BioKIT: a versatile toolkit for processing and analyzing diverse types of sequence data

Jacob L. Steenwyk[1,*], Thomas J. Buida III[2], Carla Gonçalves[3,4], Dayna C. Goltz[5], Grace Morales[O], Matthew E. Mead[1], Abigail L. LaBella[1], Christina M. Chavez[1], Jonathan E. Schmitz[6], Maria Hadjifrangiskou[6], Yuanning Li[1], & Antonis Rokas[1,*]

[1] Vanderbilt University, Department of Biological Sciences, VU Station B #35-1634, Nashville, TN 37235, United States of America

[2] 9 City Place #312, Nashville, TN 37209, United States of America

[3] Associate Laboratory i4HB - Institute for Health and Bioeconomy, NOVA School of Science and Technology, NOVA University Lisbon, Caparica, Portugal

[4] UCIBIO – Applied Molecular Biosciences Unit, Department of Life Sciences, NOVA School of Science and Technology, NOVA University Lisbon, Caparica, Portugal

[5] 2312 Elliston Place #510, Nashville, TN 37203, United States of America

[6] Vanderbilt University Medical Center, Department of Pathology, microbiology & immunology, Center for Personalized Microbiology

*Correspondence should be addressed to: jacob.steenwyk@vanderbilt.edu or antonis.rokas@vanderbilt.edu

ORCiDs

J.L. Steenwyk: 0000-0002-8436-595X

T.J. Buida III: 0000-0001-9367-6189

C. Gonçalves: 0000-0002-0420-4970

G. Morales: 0000-0001-5454-9195

M. Mead: 0000-0001-9195-7585

A.L. LaBella 0000-0003-0068-6703

Y. Li: 0000-0002-2206-5804

A. Rokas: 0000-0002-7248-6551

Running title: BioKIT: a toolkit for processing and analyzing sequence data

Keywords: multiple sequence alignment, genome assembly quality, codon, bioinformatics, gene-wise relative synonymous codon usage, genetic code

Abstract

Bioinformatic analysis—such as genome assembly quality assessment, alignment summary statistics, relative synonymous codon usage, paired-end aware quality trimming and filtering of sequencing reads, file format conversion, and processing and analysis—is integrated into diverse disciplines in the biological sciences. Several command-line pieces of software have been developed to conduct some of these individual analyses; however, the lack of a unified toolkit that conducts all these analyses can be a barrier in workflows. To address this obstacle, we introduce BioKIT, a versatile toolkit for the UNIX shell environment with 40 functions, several of which were community-sourced, that conduct routine and novel processing and analysis of genome assemblies, multiple sequence alignments, coding sequences, sequencing data, and more. To demonstrate the utility of BioKIT, we assessed the quality and characteristics of 901 eukaryotic genome assemblies, calculated alignment summary statistics for 10 phylogenomic data matrices, determined relative synonymous codon usage across 171 fungal genomes including those that use alternative genetic codes, and demonstrate that a novel metric, gene-wise relative synonymous codon usage, can accurately estimate gene-wise codon optimization. BioKIT will be helpful in facilitating and streamlining sequence analysis workflows. BioKIT is freely available under the MIT license from GitHub (https://github.com/JLSteenwyk/BioKIT), PyPi (https://pypi.org/project/biokit), and the Anaconda Cloud (https://anaconda.org/JLSteenwyk/biokit). Documentation, user tutorials, and instructions for requesting new features are available online (https://jlsteenwyk.com/BioKIT).

Introduction

Bioinformatics is the application of computational tools to process and analyze biological data, such as nucleotide or amino acid sequences in the form of genome assemblies, gene annotations, and multiple sequence alignments (Bayat, 2002). Diverse disciplines in the biological sciences rely on bioinformatic methods and software (Wren, 2016). Recently, researchers have acknowledged the need to consider diverse types of biological scientists with different levels of experience when developing software (Kumar and Dudley, 2007). It is also essential to implement high standards of software development that ensure software functionality and archival stability (Mangul, Mosqueiro, *et al.*, 2019; Mangul, Martin, *et al.*, 2019). For example, code quality can be improved by utilizing unit and integration tests, which ensure faithful function of code (Darriba *et al.*, 2018). As a result, the development of effective and user-friendly software for diverse biologists often requires an interdisciplinary team of software engineers, biologists, and others.

Even though numerous bioinformatic pieces of software are available, there are still several barriers to creating seamless and reproducible workflows (Kim *et al.*, 2018). This issue in part stems from different pieces of software requiring different input file formats, being unable to account for non-standard biological phenomena such as the use of alternative genetic codes, or can only be executed using web servers or graphical user interfaces, which cannot be incorporated into high-throughput pipelines. Another factor is that multiple pieces of software or custom scripts are typically needed to execute different steps in a larger bioinformatic pipeline; for example, bioinformatic workflows often rely on one software/script for converting file formats, another software/script for translating sequences using standard and non-standard genetic codes, another software/script to examine the properties of genomes or multiple sequence alignments, and so on. As a result, maintaining efficacious bioinformatic workflows is cumbersome (Kulkarni *et al.*, 2018). Thus, the bioinformatic community would benefit from a multi-purpose toolkit that contains diverse processing and analysis functions.

To address this need, we—an interdisciplinary team of software engineers, evolutionary biologists, molecular biologists, microbiologists, and others—developed BioKIT, a versatile toolkit with 40 functions, several of which were community sourced, that conduct routine and novel processing and analysis of diverse sequence files including genome assemblies, multiple sequence alignments, protein coding sequences, and sequencing data (Table 1). Functions implemented in BioKIT facilitate a wide variety of standard bioinformatic analyses, including genome assembly quality assessment (e.g., N50, L50, assembly size, guanine-cytosine (GC) content, number of scaffolds, and others), the calculation of multiple sequence alignment properties (i.e., number of taxa, alignment length, the number of constant sites, the number of parsimony-informative sites, and the number of variable sites), and processing and analysis of protein coding sequences (e.g., translation using 26 genetic codes including user-specified translation tables, GC content at the first, second, and third codon positions, and relative synonymous codon usage). To demonstrate the utility of BioKIT, we examined the genome assembly quality of 901 eukaryotic genomes, evaluated the properties of 10 phylogenomic data matrices, calculated relative synonymous codon usage in 171 fungal genomes, and estimated codon optimization in each gene from two *Saccharomyces* budding yeast species using a novel metric, gene-wise relative synonymous codon usage (gw-RSCU). BioKIT comes complete with common and novel functions that will help improve reproducibility and accessibility of diverse bioinformatic analysis and facilitate discovery in the biological sciences.

Materials and Methods

BioKIT is an easy-to-install command-line software that conducts diverse bioinformatic analyses in the UNIX programming environment. BioKIT is written in the Python programming language and has few dependencies, namely Biopython (Cock *et al.*, 2009) and numPy (Van Der Walt *et al.*, 2011).

BioKIT currently has 40 functions that process and analyze sequence files such as genome assemblies, multiple-sequence alignments, protein coding sequences, and sequencing data (Table 1). Processing

functions include those that convert various file formats, subset sequence reads from FASTQ files, rename entries in FASTA files, and others. Analysis functions include those that trim sequence reads in FASTQ files according to quality and length thresholds, calculate relative synonymous codon usage, estimate codon optimization, and others. Similar to other software we have developed (Steenwyk *et al.*, 2020, 2021; Steenwyk and Rokas, 2021), we plan on continuing to develop and incorporate additional functions into BioKIT to meet the needs of the research community.

Details about each function, their usage, tutorials, and other information such as how to request additional functions can be found in the online documentation (https://jlsteenwyk.com/BioKIT). To demonstrate the utility of BioKIT, we highlight four use-cases: (i) genome assembly quality assessment, (ii) summarizing properties of multiple sequence alignments, (iii) determination of relative synonymous codon usage using different genetic codes, and (iv) determination of a novel metric for estimation of gene-wise codon optimization, gene-wise relative synonymous codon usage (gw-RSCU).

Genome assembly quality assessment

Determination of genome assembly properties is essential when evaluating assembly quality (Gurevich *et al.*, 2013; Hunt *et al.*, 2013). To facilitate these analyses, the *genome_assembly_metrics* function in BioKIT calculates 14 diverse properties of genome assemblies that evaluate assembly quality and characteristics including:

• assembly size: sum length of all contigs/scaffolds;

• L50 (and L90): the number of contigs/scaffolds that make up 50% (or, in the case of L90, 90%) of the total length of the genome assembly;

• N50 (and N90): the length of the contig/scaffold which, along with all contigs/scaffolds longer than or equal to that contig/scaffold, contain 50% (or, in the case of N90, 90%) the length of a particular genome assembly;

• GC content: fraction of total bases that are either G or C;

• number of scaffolds: total number of contigs/scaffolds;

• number and sum length of large scaffolds: total number and sum length of contigs/scaffolds above 500 nucleotides in length (length threshold of a "large scaffold" can be modified by the user); and

• frequency of nucleotides: fraction of occurrences for adenine (A), thymine, (T), G, and C nucleotides.

Each metric can also be called using individual functions (e.g., the *n50* function calculates the N50 of an assembly and the *number_of_large_scaffolds* function calculates the number of large scaffolds in an assembly). We anticipate the ability of BioKIT to summarize genome assembly properties will be helpful for assessing genome quality as well as in comparative studies of genome properties, such as the evolution of genome size and GC content (Walker *et al.*, 2015; Shen *et al.*, 2020). Other pieces of software that conduct similar analyses include QUAST, REAPR, and GenomeQC (Gurevich *et al.*, 2013; Hunt *et al.*, 2013; Manchanda *et al.*, 2020).

Processing and assessing the properties of multiple sequence alignments

Multiple sequence alignments—the alignment of three or more biological sequences—contain a wealth of information. To facilitate easy use and manipulation of multiple sequence alignments, BioKIT implements 16 functions that process or analyze alignments including: generating consensus sequences; generating a position-specific score matrix (which represents the frequency of observing a particular amino acid or nucleotide at a specific position); recoding an alignment using different schemes, such as the RY-nucleotide scheme for nucleotide alignments (Woese *et al.*, 1991; Phillips *et al.*, 2001) or the Dayhoff-6, S&R-6, and KGB-6 schemes for amino acid alignments (Hrdy *et al.*, 2004; Embley *et al.*, 2003; Susko and Roger, 2007; Kosiol *et al.*, 2004); converting alignments among the following formats: FASTA, Clustal, MAF, Mauve, PHYLIP, PHYLIP-sequential, PHYLIP-relaxed, and Stockholm; extracting entries in FASTA files; removing entries from FASTA file; removing short sequences from a FASTA file; and others.

We highlight the *alignment_summary* function, which calculates numerous summary statistics for a multiple sequence alignment, a common step in many molecular evolutionary analyses (Plomion *et al.*, 2018; Winterton *et al.*, 2018). More specifically, the *alignment_summary* function calculates:

• alignment length: the total number of sites in an alignment;

• number of taxa: the total number of sequences in an alignment;

• number of parsimony-informative sites: a site in an alignment with at least two distinct nucleotides or amino acids that each occur at least twice;

• number of variable sites: a site in an alignment with at least two distinct nucleotides or amino acids;

• number of constant sites: sites with the same nucleotide or amino acid (excluding gaps); and

• the frequency of all character states: the fraction of occurrence for all nucleotides or amino acids (including gap characters represented as '-' or '?' in an alignment.

Like the *genome_assembly_metrics* function, each metric can be calculated individually (e.g., the *constant_sites* function calculates the number of constant sites in an alignment and the *character_frequency* function calculates the frequency of all character states). We anticipate the *alignment_summary* function will assist researchers in statistically evaluating the properties of their alignments. Other pieces of software that perform similar operations include AMAS (Borowiec, 2016) and Mesquite (Mesquite Project Team, 2014).

Examining features of coding sequences including relative synonymous codon usage

BioKIT contains multiple functions that process or analyze protein coding sequences including translating protein coding sequences into amino acids using one of 26 genetic codes or a user-specified translation table as well as determining the GC content at the first, second, and third codon positions.

Here, we highlight the *relative_synonymous_codon_usage* function, which calculates relative synonymous codon usage, the ratio of the observed frequency of synonymous codons to an expected

frequency in which all synonymous codons are used equally (Xu *et al.*, 2008). In this analysis, overrepresented codons have relative synonymous codon usage values greater than one whereas underrepresented codons have relative synonymous codon usage values less than one. Relative synonymous codon usage values of one fit the neutral expectation. The *relative_synonymous_codon_usage* function can be used with one of 26 genetic codes including user-specified translation tables. The ability of BioKIT to account for diverse genetic codes makes it uniquely suitable for analyses of lineages that contain multiple genetic codes (LaBella *et al.*, 2019; Krassowski *et al.*, 2018). Other software that conduct similar analyses include DAMBE and GCUA (Xia, 2013; McInerney, 1998).

We also highlight the *gene_wise_relative_synonymous_codon_usage* function, which calculates a novel metric, gw-RSCU, to examine biases in codon usage among individual genes encoded in a genome. More specifically, the gw-RSCU is calculated by determining the mean or median relative synonymous codon usage value for all codons in each gene based on their genome-wide values. Thus, BioKIT calculates relative synonymous codon usage for each codon based on codon usage in an entire set of protein coding genes, individually reexamines each gene and the relative synonymous codon usage value for each codon therein, and then determines the mean or median relative synonymous codon usage value for the individual gene. The formula for the mean gw-RSCU calculation is as follows:

$$gw\text{-}RSCU^{a} = \frac{\sum_{i=1}^{j} RSCU_i}{n}$$

where gw-RSCU^{a} is the gene that gw-RSCU is being calculated for, RSCU$_i$ is the relative synonmyouse codon usage value (calculated from all protein coding genes in a genome) for the *i*th codon of *j* codons in a gene, and *n* is the number of codons in a gene. To evaluate within-gene variation in relative synonymous codon usage, BioKIT also reports the standard deviation of relative synonymous codon usage values for each gene. Like the *relative_synonymous_codon_usage* function, gw-RSCU can be calculated using alternative genetic codes including user-specified ones. Taken together, these functions

can be used individually or in tandem to investigate diverse biological phenomena, including codon usage bias (LaBella *et al.*, 2019; Brandis and Hughes, 2016).

Implementing high standards of software development

Archival instability is a concern for bioinformatic tools and threatens the reproducibility of bioinformatic research. For example, in an analysis that aimed to evaluate the "installability" of bioinformatic software, 28% of over 36,000 bioinformatic tools failed to properly install due to implementation errors (Mangul, Mosqueiro, *et al.*, 2019). To ensure archival stability of BioKIT, we implemented a previously established protocol (Steenwyk and Rokas, 2021; Steenwyk *et al.*, 2020, 2021) for high standards of software development and design practices. More specifically, we wrote 327 unit and integration tests that ensure faithful functionality of BioKIT and span 95.46% of the codebase. We also implemented a continuous integration pipeline, which builds, packages, installs, and tests the functionality of BioKIT across Python versions 3.6, 3.7, 3.8, and 3.9. To accommodate diverse installation workflows, we also made BioKIT freely available under the MIT license across popular platforms including GitHub (https://github.com/JLSteenwyk/BioKIT), PyPi (https://pypi.org/project/biokit), and the Anaconda Cloud (https://anaconda.org/JLSteenwyk/biokit). To make BioKIT more user-friendly, we wrote online documentation, user tutorials, and instructions for requesting new features (https://jlsteenwyk.com/BioKIT). We anticipate our rigorous strategy to implement high standards of software development, coupled to our approach to facilitate easy software installation and extensive documentation, will address instabilities observed among many bioinformatic software and increase the long-term usability of BioKIT.

Results and Discussion

Genome assembly quality and characteristics among 901 eukaryotic genomes

To demonstrate the utility of BioKIT for the examination of genome assembly quality and characteristics, 14 diverse genome assembly metrics were determined among 901 scaffold-level haploid assemblies of

eukaryotic genomes, which were obtained from NCBI, and span three major classes of animals (Mammalia; N = 350), plants (Magnoliopsida; N = 336), and fungi (Eurotiomycetes; N = 215). Genome assembly properties exhibited variation both within and between the three classes (Figure 1). For example, fungi had the smallest average genome size of 32.71 ± 7.04 Megabases (Mbs) whereas mammals had the largest average genome size of 2,645.50 ± 487.48 Mbs. Extensive variation in genome size within each class corroborates previous findings of extreme genome size variation among eukaryotes (Elliott and Gregory, 2015). Variation in GC content, a genome property that has been actively investigated for decades (Romiguier *et al.*, 2010; Serres-Giardi *et al.*, 2012; Galtier *et al.*, 2001), was observed among the three eukaryotic classes—animals, plants, and fungi had an average GC content of 0.40 ± 0.04, 0.35 ± 0.04, and 0.49 ± 0.03, respectively. Lastly, there was wide variation in genome assembly metrics associated with continuity of assembly. For example, the average N50 values for animals, plants, and fungi were 12,287.64 ± 25,317.31 Mbs, 5,030.15 ± 19,358.58 Mbs, and 1,370.77 ± 1,552.13 Mbs, respectively. Taken together, these results demonstrate BioKIT can assist researchers in summarizing diverse genome assembly properties, which may be helpful not only for evaluating genome assembly quality, but also for studying genome evolution.

Properties of multiple sequence alignment from 10 phylogenomic studies

To demonstrate the utility of BioKIT in calculating summary statistics for multiple sequence alignments, we calculated six properties across 10 previously published phylogenomic data matrices of amino acid sequences (Borowiec *et al.*, 2015; Chen *et al.*, 2015; Misof *et al.*, 2014; Nagy *et al.*, 2014; Shen *et al.*, 2018; X.-X. Shen, Zhou, *et al.*, 2016; Steenwyk *et al.*, 2019; Struck *et al.*, 2015; Whelan *et al.*, 2015; Yang *et al.*, 2015) (Figure 2). Phylogenomic data matrices varied in the number of taxa (mean = 109.50 ± 87.26; median = 94; max = 343; min = 36). Alignment length is associated with greater phylogenetic accuracy and bipartition support (X.-X. Shen, Salichos, *et al.*, 2016); however, recent analyses suggest that in some instances shorter alignments that contain a wealth of informative sites (such as parsimony-informative sites) harbor robust phylogenetic signal (Steenwyk *et al.*, 2020). Interestingly, the longest

observed alignment (1,806,035 sites; *Chen, Vertebrates* in Figure 2) (Chen *et al.*, 2015) contained the highest number of constant sites (N = 610,994), which are phylogenetically uninformative, as well as the highest number of variable sites (N = 1,195,041), which are phylogenetically informative (X.-X. Shen, Salichos, *et al.*, 2016). In contrast to the multiple sequence alignment of vertebrate sequences, the second longest alignment of budding yeast sequences (1,162,805 sites; *Shen, 332 Yeast* in Figure 2) has few constant sites (N = 2,761) and many parsimony-informative (N = 1,152,145) and variable sites (N = 1,160,044). This observation may be driven in part by the rapid rate of budding yeast evolution compared to animals (Shen *et al.*, 2018). These results demonstrate BioKIT is useful in summarizing multiple sequence alignments.

Relative synonymous codon usage in 107 budding yeast and filamentous fungi

To demonstrate the utility of BioKIT in analyzing protein coding sequences, we calculated the relative synonymous codon usage of all codons in the protein coding sequences of 103 Eurotiomycetes (filamentous fungi) and 68 Saccharomycetes (budding yeasts) genomes obtained from the RefSeq database of NCBI (Figure 3). This example also demonstrates the flexibility of BioKIT to account for non-standard genetic codes, which are observed among some budding yeasts that use the CUG codon to encode a serine or alanine rather than a leucine (Krassowski *et al.*, 2018). Hierarchical clustering of relative synonymous codon usage values per codon (columns in Figure 3) revealed similar patterns across groups of codons. For example, CUA, AUA, and GUA—three of the four codons that end in UA—were underrepresented in all fungi. Hierarchical clustering of relative synonymous codon usage values per species (rows in Figure 3) revealed filamentous fungi and budding yeasts often clustered separately. For example, UGA, GUG, AAC, UAC, AAG, UUC, UCC, ACC, GCC, CGC, CUG, AUC, GUC, CUC, and GGC are more often overrepresented among filamentous fungi in comparison to budding yeasts; in contrast, UUG, GUU, CCA, and GGU are more often overrepresented among budding yeasts in comparison to filamentous fungi. Variation within each lineage was also observed; for example, UUA was underrepresented in most, but not all, budding yeasts.

Patterns of gene-wise codon usage bias can be used to assess codon optimization and predict steady-state gene expression levels

To evaluate the utility of BioKIT in examining gene-wise codon usage biases, we calculated the mean and median gw-RSCU value, a novel metric introduced in the present manuscript, for individual protein coding genes in the genome of *S. cerevisiae* (Figure 4A). Mean and median gw-RSCU values were often, but not always, similar—the average absolute difference between mean and median gw-RSCU is 0.05 ± 0.04. In *S. cerevisiae*, as well as other organisms, genes encoding ribosomal components and histones are known to be codon optimized and highly expressed (Hershberg and Petrov, 2009; LaBella *et al.*, 2021; Sharp *et al.*, 1986). Therefore, we hypothesized that genes with high gw-RSCU values will have functions related to ribosomes or histones because patterns of gene-wise codon usage bias may be indicative of codon optimization. Supporting this hypothesis, examination of the 10 genes with the highest mean gw-RSCU revealed five genes with ribosome-related functions [RPL41B (YDL133C-A), mean gw-RSCU: 1.60; RPL41A (YDL184C), mean gw-RSCU: 1.58; RPS14A (YCR031C), mean gw-RSCU: 1.44; RPS9B (YBR189W), mean gw-RSCU: 1.43; and RPL18A (YOL120C), mean gw-RSCU: 1.43] and four genes with histone-related functions [HHF1 (YBR009C), mean gw-RSCU: 1.45; HTA2 (YBL003C), mean gw-RSCU: 1.44; HHF2 (YNL030W), mean gw-RSCU: 1.43; and HTA1 (YDR225W), mean gw-RSCU: 1.43]. Examination of the 10 most optimized genes according to median gw-RSCU revealed similar observations wherein nine genes had ribosome-related functions [RPS14A (YCR031C), median gw-RSCU: 1.48; RPS12 (YOR369C), median gw-RSCU: 1.40; RPS30B (YOR182C), median gw-RSCU: 1.40; RPP2A (YOL039W), median gw-RSCU: 1.40; RPL18A (YOL120C), median gw-RSCU; RPS3 (YNL178W), median gw-RSCU: 1.40; RPL13B (YMR142C), median gw-RSCU: 1.40; RPP0 (YLR340W), median gw-RSCU: 1.40; and RPS0B (YLR048W), median gw-RSCU: 1.40]. More broadly, genes associated with the 60S and 40S ribosomal units (gold color in Figure 4A) tended to have high gw-RSCU values. These results suggest gw-RSCU values may be useful for estimating codon optimization.

To further explore the relationship between gw-RSCU and codon optimization, we compared gw-RSCU values to the values of the tRNA adaptation index, a measure of codon optimization (Sabi and Tuller, 2014), in *S. cerevisiae* as well as in steady state gene expression data from *Saccharomyces mikatae* (LaBella *et al.*, 2019). In *S. cerevisiae*, strong correlation was observed between mean gw-RSCU and tRNA adaptation index values (Figure 4B) and a less robust, but still significant, correlation was observed between median gw-RSCU and tRNA adaptation index values (Figure 4C). Examination of gw-RSCU and gene expression data from *S. mikatae* revealed a robust correlation (Figure 4E and 4F) suggesting gw-RSCU, and in particular the mean gw-RSCU, can serve as a measure of gene-wise codon optimization.

Conclusion

BioKIT is a multi-purpose toolkit that has diverse applications for bioinformatics research. The utilities implemented in BioKIT aim to facilitate the execution of seamless bioinformatic workflows that handle diverse sequence file types. Implementation of state-of-the-art software development and design principles in BioKIT help ensure faithful function and archival stability. BioKIT will be helpful for bioinformaticians with varying levels of expertise and biologists from diverse disciplines including molecular biology.

Data Availability

BioKIT is freely available under the MIT license from GitHub (https://github.com/JLSteenwyk/BioKIT), PyPi (https://pypi.org/project/biokit), and the Anaconda Cloud (https://anaconda.org/JLSteenwyk/biokit). Documentation, user tutorials, and instructions for requesting new features are available online (https://jlsteenwyk.com/BioKIT).

Acknowledgements

We thank the Rokas lab for helpful discussion and feedback. We also thank the bioinformatic community for providing suggestions of functions to implement in BioKIT. J.L.S. and A.R. were funded by the

Howard Hughes Medical Institute through the James H. Gilliam Fellowships for Advanced Study program. Research in A.R.'s lab is supported by grants from the National Science Foundation (DEB-1442113 and DEB-2110404), the National Institutes of Health/National Institute of Allergy and Infectious Diseases (R56 AI146096), and the Burroughs Wellcome Fund.

Conflict of Interest

A.R. is a scientific consultant for LifeMine Therapeutics, Inc.

References

Bayat,A. (2002) Science, medicine, and the future: Bioinformatics. *BMJ*, **324**, 1018–1022.

Bentele,K. *et al.* (2013) Efficient translation initiation dictates codon usage at gene start. *Mol. Syst. Biol.*, **9**, 675.

Bolger,A.M. *et al.* (2014) Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics*, **30**, 2114–2120.

Borowiec,M.L. (2016) AMAS: a fast tool for alignment manipulation and computing of summary statistics. *PeerJ*, **4**, e1660.

Borowiec,M.L. *et al.* (2015) Extracting phylogenetic signal and accounting for bias in whole-genome data sets supports the Ctenophora as sister to remaining Metazoa. *BMC Genomics*, **16**, 987.

Brandis,G. and Hughes,D. (2016) The Selective Advantage of Synonymous Codon Usage Bias in Salmonella. *PLOS Genet.*, **12**, e1005926.

Britten,R.J. (1998) Precise sequence complementarity between yeast chromosome ends and two classes of just-subtelomeric sequences. *Proc. Natl. Acad. Sci.*, **95**, 5906–5912.

Camacho,C. *et al.* (2009) BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Chanumolu,S.K. *et al.* (2019) FQStat: a parallel architecture for very high-speed assessment of sequencing quality metrics. *BMC Bioinformatics*, **20**, 424.

Chen,M.-Y. *et al.* (2015) Selecting Question-Specific Genes to Reduce Incongruence in Phylogenomics: A Case Study of Jawed Vertebrate Backbone Phylogeny. *Syst. Biol.*, **64**, 1104–1120.

Cock,P.J.A. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Darriba,D. *et al.* (2018) The State of Software for Evolutionary Biology. *Mol. Biol. Evol.*, **35**, 1037–1046.

Elliott,T.A. and Gregory,T.R. (2015) What's in a genome? The C-value enigma and the evolution of eukaryotic genome content. *Philos. Trans. R. Soc. B Biol. Sci.*, **370**, 20140331.

Embley,M. *et al.* (2003) Mitochondria and hydrogenosomes are two forms of the same fundamental organelle. *Philos. Trans. R. Soc. London. Ser. B Biol. Sci.*, **358**, 191–203.

Galtier,N. *et al.* (2001) GC-Content Evolution in Mammalian Genomes: The Biased Gene Conversion Hypothesis. *Genetics*, **159**, 907–911.

Gao,F. and Zhang,C.-T. (2006) GC-Profile: a web-based tool for visualizing and analyzing the variation of GC content in genomic sequences. *Nucleic Acids Res.*, **34**, W686–W691.

Glez-Pena,D. *et al.* (2010) ALTER: program-oriented conversion of DNA and protein alignments. *Nucleic Acids Res*, **38**, W14-8.

Gribskov,M. *et al.* (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci.*, **84**, 4355–4358.

Gurevich,A. *et al.* (2013) QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**, 1072–1075.

Hernandez,A.M. and Ryan,J.F. (2021) Six-State Amino Acid Recoding is not an Effective Strategy to Offset Compositional Heterogeneity and Saturation in Phylogenetic Analyses. *Syst. Biol.*

Hershberg,R. and Petrov,D.A. (2009) General Rules for Optimal Codon Choice. *PLoS Genet.*, **5**, e1000556.

Hrdy,I. *et al.* (2004) Trichomonas hydrogenosomes contain the NADH dehydrogenase module of mitochondrial complex I. *Nature*, **432**, 618–622.

Hunt,M. *et al.* (2013) REAPR: a universal tool for genome assembly evaluation. *Genome Biol.*, **14**, R47.

Kim,Y.-M. *et al.* (2018) Experimenting with reproducibility: a case study of robustness in bioinformatics. *Gigascience*, **7**.

Kolde,R. (2012) Package `pheatmap'. *Bioconductor*, 1–6.

Kosiol,C. *et al.* (2004) A new criterion and method for amino acid classification. *J. Theor. Biol.*, **228**, 97–106.

Krassowski,T. *et al.* (2018) Evolutionary instability of CUG-Leu in the genetic code of budding yeasts. *Nat. Commun.*, **9**, 1887.

Kulkarni,N. *et al.* (2018) Reproducible bioinformatics project: a community for reproducible bioinformatics analysis pipelines. *BMC Bioinformatics*, **19**, 349.

Kumar,S. *et al.* (2016) MEGA7: Molecular Evolutionary Genetics Analysis Version 7.0 for Bigger Datasets. *Mol. Biol. Evol.*

Kumar,S. and Dudley,J. (2007) Bioinformatics software for biologists in the genomics era. *Bioinformatics*, **23**, 1713–1717.

LaBella,A.L. *et al.* (2021) Signatures of optimal codon usage in metabolic genes inform budding yeast ecology. *PLOS Biol.*, **19**, e3001185.

LaBella,A.L. *et al.* (2019) Variation and selection on codon usage bias across an entire subphylum. *PLOS Genet.*, **15**, e1008304.

Li,H. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Manchanda,N. *et al.* (2020) GenomeQC: a quality assessment tool for genome assemblies and gene structure annotations. *BMC Genomics*, **21**, 193.

Mangul,S., Mosqueiro,T., *et al.* (2019) Challenges and recommendations to improve the installability and archival stability of omics computational tools. *PLOS Biol.*, **17**, e3000333.

Mangul,S., Martin,L.S., *et al.* (2019) Improving the usability and archival stability of bioinformatics software. *Genome Biol.*, **20**, 47.

McInerney,J.O. (1998) GCUA: general codon usage analysis. *Bioinformatics*, **14**, 372–373.

Mesquite Project Team (2014) Mesquite: A modular system for evolutionary analysis. *Available from http// mesquiteproject.wikispaces.com/home*.

Minh,B.Q. *et al.* (2020) IQ-TREE 2: New Models and Efficient Methods for Phylogenetic Inference in the Genomic Era. *Mol. Biol. Evol.*, **37**, 1530–1534.

Misof,B. *et al.* (2014) Phylogenomics resolves the timing and pattern of insect evolution. *Science (80-.).*, **346**, 763–767.

Nagy,L.G. *et al.* (2014) Latent homology and convergent regulatory evolution underlies the repeated emergence of yeasts. *Nat. Commun.*, **5**, 4471.

Ou,S. *et al.* (2020) Effect of sequence depth and length in long-read assembly of the maize inbred NC358. *Nat. Commun.*, **11**, 2288.

Phillips,M.J. *et al.* (2001) Mitochondrial genomes of a bandicoot and a brushtail possum confirm the monophyly of australidelphian marsupials. *Proc. R. Soc. London. Ser. B Biol. Sci.*, **268**, 1533–1538.

Plomion,C. *et al.* (2018) Oak genome reveals facets of long lifespan. *Nat. Plants*, **4**, 440–452.

Rice,P. *et al.* (2000) EMBOSS: The European Molecular Biology Open Software Suite. *Trends Genet.*, **16**, 276–277.

Romiguier,J. *et al.* (2010) Contrasting GC-content dynamics across 33 mammalian genomes: Relationship with life-history traits and chromosome sizes. *Genome Res.*, **20**, 1001–1009.

Sabi,R. and Tuller,T. (2014) Modelling the Efficiency of Codon–tRNA Interactions Based on Codon Usage Bias. *DNA Res.*, **21**, 511–526.

Serres-Giardi,L. *et al.* (2012) Patterns and Evolution of Nucleotide Landscapes in Seed Plants. *Plant Cell*, **24**, 1379–1397.

Sharp,P.M. *et al.* (1986) Codon usage in yeast: cluster analysis clearly differentiates highly and lowly expressed genes. *Nucleic Acids Res.*, **14**, 5125–5143.

Shen,W. *et al.* (2016) SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation. *PLoS One*, **11**, e0163962.

Shen,X.-X., Salichos,L., *et al.* (2016) A Genome-Scale Investigation of How Sequence, Function, and Tree-Based Gene Properties Influence Phylogenetic Inference. *Genome Biol. Evol.*, **8**, 2565–2580.

Shen,X.-X. *et al.* (2020) Genome-scale phylogeny and contrasting modes of genome evolution in the fungal phylum Ascomycota. *Sci. Adv.*, **6**, eabd0079.

Shen,X.-X., Zhou,X., *et al.* (2016) Reconstructing the Backbone of the Saccharomycotina Yeast Phylogeny Using Genome-Scale Data. *G3 Genes|Genomes|Genetics*, **6**, 3927–3939.

Shen,X.-X. *et al.* (2018) Tempo and Mode of Genome Evolution in the Budding Yeast Subphylum. *Cell*, **175**, 1533-1545.e20.

Steenwyk,J.L. *et al.* (2019) A Robust Phylogenomic Time Tree for Biotechnologically and Medically Important Fungi in the Genera Aspergillus and Penicillium. *MBio*, **10**.

Steenwyk,J.L. *et al.* (2020) ClipKIT: A multiple sequence alignment trimming software for accurate

phylogenomic inference. *PLOS Biol.*, **18**, e3001007.

Steenwyk,J.L. *et al.* (2021) PhyKIT: a broadly applicable UNIX shell toolkit for processing and analyzing phylogenomic data. *Bioinformatics*.

Steenwyk,J.L. and Rokas,A. (2021) orthofisher: a broadly applicable tool for automated gene identification and retrieval. *G3 Genes|Genomes|Genetics*, **11**.

Sternke,M. *et al.* (2019) Consensus sequence design as a general strategy to create hyperstable, biologically active proteins. *Proc. Natl. Acad. Sci.*, **116**, 11275–11284.

Struck,T.H. *et al.* (2015) The Evolution of Annelids Reveals Two Adaptive Routes to the Interstitial Realm. *Curr. Biol.*, **25**, 1993–1999.

Susko,E. and Roger,A.J. (2007) On Reduced Amino Acid Alphabets for Phylogenetic Inference. *Mol. Biol. Evol.*, **24**, 2139–2150.

Walker,P.J. *et al.* (2015) Evolution of Genome Size and Complexity in the Rhabdoviridae. *PLOS Pathog.*, **11**, e1004664.

Van Der Walt,S. *et al.* (2011) The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.*, **13**, 22–30.

Whelan,N. V. *et al.* (2015) Error, signal, and the placement of Ctenophora sister to all other animals. *Proc. Natl. Acad. Sci.*, **112**, 5773–5778.

Wickham,H. (2009) ggplot2 Springer New York, New York, NY.

Winterton,S.L. *et al.* (2018) Evolution of lacewings and allied orders using anchored phylogenomics (Neuroptera, Megaloptera, Raphidioptera). *Syst. Entomol.*, **43**, 330–354.

Woese,C.R. *et al.* (1991) Archaeal Phylogeny: Reexamination of the Phylogenetic Position of Archaeoglohus fulgidus in Light of Certain Composition-induced Artifacts. *Syst. Appl. Microbiol.*, **14**, 364–371.

Wren,J.D. (2016) Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades. *Bioinformatics*, **32**, 2686–2691.

Xia,X. (2013) DAMBE5: A Comprehensive Software Package for Data Analysis in Molecular Biology

and Evolution. *Mol. Biol. Evol.*, **30**, 1720–1728.

Xu,X. *et al.* (2008) Analysis of synonymous codon usage and evolution of begomoviruses. *J. Zhejiang Univ. Sci. B*, **9**, 667–674.

Yang,Y. *et al.* (2015) Dissecting Molecular Evolution in the Highly Diverse Plant Clade Caryophyllales Using Transcriptome Sequencing. *Mol. Biol. Evol.*, **32**, 2001–2014.

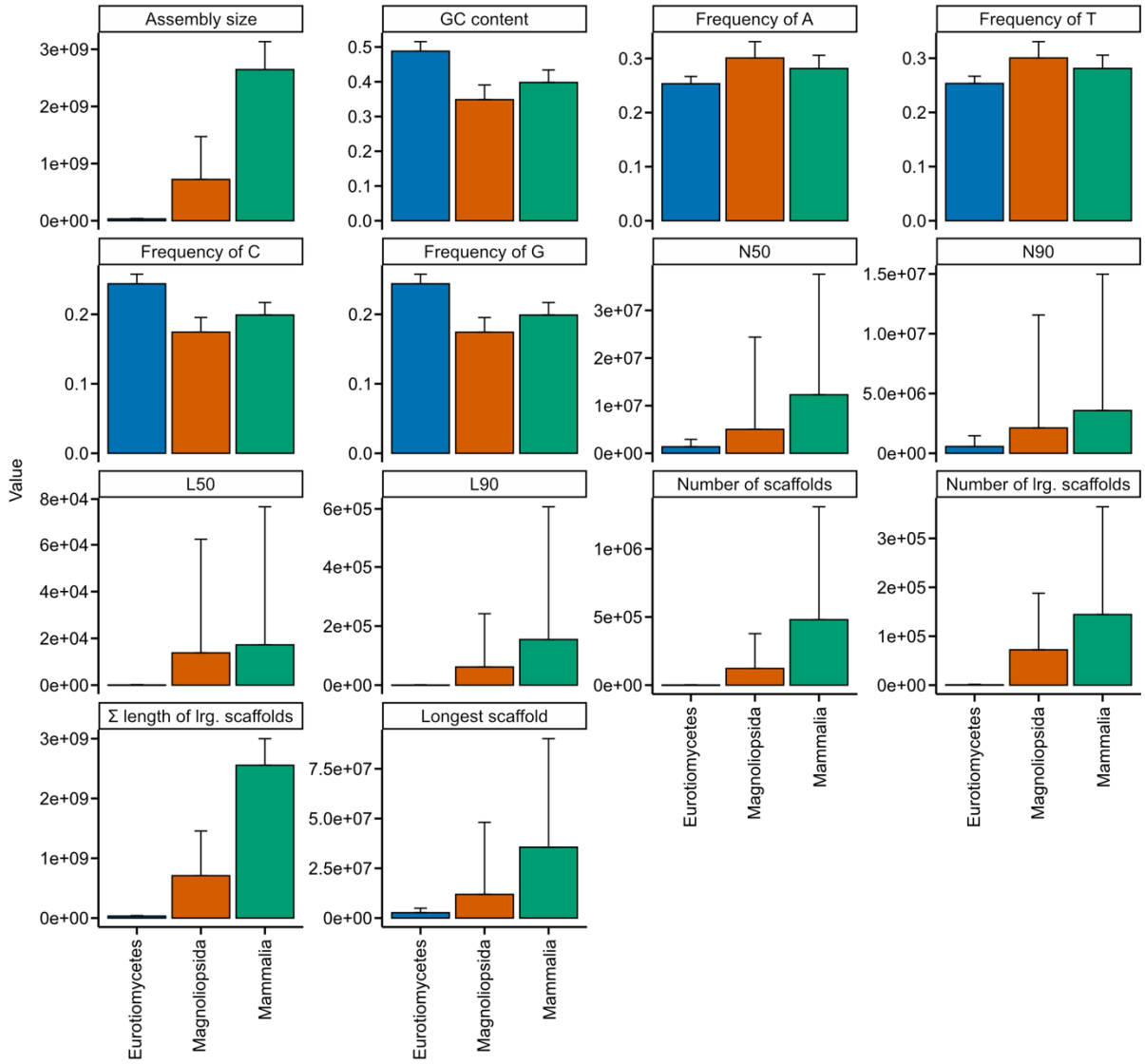

Figure 1. Summary of genome assembly metrics across 901 genomes from three eukaryotic classes. Nine hundred and one scaffold-level genome assemblies from three major eukaryotic classes (215 Eurotiomycetes (kingdom: Fungi), 336 Magnoliopsida (kingdom: Plantae), 350 Mammalia (kingdom: Animalia)) were obtained from NCBI and examined for diverse metrics including assembly size, GC content, frequency of A, T, C, and G, N50, N90, L50, L90, number of scaffolds, number of large scaffolds (defined as being greater than 500 nucleotides, which can be modified by the user), sum length of large scaffolds, and longest scaffold in the assembly. Bar plots represent the mean for each taxonomic

class. Error bars represent the standard deviation of values. This figure was made using ggplot2 (Wickham, 2009) and ggpubfigs (https://github.com/JLSteenwyk/ggpubfigs).

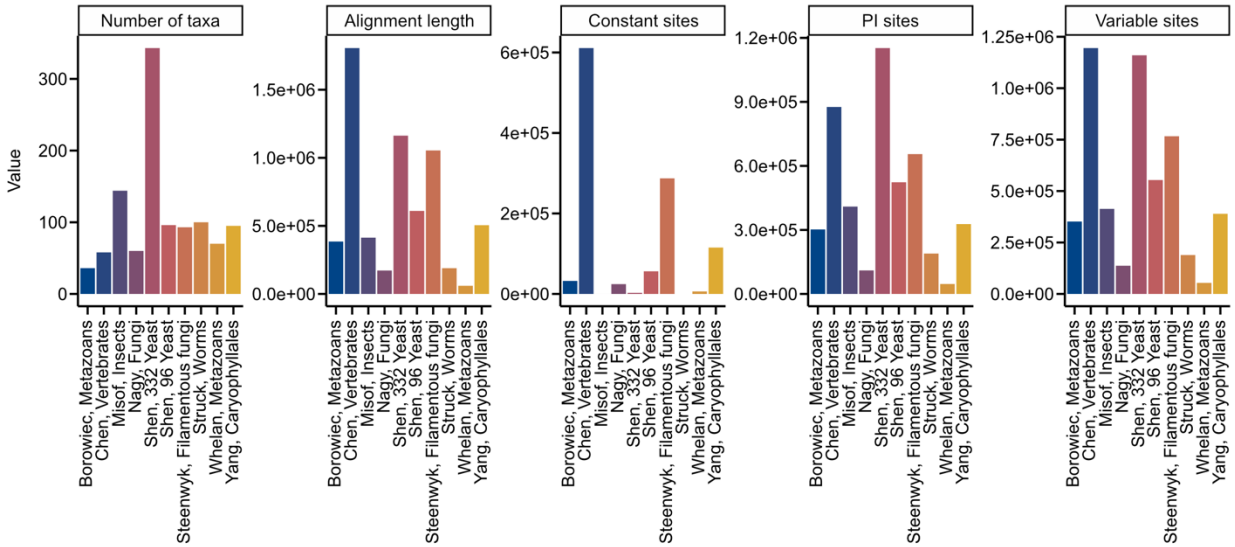

Figure 2. Summary metrics among multiple sequence alignments from phylogenomic studies. Multiple sequence alignments of amino acid sequences from ten phylogenomic data matrices (Borowiec *et al.*, 2015; Chen *et al.*, 2015; Misof *et al.*, 2014; Nagy *et al.*, 2014; Shen *et al.*, 2018; X.-X. Shen, Zhou, *et al.*, 2016; Steenwyk *et al.*, 2019; Struck *et al.*, 2015; Whelan *et al.*, 2015; Yang *et al.*, 2015) were examined for five metrics: number of taxa, alignment length, number of constant sites, number of parsimony-informative sites, and number of variable sites. The x-axis depicts the last name of the first author of the phylogenomic study followed by a description of the organisms that were under study. The abbreviation PI represents parsimony-informative sites. Although excluded here for simplicity and clarity, BioKIT also determines character state frequency (nucleotide or amino acid) when summarizing alignment metrics. This figure was made using ggplot2 (Wickham, 2009) and ggpubfigs (https://github.com/JLSteenwyk/ggpubfigs).

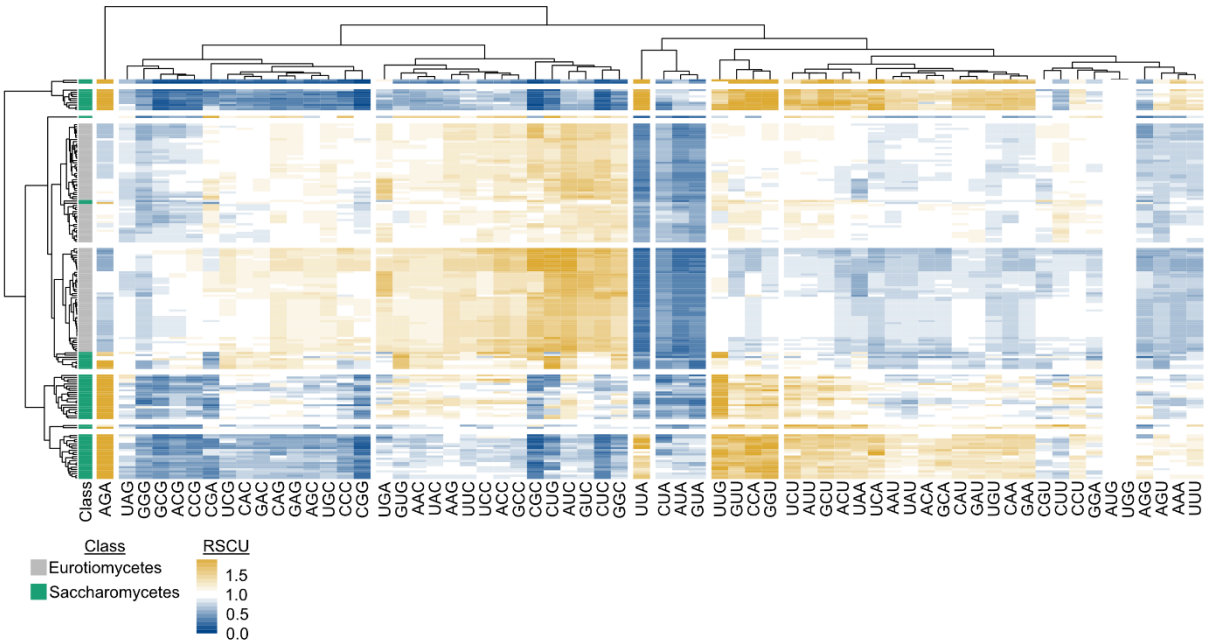

Figure 3. Relative synonymous codon usage across 171 fungal genomes. Relative synonymous codon usage (RSCU) was calculated from the coding sequences of 103 Eurotiomycetes (filamentous fungi) and 68 Saccharomycetes (budding yeasts) genomes obtained from NCBI. Hierarchical clustering was conducted across the fungal species (rows) and codons (columns). Eight groups of clustered rows were identified; seven groups of clustered columns were identified. Broad differences were observed in the RSCU values of Eurotiomycetes and Saccharomycetes genomes. For example, Saccharomycetes tended to have higher RSCU values for the AGA codon, whereas Eurotiomycetes tended to have higher RSCU values for the CUG codon. To account for the use of an alternative genetic code in budding yeast genomes from the CUG-Ser1 and CUG-Ser2 lineages (Krassowski *et al.*, 2018), the alternative yeast nuclear code—which is one of 26 alternative genetic codes incorporated into BioKIT—was used during RSCU determination. User's may also provide their own genetic code if it is unavailable in BioKIT. Overrepresented codons (RSCU>1) are depicted in a gold gradient; underrepresented codons (RSCU<1) are depicted in a blue gradient. RSCU values greater than 2 are depicted with the maximum gold color. Eurotiomycetes are depicted in grey; Saccharomycetes are depicted in green. This figure was made using pheatmap (Kolde, 2012).

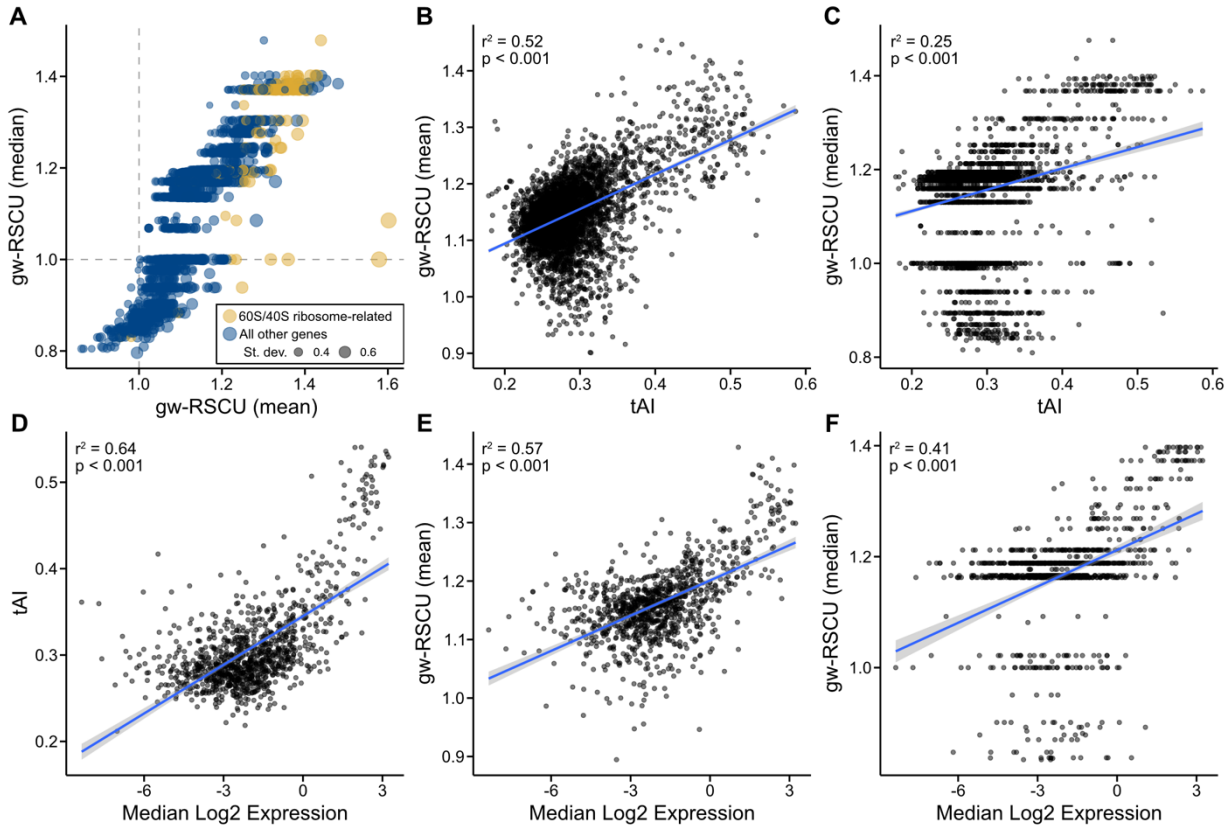

Figure 4. Mean gene-wise relative synonymous codon usage accurately estimates codon optimization. (A) Gene-wise relative synonymous codon usage (gw-RSCU), the mean (x-axis) or median (y-axis) relative synonymous codon usage value per gene (based on RSCU values calculated from the entire set of protein coding genes), was calculated from the coding sequences of the model budding yeast *Saccharomyces cerevisiae*. (B, C) In *S. cerevisiae*, a significant correlation was observed between tRNA adaptation index (tAI), a well-known measure of codon optimization (Sabi and Tuller, 2014), and mean as well as median gw-RSCU ($r^2 = 0.52$, $p < 0.001$ and $r^2 = 0.25$, $p < 0.001$, respectively; Pearson's Correlation Coefficient). (D) Using previously published data, a correlation is observed between median log2 gene expression and tAI in *Saccharomyces mikatae* (LaBella *et al.*, 2019), which is evidence of tAI values being indicative of codon optimization. Comparison of mean and median gw-RSCU (E and F, respectively) and median log2 gene expression revealed similarly strong correlations ($r^2 = 0.57$, $p < 0.001$ and $r^2 = 0.41$, $p < 0.001$, respectively; Pearson's Correlation Coefficient). Of note, mean gw-RSCU had a strong correlation to gene expression than median gw-RSCU. Each gene is represented by a dot. In panel

A, the size of each dot represents the standard deviation of RSCU values observed in the gene and the color of each dot represents if the protein encoded by the gene has functions related to the 60S and 40S ribosomal subunits (gold) or a different function (blue).

Table 1. Summary of 40 functions in BioKIT

Function name	Description	Type of function	Input data	Citation	Example software that performs this function
alignment_length	Calculate alignment length	Analysis	Multiple-sequence file in FASTA format	NA	AMAS (Borowiec, 2016)
alignment_recoding	Recode alignments using reduced character states	Processing	Multiple-sequence file in FASTA format	(Woese *et al.*, 1991; Hrdy *et al.*, 2004; Embley *et al.*, 2003; Susko and Roger, 2007; Kosiol *et al.*, 2004)	Custom scripts (Hernandez and Ryan, 2021)
alignment_summary	Summarize diverse properties of a multiple sequence alignment	Analysis	Multiple-sequence file in FASTA format	NA	AMAS (Borowiec, 2016); custom scripts (X.-X. Shen, Salichos, *et al.*, 2016); PhyKIT (Steenwyk *et al.*, 2021)
consensus_sequence	Generates a consensus sequence	Analysis	Multiple-sequence file in FASTA format	(Sternke *et al.*, 2019)	Geneious (https://www.geneious.com)
constant_sites	Determine the number of constant sites in an alignment	Analysis	Multiple-sequence file in FASTA format	(Kumar *et al.*, 2016)	IQ-TREE (Minh *et al.*, 2020)
parsimony_informative_sites	Determine the number of parsimony-informative sites in an alignment	Analysis	Multiple-sequence file in FASTA format	(Kumar *et al.*, 2016)	AMAS (Borowiec, 2016); custom scripts (X.-X. Shen, Salichos, *et al.*, 2016)
position_specific_score_matrix	Generates a position specific score matrix for an alignment	Analysis	Multiple-sequence file in FASTA format	(Gribskov *et al.*, 1987)	BLAST+ (Camacho *et al.*, 2009)
variable_sites	Determine the number of variable sites in an alignment	Analysis	Multiple-sequence file in FASTA format	(X.-X. Shen, Salichos, *et al.*, 2016)	AMAS (Borowiec, 2016); custom scripts (X.-X. Shen, Salichos, *et al.*, 2016); PhyKIT (Steenwyk *et al.*, 2021)
gc_content_first_position	Determine the GC content of the first codon position among protein coding sequences	Analysis	Protein coding sequences in FASTA format	(Bentele *et al.*, 2013)	Custom scripts (Bentele *et al.*, 2013)
gc_content_second_position	Determine the GC content of the second codon position among protein coding sequences	Analysis	Protein coding sequences in FASTA format	(Bentele *et al.*, 2013)	Custom scripts (Bentele *et al.*, 2013)

gc_content_third_position	Determine the GC content of the third codon position among protein coding sequences	Analysis	Protein coding sequences in FASTA format	(Bentele *et al.*, 2013)	Custom scripts (Bentele *et al.*, 2013)
gene_wise_relative_synonymous_codon_usage	Calculate gene-wise relative synonymous codon usage	Analysis	Protein coding sequences in FASTA format	This study	This study
relative_synonymous_codon_usage	Calculate relative synonymous codon usage	Analysis	Protein coding sequences in FASTA format	(Xu *et al.*, 2008)	MEGA (Kumar *et al.*, 2016)
translate_sequence	Translate protein coding sequences to amino acid sequences	Processing	Protein coding sequences in FASTA format	NA	EMBOSS (Rice *et al.*, 2000)
fastq_read_lengths	Examine the distribution of read lengths	Analysis	Sequence reads in FASTQ format	NA	FQStat (Chanumolu *et al.*, 2019)
subset_pe_fastq_reads	Down sample paired-end reads	Processing	Sequence reads in FASTQ format	NA	SeqKit (W. Shen *et al.*, 2016)
subset_se_fastq_reads	Down sample single-end reads	Processing	Sequence reads in FASTQ format	NA	SeqKit (W. Shen *et al.*, 2016)
trim_pe_fastq_reads	Trim paired-end reads based on quality and length thresholds	Analysis	Sequence reads in FASTQ format	NA	Trimmomatic (Bolger *et al.*, 2014)
trim_se_fastq_reads	Trim single-end reads based on quality and length thresholds	Analysis	Sequence reads in FASTQ format	NA	Trimmomatic (Bolger *et al.*, 2014)
gc_content	Determine GC content	Analysis	FASTA file of nucleotide sequences	(Romiguier *et al.*, 2010)	custom scripts (X.-X. Shen, Salichos, *et al.*, 2016); GC-Profile (Gao and Zhang, 2006)
genome_assembly_metrics	Determine diverse properties of a genome assembly for quality assessment and characterization	Analysis	FASTA file of a genome assembly	(Gurevich *et al.*, 2013)	QUAST (Gurevich *et al.*, 2013); REAPR (Hunt *et al.*, 2013)
l50	L50	Analysis	FASTA file of a genome assembly	(Gurevich *et al.*, 2013)	QUAST (Gurevich *et al.*, 2013)

l90	L90	Analysis	FASTA file of a genome assembly	(Gurevich *et al.*, 2013)	QUAST (Gurevich *et al.*, 2013)
longest_scaffold	Determine the length of the longest entry in a FASTA file	Analysis	FASTA file	(Gurevich *et al.*, 2013)	Custom scripts (Ou *et al.*, 2020)
n50	N50	Analysis	FASTA file of a genome assembly	(Gurevich *et al.*, 2013)	QUAST (Gurevich *et al.*, 2013)
n90	N90	Analysis	FASTA file of a genome assembly	(Gurevich *et al.*, 2013)	QUAST (Gurevich *et al.*, 2013)
number_of_large_scaffolds	Determine the number and length of scaffolds longer than 500 nucleotides. Length threshold of 500 nucleotides can be modified by the user	Analysis	FASTA file	NA	QUAST (Gurevich *et al.*, 2013)
number_of_scaffolds	Determine the number of FASTA entries	Analysis	FASTA file	NA	QUAST (Gurevich *et al.*, 2013)
sum_of_scaffold_lengths	Determine the total length of all FASTA entries	Analysis	FASTA file	NA	QUAST (Gurevich *et al.*, 2013)
character_frequency	Determine the frequency of each character. Gaps are assumed to be represented as '?' and '-' characters	Analysis	FASTA file	NA	Biostrings (https://rdrr.io/bioc/Biostrings/)
faidx	Get sequence entry from FASTA file	Processing	FASTA file	NA	SAMtools (Li *et al.*, 2009)
file_format_converter	Converts multiple sequence alignments from one format to another	Processing	FASTA, Clustal, MAF, Mauve, Phylip, Phylip-sequential, Phylip-relaxed, and Stockholm	NA	ALTER (Glez-Pena *et al.*, 2010)
multiple_line_to_single_line_fasta	Reformat sequences to be represented on one line	Processing	FASTA file	NA	FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/)

remove_fasta_entry	Remove sequence based on entry identifier	Processing	FASTA file	NA	NA
remove_short_sequences	Remove short sequences	Processing	FASTA file	NA	NA
rename_fasta_entries	Rename FASTA entries	Processing	FASTA file	NA	FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/)
reorder_by_sequence_length	Reorder FASTA entries by length	Processing	FASTA file	NA	SeqKit (W. Shen *et al.*, 2016)
sequence_complement	Generate sequence complements in the forward or reverse direction	Processing	FASTA file	(Britten, 1998)	EMBOSS (Rice *et al.*, 2000)
sequence_length	Calculate the length of each FASTA file	Analysis	FASTA file	NA	bioawk (https://github.com/lh3/bioawk)
single_line_to_multiple_line_fasta	Reformat sequences to be represented on multiple lines	Processing	FASTA file	NA	FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/)