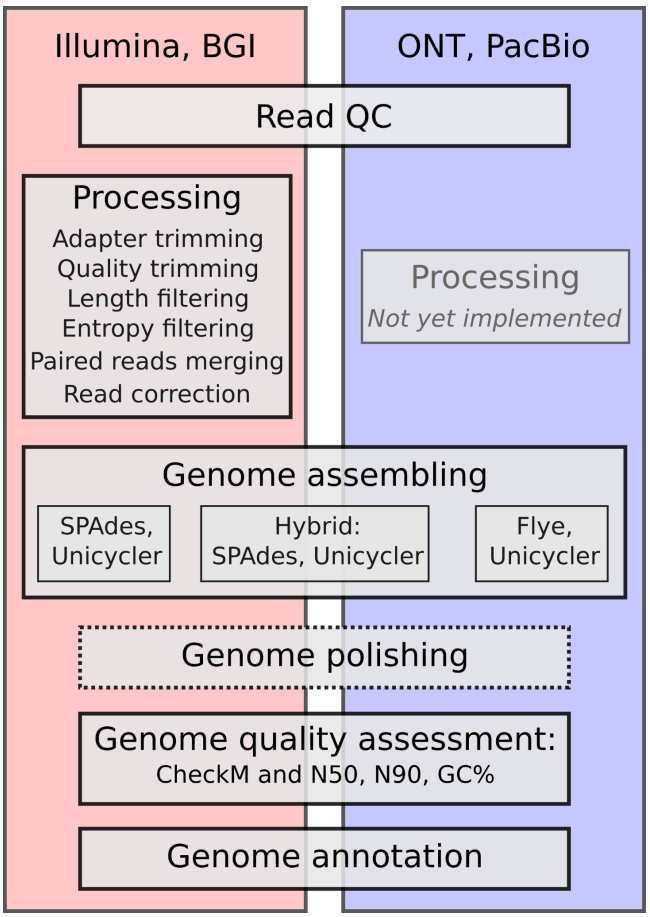

Figure 1. ZGA pipeline workflow representation

To evaluate performance of ZGA pipeline several genomic datasets from NCBI database were selected for which both genome assembly (NCBI GenBank) and raw reads (NCBI SRA) were available. Genomes with high and low GC-content were selected to show absence of limitations of the pipeline. All datasets were successfully processed in a reasonable time, depending on a sequencing data volume. Results of the pipeline on public data from the NCBI database are shown in Supplementary Table S4. All short-read assemblies produced by ZGA show better or at least same assembly quantitative metrics in comparison with assemblies deposited in NCBI.

Conclusion

ZGA pipeline successfully performs key steps of prokaryotic genome assembly and annotation for data from different sequencing platforms. In less than one hour researchers obtain an annotated genome assembly ready for further analysis or deposition in public databases. The source code structure allows tools replacement or addition to keep the pipeline using state-of-the-art tools and methods.

Acknowledgements

This work was supported by the Ministry of Science and Higher Education of Russian Federation allocated to the Kurchatov Center for Genome Research (grant No 075-15-2019-1659).

The author thanks members of the Bioconda team for helping with recipe updating.

References

Amann R, Rosselló-Móra R. (2016). After all, only millions? mBio 7(4):e00999-16. https://doi.org/10.1128/mBio.00999-16

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., ... & Pyshkin, A. V. (2012). SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. Journal of computational biology, 19(5), 455-477. https://doi.org/10.1089/cmb.2012.0021

Bushnell, B., Rood, J., & Singer, E. (2017). BBMerge–accurate paired shotgun read merging via overlap. PloS one, 12(10). https://doi.org/10.1371/journal.pone.0185056

Camacho, C., Coulouris, G., Avagyan, V. et al. (2009). BLAST+: architecture and applications. BMC Bioinformatics 10, 421. https://doi.org/10.1186/1471-2105-10-421

Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics, 34(17), i884-i890. https://doi.org/10.1093/bioinformatics/bty560

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., ... & De Hoon, M. J. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25(11), 1422-1423. https://doi.org/10.1093/bioinformatics/btp163

Fleischmann RD, Adams MD, White O, et al. (1995) Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. Science, 269(5223):496-512. https://doi.org/10.1126/science.7542800

Gurevich A., Saveliev V., Vyahhi N., Tesler G. (2013). QUAST: quality assessment tool for genome assemblies, Bioinformatics, 29(8), 1072–1075. https://doi.org/10.1093/bioinformatics/btt086

Kim, H.-M. et al. (2021) Comparative analysis of 7 short-read sequencing platforms using the Korean Reference Genome: MGI and Illumina sequencing benchmark for whole-genome sequencing. Gigascience, 10. https://doi.org/10.1093/gigascience/giab014

Kolmogorov, M., Yuan, J., Lin, Y., & Pevzner, P. A. (2019). Assembly of long, error-prone reads using repeat graphs. Nature biotechnology, 37(5), 540-546. https://doi.org/10.1038/s41587-019-0072-8

Louca S, Mazel F, Doebeli M, Parfrey LW (2019) A census-based estimate of Earth's bacterial and archaeal diversity. PLoS Biol 17(2): e3000106. https://doi.org/10.1371/journal.pbio.3000106

Mukherjee, S., Huntemann, M., Ivanova, N. *et al.* Large-scale contamination of microbial isolate genomes by Illumina PhiX control. *Stand in Genomic Sci* **10,** 18 (2015). https://doi.org/10.1186/1944-3277-10-18

O'Connell, J., et al. (2015) NxTrim: optimized trimming of Illumina mate pair reads. Bioinformatics 31(12), 2035-2037. https://doi.org/10.1093/bioinformatics/btv057

Ondov, B.D., Treangen, T.J., Melsted, P. et al. (2016) Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol, 17, 132. https://doi.org/10.1186/s13059-016-0997-x

Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P., & Tyson, G. W. (2015). CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome research, 25(7), 1043-1055. https://doi.org/10.1101/gr.186072.114

Parks, D.H., Chuvochina, M., Chaumeil, P. et al. A complete domain-to-species taxonomy for Bacteria and Archaea. Nat Biotechnol (2020). https://doi.org/10.1038/s41587-020-0501-8

Quijada, N. M., Rodríguez-Lázaro, D., Eiros, J. M., & Hernández, M. (2019). TORMES: an automated pipeline for whole bacterial genome analysis. *Bioinformatics*, *35*(21), 4207-4212. https://doi.org/10.1093/bioinformatics/btz220

Seemann T. (2014). Prokka: rapid prokaryotic genome annotation. Bioinformatics. 30(14):2068-9. https://doi.org/10.1093/bioinformatics/btu153

Seemann T, Goncalves da Silva A, Bulach DM, Schultz MB, Kwong JC, Howden BP. Nullarbor Github https://github.com/tseemann/nullarbor

Tanizawa, Y., Fujisawa, T., & Nakamura, Y. (2018). DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication. Bioinformatics, 34(6), 1037-1039. https://doi.org/10.1093/bioinformatics/btx713

Toshchakov, S. V., Kublanov, I. V., Messina, E., Yakimov, M. M. & Golyshin, P. N. Genomic Analysis of Pure Cultures and Communities. in Hydrocarbon and Lipid Microbiology Protocols: Genetic, Genomic and System Analyses of Communities (eds. McGenity, T. J., Timmis, K. N. & Nogales, B.) 5–27 (Springer, 2017). https://do.org/10.1007/8623_2015_126

Vaser, R., Sović, I., Nagarajan, N., & Šikić, M. (2017). Fast and accurate de novo genome assembly from long uncorrected reads. Genome research, 27(5), 737-746. https://doi.org/10.1101/gr.214270.116

Walker, B. J., Abeel, T., Shea, T., Priest, M., Abouelliel, A., Sakthikumar, S., Cuomo, C. A., Zeng, Q., Wortman, J., Young, S. K., & Earl, A. M. (2014). Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLOS ONE*, *9*(11), e112963. https://doi.org/10.1371/journal.pone.0112963

Wick, R. R., Judd, L. M., Gorrie, C. L., & Holt, K. E. (2017). Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. PLoS computational biology, 13(6), e1005595. https://doi.org/10.1371/journal.pcbi.1005595