
Genome Analysis

Tychus: a whole genome sequencing pipeline for assembly, annotation and phylogenetics of bacterial genomes

Christopher Dean^{1,†}, Noelle Noyes^{1,†}, Steven Lakin^{1,†}, Pablo Rovira-Sanz³, Xiang Yang⁴, Keith Belk³, Paul S. Morley², Rick Meinersmann⁵, Zaid Abdo^{1,*}

¹Department of Microbiology, Immunology and Pathology, ²Department of Clinical Sciences, ³Department of Animal Sciences, Colorado State University, Fort Collins, CO, 80523, USA

⁴Department of Animal Science, University of California, Davis, CA, 95616, USA, and

⁵Bacterial Epidemiology and Antimicrobial Resistance Research, USDA-Agricultural Research Service, Athens, Georgia, USA

Abstract

Summary: Tychus is a tool that allows researchers to perform massively parallel whole genome sequence (WGS) analysis with the goal of producing a high confidence and comprehensive description of the bacterial genome. Key features of the Tychus pipeline include the assembly, annotation, alignment, variant discovery and phylogenetic inference of large numbers of WGS isolates in parallel using open-source bioinformatics tools and virtualization technology. All prerequisite tools and dependencies come packaged together in a single suite that can be easily downloaded and installed on Linux and Mac operating systems.

Availability: Tychus is freely available as an open-source package under the MIT license, and can be downloaded via GitHub (<https://github.com/Abdo-Lab/Tychus>).

Contact: zaid.abdo@colostate.edu

1 Introduction

The zeitgeist of the genomics era has been defined by the accessibility and affordability of next-generation sequencing platforms, which are capable of producing large amounts of data for genomics research. Whole genome sequencing (WGS) allows for the interrogation and analysis of complete genomes and can be applied to a wide range of organisms including plants, animals, bacteria, and viruses. Recently, WGS methods have been used in the domain of public health in order to better understand and track foodborne illnesses caused by bacterial pathogens. These efforts have led to the development of open-source reference databases and foodborne outbreak detection and investigation initiatives (Barkley et al., 2016; see also Hu et al., 2016). However, the accuracy of WGS for bacterial pathogen identification and outbreak analysis relies not only on the methods used to isolate, extract, and sequence the bacterial DNA, but also on the bioinformatics and statistical analyses applied to the resulting sequence data. In addition, as WGS datasets expand exponentially, there is a critical need for widely-accessible and computationally-efficient analytic pipelines. To this end, we present Tychus, an open-source, easy-install, user-friendly and rapid software tool for performing large-scale, parallel bacterial sequence analysis that produces a high confidence and comprehensive description of the bacterial genome.

* To whom correspondence should be addressed.

† The authors wish it to be known that in their opinion, the first three authors should be regarded as joint first authors.

2 Methods and Implementation

Tychus is written using Nextflow (Tommaso *et al.*, 2015), a parallel DSL workflow framework and integrated with Docker (Merkel, 2014), a software containerization platform that resolves the installation and configuration issues of the many open-source bioinformatics tools utilized throughout the pipeline. Tychus is split into two complementary modules (see Figure 1): assembly and alignment, as described below.

2.1 Assembly Module

Because no single assembler consistently produces an optimal genome assembly (Bradnam et al., 2013) in isolation, we utilize the results of four *de novo* genome assemblers to construct a consensus assembly of higher contiguity and accuracy. This module also supports methods for identifying and classifying genomic features of interest, a process called annotation, as well as reporting on common quality score metrics for each resulting assembly.

Input to the Tychus assembly module consists of a collection of paired fastq files. These files are preprocessed using Trimmomatic (Bolger, Lohse, & Usadel, 2014) to remove adapter sequences, low quality base pairs and sequence fragments. This prevents adapter sequences from being used to build the resulting assemblies, and has been shown to improve the quality of *de novo* assemblies as measured by their N50 statistic (2014). Preprocessed reads are then assembled with Abyss

Dean et al.

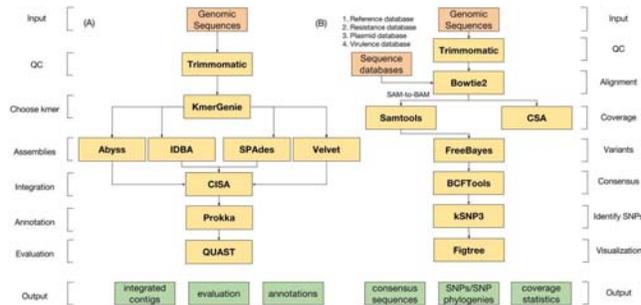


Fig. 1. Implementation diagram for each step in the assembly and alignment modules (inputs, tools utilized, and outputs).

(Simpson et al., 2009), IDBA-UD (Peng et al., 2012), SPAdes (Bankevich et al., 2012), and Velvet (Zerbino & Birney, 2008). Prior to *de novo* assembly, an important step is to choose a *k*-mer value with which to build the underlying De Bruijn graph. As Abyss and Velvet do not iterate on multiple *k*-mer values, KmerGenie (Chikhi & Medvedev, 2014) is used to optimize this value prior to assembly. Contigs produced from each assembler are combined and integrated using CISA (Lin & Liao, 2013) to construct a super (consensus) assembly of higher quality and contiguity. Prokka (Seemann, 2014) is then used to annotate the super assembly. Lastly, contigs are evaluated with QUAST (Gurevich et al., 2013) to evaluate various common assembly score metrics, such as number of contigs, largest contig, and N50.

2.2 Alignment Module

The Tychus alignment module takes in six inputs: a collection of paired-end fastq files, a fasta-formatted reference genome, and optional databases of plasmid, resistance, virulence factor, and draft genome nucleotide sequences. Similar to the assembly module, reads are first quality-filtered using Trimmomatic (Bolger, Lohse, & Usadel, 2014), which helps to decrease the number of misalignments downstream. Next, Bowtie2 (Langmead & Salzberg, 2012) is used to align reads to the user-input reference genome, as well as the plasmid, resistance, and virulence databases. The results from alignment to the resistance, virulence and plasmid databases are used by an in-house tool to determine the overall coverage of each feature (plasmids, virulence factors, and antimicrobial resistance genes) present in each sample, with higher coverage features reducing the number of false-positive gene identifications. Files resulting from alignment to the reference genome are utilized by Freebayes (Garrison & Marth, 2012) to identify sequence variants and single nucleotide polymorphisms (SNPs), which are utilized by BCFtools (Li, 2011) to obtain consensus sequences for each sample. These consensus sequences, in addition to the optional genomes produced from the assembly module, act as draft genomes, which can be used by kSNP3 (Gardner, Slezak, & Hall, 2015) to identify related SNPs and build SNP phylogenies. The Newick formatted phylogenies produced by kSNP3 (2015) are then rendered into a user-defined image format using the command-line version of Figtree (<https://github.com/rambaut/figtree>).

3 Case Study

We used 15 *L. monocytogenes* samples (acc. no. PRJNA374745) with a reference genome (acc. no. PRJNA61583) and a 64-core (AMD Opteron Processor 6378) Ubuntu server to provide results for a common use case

Table 1. Assembly results for 15 *L. monocytogenes* samples.

	S935	S937	S938	S940	S941	S942	S943	S944	S956	S993	S995	S997	S999	S101	S103
Abyss															
# contigs	59	120	63	27	31	70	33	127	19	32	47	31	47	14	10
Largest contig	335,900	207,467	329,432	550,131	746,468	294,456	378,754	159,856	595,207	406,585	344,175	824,747	336,382	1,531,175	1,160,011
N50	109,496	89,120	127,399	312,769	247,290	65,681	257,693	39,992	390,544	203,305	207,506	350,896	332,132	543,926	421,870
# average	20	10	30	24	24	29	32	40	20	24	25	31	24	27	24
IDBA															
# contigs	608,650	434,230	546,931	891,939	891,939	860,794	434,896	433,202	1,243,429	779,103	542,550	891,919	745,546	1,256,470	895,457
Largest contig	275,937	197,466	327,831	568,290	568,290	332,476	333,396	193,969	577,456	397,832	323,681	346,156	438,456	524,909	291,726
N50	109,130	178,796	309,216	368,609	368,609	181,626	424,369	222,398	376,264	439,466	306,936	346,876	317,617	519,476	433,610
# average	14	97	43	22	22	40	21	18	16	34	27	22	29	15	16
SPAdes															
# contigs	609,844	363,494	571,501	1,221,296	1,221,296	579,209	1,026,362	424,096	1,245,062	1,046,660	711,796	1,221,268	745,826	1,512,476	1,162,026
Largest contig	301,130	178,796	309,216	368,609	368,609	181,626	424,369	222,398	376,264	439,466	306,936	346,876	317,617	519,476	433,610
N50	109,130	178,796	309,216	368,609	368,609	181,626	424,369	222,398	376,264	439,466	306,936	346,876	317,617	519,476	433,610
# average	16	95	43	22	22	40	21	18	16	34	27	22	29	15	16
Velvet															
# contigs	177,720	164,049	343,809	501,114	482,848	346,140	486,043	213,403	854,880	344,884	348,896	553,064	349,063	1,511,967	1,161,688
Largest contig	96,843	164,049	164,049	270,705	292,040	179,616	279,496	94,961	949,660	374,446	181,506	369,496	379,240	1,511,967	1,161,688
N50	11	35	8	10	9	14	11	23	11	12	14	8	13	7	9
# average	11	35	8	10	9	14	11	23	11	12	14	8	13	7	9
CISA															
# contigs	609,866	435,935	605,168	1,222,847	1,222,701	663,416	1,026,165	434,847	1,246,139	1,046,362	713,473	1,213,366	745,870	1,512,476	1,162,026
Largest contig	301,130	178,796	309,216	368,609	368,609	181,626	424,369	222,398	376,264	439,466	306,936	346,876	317,617	519,476	433,610
N50	109,130	178,796	309,216	368,609	368,609	181,626	424,369	222,398	376,264	439,466	306,936	346,876	317,617	519,476	433,610

for each Tychus module. By default, all 15 samples were run in parallel using all available computing cores.

The assembly module was run with default parameters for each sample. Assemblies, annotations, and all assembly summary statistics were computed in under an hour, using approximately 40 gigabytes (GB) of RAM. Consistent with previous findings (Lin & Liao, 2013) the CISA integrated contigs produced superior assemblies in terms of N50, contig length, and number of contigs in nearly all samples (see Table 1).

Similarly, the alignment module was run with default settings for each sample. All sequence alignments, consensus sequences, SNPs, SNP phylogenies, and phylogenetic trees were produced in under 30 minutes using approximately 18 GB of RAM. A total of 77,601 SNPs were identified from all 15 strains, with 36,503 of these identified as core (SNPs present in all genomes); 41,098 identified as non-core; and 56,894 identified as majority SNPs present in a user-defined fraction of all samples (default 0.75).

In the end, it took Tychus <2 hours and <41GB of RAM to turn raw WGS data for 15 samples into high-quality assemblies with complete annotation and robust phylogenetic trees.

Conclusion

Tychus is built upon existing open-source virtualization technology and provides a pipeline framework that is intuitive and easy-to-use for both novices and developers. It takes advantage of emerging multi-core computing architectures and large memory servers to deliver results quickly and reliably without the need for extensive bioinformatics or computing expertise. Though Tychus can be used on modest machines (with a penalty to run time), it is intended for use on servers with large amounts of RAM and disk space and with multiple computing cores.

Acknowledgements

Funding: This work has been supported by startup funding from Colorado State University provided to ZA.

Conflict of Interest: none declared.

References

Barkley, J. S., Goscinski, M., & Miller, A. (2016). Whole-Genome Sequencing Detection of Ongoing *Listeria* Contamination at a Restaurant, Rhode Island, USA, 2014. *Emerging Infectious Diseases*, 22(8), 1474–1476. <https://doi.org/10.3201/eid2208.151917>

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., ... Pevzner, P. A. (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>

Bradnam, K. R., Fass, J. N., Alexandrov, A., Baranay, P., Bechner, M., Birol, I., ... Korf, I. F. (2013). Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species. *GigaScience*, 2(1), 10. <https://doi.org/10.1186/2047-217X-2-10>

Dean et. al

- Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina Sequence Data. *Bioinformatics*, *btu170*. <https://doi.org/10.1093/bioinformatics/btu170>
- Garrison, E., & Marth, G. (2012). Haplotype-based variant detection from short-read sequencing. *arXiv:1207.3907 [q-Bio]*. Retrieved from <http://arxiv.org/abs/1207.3907>
- Hu, K., Renly, S., Edlund, S., Davis, M., & Kaufman, J. (2016). A modeling framework to accelerate food-borne outbreak investigations. *Food Control*, *59*, 53–58. <https://doi.org/10.1016/j.foodcont.2015.05.017>
- Li, H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, *27*(21), 2987–2993. <https://doi.org/10.1093/bioinformatics/btr509>
- Lin, S.-H., & Liao, Y.-C. (2013). CISA: Contig Integrator for Sequence Assembly of Bacterial Genomes. *PLoS ONE*, *8*(3). <https://doi.org/10.1371/journal.pone.0060843>
- Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux J.*, *2014*(239). Retrieved from <http://dl.acm.org/citation.cfm?id=2600239.2600241>
- Peng, Y., Leung, H. C. M., Yiu, S. M., & Chin, F. Y. L. (2012). IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics (Oxford, England)*, *28*(11), 1420–1428. <https://doi.org/10.1093/bioinformatics/bts174>
- Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics (Oxford, England)*, *30*(14), 2068–2069. <https://doi.org/10.1093/bioinformatics/btu153>
- Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. M., & Birol, I. (2009). ABySS: A parallel assembler for short read sequence data. *Genome Research*, *19*(6), 1117–1123. <https://doi.org/10.1101/gr.089532.108>
- Simpson, Jared T., Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J.M. Jones, and Inanc Birol. “ABySS: A Parallel Assembler for Short Read Sequence Data.” *Genome Research* 19, no. 6 (June 2009): 1117–23. doi:10.1101/gr.089532.108.
- Tommaso, P. D., Chatzou, M., Prieto, P., Palumbo, E., Notredame, C. (2014). A novel tool for highly scalable computational pipelines. figshare. <https://doi.org/https://doi.org/10.6084/m9.figshare.1254958.v2>
- Zerbino, D. R., & Birney, E. (2008). Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, *18*(5), 821–829. <https://doi.org/10.1101/gr.074492.107>