MetaSanity: An integrated, customizable microbial genome evaluation and annotation pipeline

Christopher J Neely[1#], Elaina D Graham[1], Benjamin J Tully[1,2#]

[1] Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089, USA

[2] Center for Dark Energy Biospheres Investigations, University of Southern California, Los Angeles, CA 90089, USA

[#] corresponding author(s)

Abstract

Summary

As the importance of microbiome research continues to become more prevalent and essential to understanding a wide variety of ecosystems (e.g., marine, built, host-associated, etc.), there is a need for researchers to be able to perform highly reproducible and quality analysis of microbial genomes. MetaSanity incorporates analyses from eleven existing and widely used genome evaluation and annotation suites into a single, distributable workflow, thereby decreasing the workload of microbiologists by allowing for a flexible, expansive data analysis pipeline. MetaSanity has been designed to provide separate, reproducible workflows, that (1) can determine the overall quality of a microbial genome, while providing a putative phylogenetic assignment, and (2) can assign structural and functional gene annotations with varying degrees of specificity to suit the needs of the researcher. The software suite combines the results from several tools to provide broad insights into overall metabolic function and putative extracellular localization of peptidases and carbohydrate-active enzymes. Importantly, this software provides built-in optimization for "big data" analysis by storing all relevant outputs in an SQL database, allowing users to query all the results for the elements that will most impact their research.

Availability and implementation

MetaSanity is provided under the GNU General Public License v.3.0 and is available for download at https://github.com/cjneely10/MetaSanity. This application is distributed as a Docker image. MetaSanity is implemented in Python3/Cython and C++.

Supplementary information

Supplementary data are available below.

1 Introduction

The analysis of microbial genomes has become an increasingly common task for many fields of biology and geochemistry. Researchers can routinely generate hundreds/thousands of environmentally derived microbial genomes using methodologies such as metagenomics (Tully et al., 2018), high-throughput culturing (Thrash et al., 2015), and single cell sorting (Stepanauskas et al., 2017). However, analyzing the data can be problematic, as data analysis is

computationally intensive and requires a knowledge of software that is constantly changing and may be difficult to install or execute. For the average researcher, the task of evaluating and annotating a set of microbial genomes may be time intensive and computationally rigorous. Here, we present MetaSanity, a comprehensive and customizable solution for generating evaluation and annotation pipelines for bacterial and archaeal isolate genomes, metagenome-assembled genomes (MAGs), and single-amplified genomes (SAGs). MetaSanity provides genome quality evaluation, phylogenetic assignment, as well as structural and functional annotation through a variety of integrated programs based on the procedure described in ref. Tully (2019). MetaSanity provides a workflow that combines all outputs into a single queryable database that operates easily from the command line. Installation can be performed at the user level, limiting the need for intervention by system administrators, and, except for certain memory intensive programs, can be run locally on high-end personal computers.

2 Description of Methods

MetaSanity consists of two smaller workflows (Figure 1): (1) PhyloSanity, to evaluate the completion, contamination, redundancy, and phylogeny of each genome in a dataset, and (2) FuncSanity, to provide structural and functional annotations of each genome. Each component consists of several optional applications that can be customized to specific research needs. While each component contained within the two pipelines runs independently and generates component specific outputs, MetaSanity combines all outputs into a single queryable SQL database that allows fast and easy retrieval of data – in this case, gene annotations and other related genomic data. MetaSanity focuses on allowing users the ability to fine-tune and customize their data analysis pipelines with minimal effort and maximized computational and storage efficiency (Supplemental Table 1). MetaSanity is distributed as a Docker image (Merkel et al., 2014) and is implemented using a combination of Python3 (Python Software Foundation 2014) /Cython (Bradshaw et al., 2011) and C++ (ISO/IEC 2014).

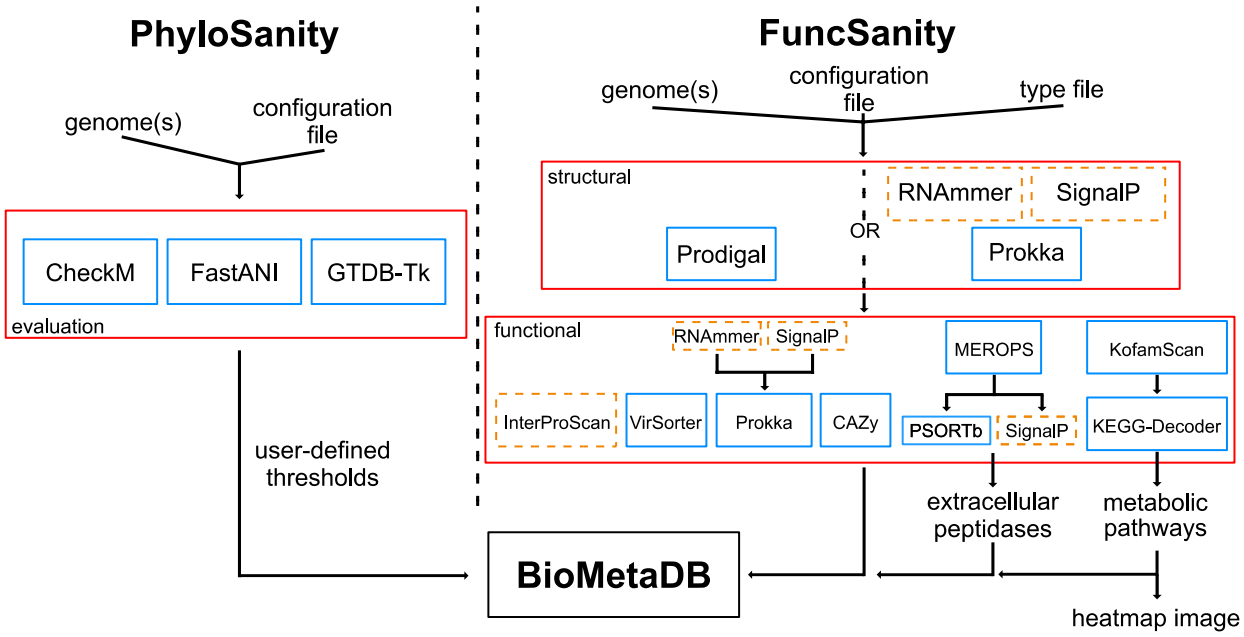


Figure 1 MetaSanity pipeline schema. Programs and databases that are part of the MetaSanity installation are in blue boxes. Programs in the dotted orange boxes must be installed separately by the user due to licensing agreements.

2.1 PhyloSanity

PhyloSanity is designed to provide metrics of genome quality and to filter genomes for downstream analysis based on user defined quality metrics. The workflow integrates CheckM v1.0.18 (Parks et al., 2015), GTDB-Tk v.0.3.2 (Parks et al., 2018), and FastANI (Jain et al., 2019) as part of its evaluation pipeline. CheckM estimates the completion and contamination of each genome (Parks et al., 2015). Next, FastANI compares each genome in a pairwise fashion against all other genomes to determine the average nucleotide identify (ANI) for each genome pair (Jain et al., 2019). For any set of genomes that shares an ANI above a user-defined value, a non-redundant genome representative will be selected from the set that is the most complete and least contaminated. This allows users the option to exclude redundant genomes from further analysis. Differentiating genomes as non-redundant versus redundant can be useful for researchers working with MAGs or SAGs that are generated from replicate samples and may not have biological meaning when working with isolates or strain level differences. All genomes can undergo phylogenetic assignment based on relative evolutionary distance (Parks et al., 2018) through GTDB-Tk, which will replace the CheckM-returned taxonomic assignment.

2.2 FuncSanity

FuncSanity provides structural and functional annotation of microbial genomes. The workflow incorporates annotation suites from eight existing and widely used programs. The use of multiple annotation programs has the advantage of capturing functional predictions that may not have been detected due to database or search limitations. Specialized annotation programs, such as VirSorter (Roux et al., 2015), use custom tools and/or databases to return relevant annotations that are not captured by other programs in MetaSanity. Open reading frames (ORFs) are predicted using Prodigal v2.6.3 (Hyatt et al., 2010); however, users may opt to use the putative coding DNA sequences (CDS) generated by Prokka v1.13.3 (Seeman, 2014). From here, putative ORFs are processed by a set of annotation tools that can be selected by the user with user-defined filtering and cutoff values.

Kyoto Encyclopedia of Genes and Genomes (KEGG) Annotation

Putative ORFs can be searched against the KofamKOALA database using KofamScan v.1.1.0 (Aramaki et al., 2019). Default parameters are used and the 'mapper' tab-delimited output option is generated, linking ORF IDs to KEGG Ontology (KO) IDs. Users can query any KO ID to generate specific functional search results in BioMetaDB.

KEGG-Decoder

KEGG annotations can be used to estimate the completeness of various biogeochemically-relevant metabolic pathways in a genome using KEGG-Decoder v.1.0.1 (Graham et al., 2018; https://github.com/bjtully/BioData/tree/master/KEGGDecoder). Users can search genomes based on completeness of a pathway or function of interest. An additional heatmap summary visualization is generated.

VirSorter

VirSorter v1.0.5 (Roux et al., 2015) can be implemented to identify phage and prophage signatures in each genome using default parameters. Users can search for matches to each of the phage and prophage categories returned by VirSorter and generate lists of contigs and/or genomes with the assignments (Supplementary Table 1).

InterProScan

InterProScan 5.36-75.0 (Jones et al., 2019) is an optional installation and can be used for domain prediction on putative ORFs. Users have the option of downloading all of the InterProScan databases, including TIGRfam (Haft et al., 2003), Pfam (Finn et al., 2016), CDD (Marchler-Bauer et al., 2017), and PANTHER (Mi et al., 2019). Each InterProScan database result is indexed separately in BioMetaDB and can be used to return matching genomes using database specific IDs (e.g., PF01036 would return putative rhodopsin ORFs from a Pfam result).

Prokka Annotation

If not chosen as the option for structural annotation, genomes can be annotated using Prokka and its associated databases with the parameters --addgenes (adds the "gene" feature to each CDS in the GenBank output format), --addmrna (adds the "mRNA" feature to each CDS in the GenBank output format), --usegenus (use the genus-specific databases), --metagenome (improve gene predictions for fragmented genomes), and --rnammer (sets RNAmmer as the preferred rRNA prediction tool). rRNA identification with RNAmmer v.1.2 (Lagesen et al., 2007) and signal peptide detection with SignalP v.4.1 (Nielson, 2017) are optional installations.

Carbohydrate-active enzyme (CAZy) Annotation

Putative ORFs can be assigned a putative CAZy functionality (Cantarel et al., 2009) based on the dbCANv2 database (Zhang et al., 2018). ORFs are searched against dbCANv2 using HMMER v3.1b2 (Eddy, 2011) with the minimum score threshold set to 75 (-T parameter). PSORTb v.3.0 (Yu et al., 2010) and SignalP can be optionally performed on CAZy matches to determine if a putative enzyme is predicted to be extracellular. An extracellular assignment is made if PSORTb predicts "extracellular" or "outer membrane" localization or if PSORTb returns "unknown" localization and SignalP predicts the presence of a signal peptide. Users can search for genes and genomes based on overall CAZy annotations or by searching for specific designations (e.g., GT41 for glycosyl transferase family 41).

Peptidase Annotation

Putative ORFs can be assigned to a peptidase family using a set of HMMs that represent the MEROPS database (Rawlings et al., 2013). The putative extracellular nature of a MEROPS match can be determined as above. Users can search for genes and genomes based on overall MEROPS annotations or by searching specific peptidase families.

InterProScan, SignalP, and RNAmmer are not automatically distributed with MetaSanity and require users to download their binaries separately and agree to their individual license requirements.

2.3 BioMetaDB

BioMetaDB is a specialized relational database management system project that integrates modularized storage and retrieval of FASTA records with the metadata describing them. This application uses tab-delimited data files to generate table relation schemas via Python3. Based on SQLAlchemy v.1.3.7 (Bayer, 2012), BioMetaDB allows researchers to efficiently manage data from the command line by providing operations that include: (1) the ability to store information from any valid tab-delimited data file and to quickly retrieve FASTA records or annotations related to these datasets by using SQL-optimized command-line queries; and, (2) the ability to run all CRUD operations (create, read, update, delete) from the command line and from python scripts. Output from both workflows is stored into a BioMetaDB project, providing users a simple interface to comprehensively examine their data (Supplemental Table 2). Users can query application results used across the entire genome set for specific information that is relevant to their research, allowing the potential to screen genomes based on returned taxonomy, quality, annotation, putative metabolic function, or any combination thereof.

3 Results

MetaSanity was tested on two separate systems – a personal computer with an Intel core i5-4570 CPU @ 3.20 GHz processor with 4 cores and 32 GB of RAM operating the Deepin 15.11 Linux distribution, and an academic server with an Intel Xeon E7-4850 v2 @ 2.30 GHz processor with 96 cores and 1 TB of RAM operating the Ubuntu 18.04.3 LTS Linux distribution. Reduced options were calculated on the personal computer using all four available threads and preset parameter flags that skip memory intensive processes. Complete options were calculated on the academic server using 10 threads and no parameters to reduce memory usage. Runtime results are available in Supplemental Table 3. The current architecture relies on sequential completion of time intensive processes, several of which are optional for users. Ongoing modifications that take advantage of parallelizing these processes should decrease the overall computation time.

169

Workflow	Calling Program	Available Flags
PhyloSanity	CheckM	--aai_strain -t --pplacer_threads --unique --e_value --length --reduced_tree --force_domain
	FastANI	--fragLen --kmer --threads --minFrag
	GTDB-Tk	--cpus --min_perc_aa
FuncSanity	Prodigal	-p -m -c
	HMMSearch	-T --cpu
	Diamond	--threads
	PSORTb	--cutoff --divergent -M -c

	Kofamscan	--cpu
	Prokka	--addgenes, --addmra, --usegenus, --metagenome, --rnammer, --force --evalue, --cpus --mincontiglen, --norrna, --notrna, --rfam
	InterProScan	--applications --cpu --minsize --goterms, --iprlookup --pathways
	VirSorter	--db --ncpu --virome --diamond

Supplementary Table 1. Parameters flags available to each program within the MetaSanity workflows. Modification to the configuration files will allow users may include any set of these flags for specific analyses.

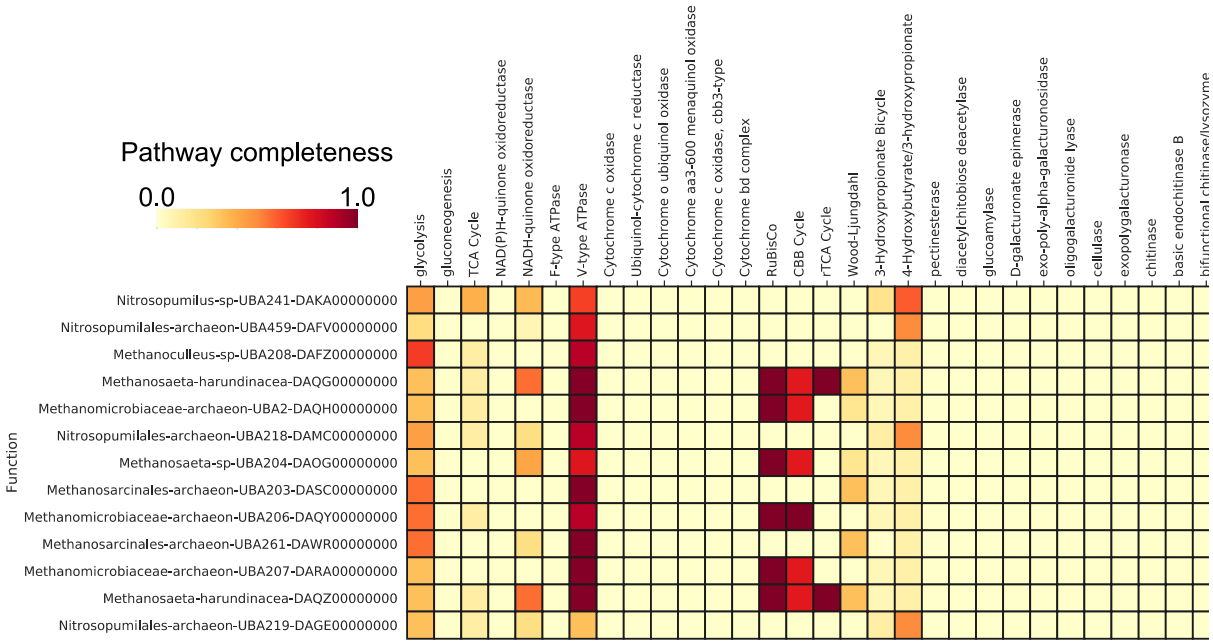
Workflow	**Name of table generated**	**Database table information**	
		Program	Searchable fields
PhyloSanity	"evaluation"	CheckM	completion contamination domain
		FastANI	redundant_copies
		GTDB-Tk	domain, phylum, _class, _order, family, genus, species
		Added	is_complete is_contaminated

			is_nonredundant
FuncSanity	"genome-id"	Prokka	prokka
		VirSorter	phage_contig_1 phage_contig_2 phage_contig_3 prophage_1 prophage_2 prophage_3
		Kofamscan	ko
		CAZy	cazy
		MEROPS	merops_pfam
		PSORTb/ SignalP	is_extracellular
		InterProScan	cdd, hamap, panther, pfam, prodom, sfld, smart, superfamily, tigrfam
	"functions"	Peptidase	<MEROPS and CAZy designations>
		KEGG-Decoder	<Metabolic/functional pathways>

Supplementary Table 2. Underlying database structure and queryable objects from the complete MetaSanity workflow.

	PhyloSanity		**FuncSanity**	
	No GTDB-Tk; CheckM reduced_tree	Complete evaluation	Recommended & optional; no InterProScan	Complete annotation
1 genome	0.05 hr	0.63 hr	0.52 hr	1.62 hr
10 genomes	0.15 hr	1.02 hr	6.17 hr	20.3 hr

Supplemental Table 3. Runtimes of each core component of the MetaSanity pipeline. InterProScan search ran using the databasesTIGRFAM, SFLD, SMART, SUPERFAMILY, Pfam, ProDom, Hamap, CDD, and PANTHER, with parameter flags --goterms, --iprlookup, and --pathways.



Supplemental Figure 1. Example KEGG-Decoder heatmap output. The completeness of various biogeochemically-relevant pathways for a collection of marine metagenome-assembled genomes, scaled from 0.0-1.0.

Acknowledgements

We would like to thank Taylor Reiter, Roth Conrad, Jay Osvatic, and Luiz Irber for providing code contributions to KEGG-Decoder as part of the Moore Foundation funded 'Speeding Up Science' hackathon. This is C-DEBI Contribution XXX.

Funding

This work was supported by the National Science Foundation Science and Technology Center, the Center for Dark Energy Biosphere Investigations (C-DEBI) [OCE- 0939654 to B.J.T.]. C.J.N. was supported by the University of Southern California SOAR program.

References

Aramaki T., Blanc-Mathieu R., Endo H., Ohkubo K., Kanehisa M., Goto S., Ogata H. (2019). KofamKOALA: KEGG ortholog assignment based on profile HMM and adaptive score threshold. *bioRxiv* doi: https://doi.org/10.1101/602110.

Bayer, M. (2012). SQLAlchemy. In Amy Brown and Greg Wilson, editors, "The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks." http://aosabook.org.

Bradshaw, R., Behnel S., Seljebotn D.S., Ewing, G., et al. (2011). The Cython compiler. http://cython.org.

Buchfink B., Xie C., Huson D. H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nature Methods 12*, 59-60. doi:10.1038/nmeth.3176.

Camacho C., Coulouris G., Avagyan V., Ma N., Papadopoulos J., Bealer K., & Madden T.L. (2008) "BLAST+: architecture and applications." *BMC Bioinformatics* 10:421 https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-421.

Cantarel, B. L., Coutinho, P. M., Rancurel, C., Bernard, T., Lombard, V., & Henrissat, B. (2009). The Carbohydrate-Active EnZymes database (CAZy): an expert resource for Glycogenomics. *Nucleic Acids Research, 37(Database)*, D233–D238. http://doi.org/10.1093/nar/gkn663.

Eddy S.R. (2011) Accelerated Profile HMM Searches. *PLoS Comput Biol 7(10)*: e1002195. https://doi.org/10.1371/journal.pcbi.1002195.

Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., et al. (2016). The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, *44*(D1), D279–D285. http://doi.org/10.1093/nar/gkv1344.

Graham E. D., Heidelberg J. F., Tully B. J. (2018) Potential for primary productivity in a globally-distributed bacterial phototroph. *ISME J 350*, 1–6.

Haft, D. H., Selengut, J. D., & White, O. (2003). The TIGRFAMs database of protein families. *Nucleic Acids Research, 31(1)*, 371–373. http://doi.org/10.1093/nar/gkg128.

Hyatt, D., Chen, G. L., Locascio, P. F., Land, M. L., Larimer, F. W., & Hauser, L. J. (2010). Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics*, *11*, 119. doi:10.1186/1471-2105-11-119.

ISO/IEC. (2014). ISO International Standard ISO/IEC 14882:2014(E) – Programming Language C++. [Working draft]. Geneva, Switzerland: International Organization for Standardization (ISO). Retrieved from https://isocpp.org/std/the-standard.

Jain C., et al. 2019. High-throughput ANI Analysis of 90K Prokaryotic Genomes Reveals Clear Species Boundaries. *Nature Communications*, doi: 10.1038/s41467-018-07641-9.

Jones, P., Binns, D., Chang, H. Y., Fraser, M., Li, W., McAnulla, C., … Hunter, S. (2014). InterProScan 5: genome-scale protein function classification. *Bioinformatics (Oxford, England), 30(9)*, 1236–1240. doi:10.1093/bioinformatics/btu031.

Lagesen, K., Hallin, P., Rødland, E. A., Staerfeldt, H. H., Rognes, T., & Ussery, D. W. (2007). RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic acids research, 35(9)*, 3100–3108. doi:10.1093/nar/gkm160.

Marchler-Bauer A., Bo Y., Han L., He J., Lanczycki C. J., Lu S., Chitsaz F., Derbyshire M. K., Geer R. C., Gonzales N. R., Gwadz M., Hurwitz D. I., Lu F., Marchler G. H., Song J. S., Thanki N., Wang Z., Yamashita R. A.,

Matsen F. A., Kodner R. B., Armbrust E. V. (2010). pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics 11*: doi:10.1186/1471-2105-11-538.

Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux J*.

Mi, H., Muruganujan, A., Huang, X., Ebert, D., Mills, C., Guo, X., & Thomas, P. D. (2019). Protocol Update for large-scale genome and gene function analysis with the PANTHER classification system (v.14.0). *Nature Protocols*, 1–21. http://doi.org/10.1038/s41596-019-0128-8.

Nielsen H. (2017) Predicting Secretory Proteins with SignalP. In: Kihara D. (eds) *Protein Function Prediction. Methods in Molecular Biology, vol 1611*. Humana Press, New York, NY.

Parks, D. H., Chuvochina, M., Waite, D. W., Rinke, C., Skarshewski, A., Chaumeil, P.-A., & Hugenholtz, P. (2018). A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology*, *15*, 1–14. http://doi.org/10.1038/nbt.4229.

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*: 1043–1055.

Python Software Foundation. Python Language Reference, version 3. http://www.python.org

Rawlings, N. D., Waller, M., Barrett, A. J., & Bateman, A. (2013). MEROPS: the database of proteolytic enzymes, their substrates and inhibitors. *Nucleic Acids Research, 42(D1)*, D503–D509. http://doi.org/10.1093/nar/gkt953.

Roux, S., Enault, F., Hurwitz, B. L., & Sullivan, M. B. (2015). VirSorter: mining viral signal from microbial genomic data. *PeerJ, 3*, e985. https://doi.org/10.7717/peerj.985.

Seemann T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics 30(14)*, 2068-9. PMID:24642063.

Stepanauskas, R., Fergusson, E. A., Brown, J., Poulton, N. J., Ben Tupper, Labonté, J. M., et al. (2017). Improved genome recovery and integrated cell-size analyses of individual uncultured microbial cells and viral particles. *Nature Communications*, *8(1)*, 1–10. http://doi.org/10.1038/s41467-017-00128-z.

Thrash, J. C., Weckhorst, J. L., & Pitre, D. M. (2015). Cultivating Fastidious Microbes. In *Hydrocarbon and Lipid Microbiology Protocols* (Vol. 39, pp. 57–78). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/8623_2015_67.

Tully, B. J. (2019). Metabolic diversity within the globally abundant Marine Group II Euryarchaea offers insight into ecological patterns. *Nature Communications*, *10(1)*, 1–12. http://doi.org/10.1038/s41467-018-07840-4.

Tully, B. J., Graham, E. D., & Heidelberg, J. F. (2018). The reconstruction of 2,631 draft metagenome-assembled genomes from the global oceans. *Scientific Data*, *5*, 170203. http://doi.org/10.1038/sdata.2017.203.

Yu, N. Y., Wagner, J. R., Laird, M. R., Melli, G., Rey, S., Lo, R., Dao, P., Sahinalp, S. C., Ester, M., Foster, L.J., and Brinkman, F. S. L. (2010). PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. *Bioinformatics 26(13)*:1608-1615.

Zhang D., Zheng C., Geer L. Y., Bryant S. H. (2017). CDD/SPARCLE: functional classification of proteins via subfamily domain architectures. *Nucleic Acids Research 45(D1)*: D200-D203. doi: 10.1093/nar/gkw1129.

Zhang, H., Yohe, T., Huang, L., Entwistle, S., Wu, P., Yang, Z., et al. (2018). dbCAN2: a meta server for automated carbohydrate-active enzyme annotation. *Nucleic Acids Research*, *46(W1)*, W95–W101. http://doi.org/10.1093/nar/gky418.