PAQman: reference-free ensemble evaluation of long-read eukaryotic genome assemblies

Samuel O'Donnell[1]*, Ningxiao Li[2,4], Jacob L. Steenwyk[3], David Geiser[4], Frank N. Martin[2], Emile Gluck-Thaler[1,5]*

1: Department of Plant Pathology, University of Wisconsin-Madison, Madison, WI 53706, USA
2: Crop Improvement and Protection Research Unit, United States Department of Agriculture – Agricultural Research Service, Salinas, CA 93905, USA
3: Howard Hughes Medical Institute and the Department of Molecular and Cell Biology, University of California, Berkeley, CA, USA
4: Department of Plant Pathology and Environmental Microbiology, The Pennsylvania State University, University Park, PA 16802, USA
5: Wisconsin Institute for Discovery, Madison, WI 53706, USA.

*Co-corresponding authors. Correspondence should be addressed to:
samuel.a.odonnell@gmail.com and gluckthaler@wisc.edu

ORCiD IDs:
S. O'Donnell: 0000-0003-2447-1993
N. Li: 0000-0002-7847-3567
J.L. Steenwyk: 0000-0002-8436-595X
D. Geiser: 0000-0002-1590-2045
F.N. Martin: 0000-0002-8050-1248
E. Gluck-Thaler: 0000-0003-0438-7495

Running title:
PAQman: quality assessment of long-read genomes

Keywords:
Assembly quality; Assembly evaluation; Oxford Nanopore Technologies, Pacific Biosciences

Abstract

Advances in long-read sequencing and assembly algorithms have made it easier and more cost effective to generate high-quality genome assemblies. However, assessing assembly quality remains challenging, as existing tools frequently rely on few metrics and/or require a reference assembly for comparison. To address this, we developed the Post-Assembly Quality manager (PAQman), a tool that evaluates seven reference-free features of genome quality: Contiguity, Gene content, Completeness, Accuracy, Correctness, Coverage, and Telomerality. PAQman integrates multiple commonly used tools alongside custom scripts, requiring users to provide only the genome assembly and its underlying long-read data, while providing a streamlined and consistent framework for quality assessment across datasets and organisms.

Summary

PAQman is an ensemble-based tool for the comprehensive and reference-free evaluation of eukaryotic genome assemblies derived from long-read sequencing data. The simultaneous integration of seven quality features allows users to easily assess assembly quality using a standardized and reproducible framework.

Availability and implementation

PAQman is distributed as a conda environment and an Apptainer image. Source code and documentation are freely available through the GitHub repository at https://github.com/samtobam/paqman and a Zenodo archive at https://doi.org/10.5281/zenodo.16039705 for v1.1.0 used in this study.

Introduction

The ease with which researchers can generate reference-quality assemblies has advanced rapidly with the continued improvement of long-read sequencing technologies, namely those offered by Oxford Nanopore Technologies (ONT) and Pacific Biosciences. In parallel, assembly algorithms and pipelines have significantly improved in both accuracy and speed (Li 2018; Kolmogorov et al. 2019; Giani et al. 2020; Espinosa et al. 2024). Despite these advancements, assembly quality is often assessed by disparate software that each use a subset of metrics, leading to inefficiencies in throughput and evaluation. Often these different statistics each evaluate distinct genomic features and thus do not always correlate with one another (Zhang et al. 2023). Determining the most appropriate metrics for quality assessment is therefore challenging, especially when benchmarking software or parameter settings to optimize assembly quality for a particular dataset or organism (Bradnam et al. 2013; Zhang et al. 2022; Cosma et al. 2023). These difficulties are further compounded in the absence of a reference genome, where defining the "best" assembly becomes even more ambiguous.

Several tools have been developed in order to address this question, such as Genome Assembly Evaluation Process (GAEP) (Zhang et al. 2023), GenomeQC (Manchanda et al. 2020) and the most commonly used QUAST (Mikheenko et al. 2023). However, most are either poorly documented, difficult to install and/or use, or only evaluate only a specific subset of relevant statistics. For example, GenomeQC, without a reference assembly or gene annotations, only calculates simple assembly stats such as contig N50 and gene content using BUSCO. Similarly, QUAST relies on reference assembly comparisons for several statistics such as structural accuracy and completeness. Although GAEP evaluates a number of important statistics such as contiguity, gene content, accuracy and structural errors, it does not assess coverage, completeness or telomerality, and is not packaged for use as a software container or conda package for use in high-throughput computing environments.

To address these challenges, we developed the Post-Assembly Quality manager (PAQman), a tool that calculates seven reference-free features of genome quality—Contiguity, Gene content, Completeness, Accuracy, Correctness, Coverage and Telomerality—by integrating multiple existing and popular tools and custom scripts. PAQman requires only the assembly itself and the

long-read data used to generate it, while offering users an accessible and standardized approach for genome quality assessment in high-throughput computing environments.

Description

The workflow summarized in Fig. 1 describes the two commands used in PAQman. The first, *paqman*, performs the separate evaluation of seven features of assembly quality prior to compiling the summary statistics table. By default, the minimum requirements for this command are an assembly in FASTA format (-a) and long-reads in FASTQ format (-l). Additional short paired-end reads may also be provided (-1 -2; FASTQ). Each feature and the tool/s used for its evaluation are outlined below:

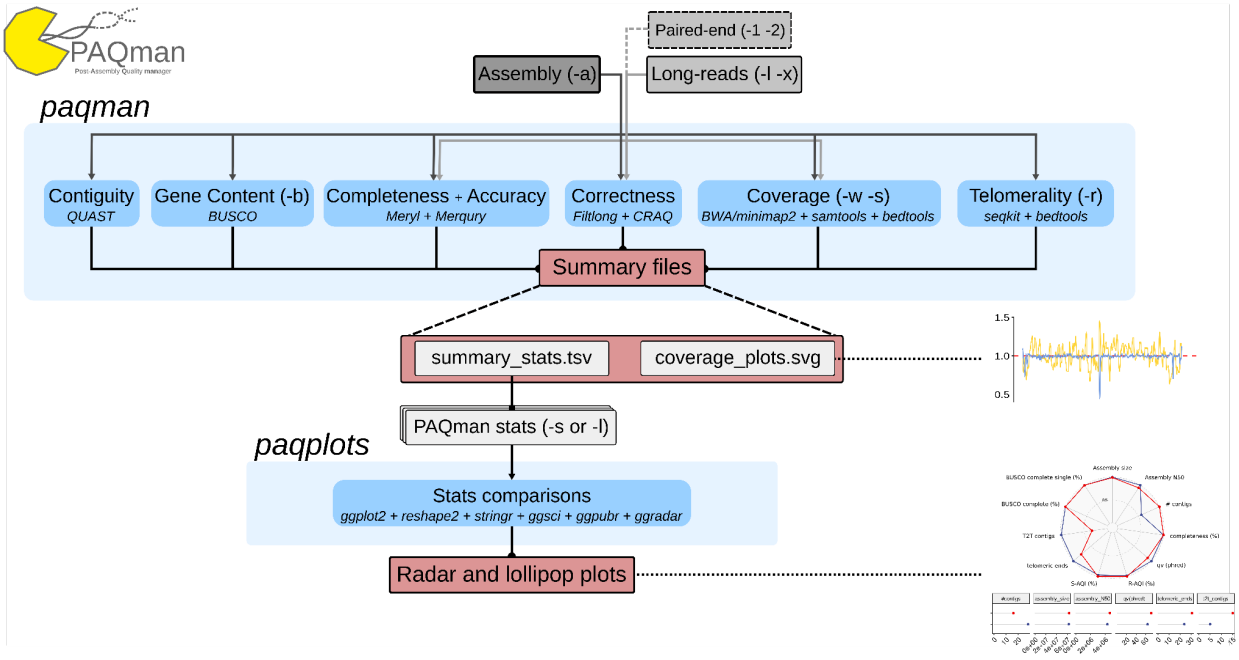


Figure 1. Schematic of the PAQman pipeline consisting of the *paqman* and paqplots commands. Grey boxes indicate files, blue boxes indicate computational steps calculating specific assembly metrics (with software names italicized below) and red indicate outputs. Example figures output by PAQman are to the right. Brackets indicate the short-hand command line arguments for input files and parameters impacting that specific step.

<u>Feature 1: Contiguity.</u> By far and above, the most common metrics evaluated post-assembly are the number of contigs/scaffolds, the assembly size and the contig/scaffold N50. PAQman runs QUAST (Mikheenko et al. 2023) for the rapid calculation of these important metrics. Generally

speaking, a lower number of scaffolds and larger assembly size and contig N50 are associated with higher quality assemblies.

Feature 2: Gene content. Apart from contiguity, the most common tool used to evaluate assemblies is BUSCO, which quantifies the recovery of various databases of universally conserved single copy orthologs (Tegenfeldt et al. 2025). Using the presence/absence of near-universal single-copy orthologs, BUSCO gives a proxy for assembly completeness in terms of gene content. PAQman parameter *-b* allows users to provide the most appropriate database for BUSCO to use given the taxonomic lineage of the focal organism (more database information at https://busco.ezlab.org/busco_userguide.html#obtain-busco). The percent recovery of single copy orthologs (ranging from 0-100%) is positively associated with assembly quality.

Feature 3: Completeness. Genomic regions containing a high density of protein-coding genes are generally the easiest to assemble due to relatively low repetitiveness compared to non-genic regions. Therefore, in addition to gene content, assembly completeness may also be more globally evaluated in terms of whether k-mers present in the raw read inputs are also present in the assembly. The k-mer-based completeness statistic is thus a more accurate representation of how much of the raw sequencing data is present in a final assembly, including both repetitive and non-repetitive content. PAQman first builds a k-mer distribution of the reads, selecting the short-reads (-1 -2) if provided, using Meryl *count* (Miller et al. 2008) then provides this Meryl database and the assembly to Merqury (Rhie et al. 2020) to calculate the k-mer based completeness. Completeness is calculated as a percentage with a maximum of 100%.

Feature 4: Accuracy. Using the k-mer distributions calculated for Feature 3, Merqury also estimates the number of genome-wide sequencing errors and provides a Phred quality score ($=-10\times\log_{10}(Pe)$; Pe: estimated probability of error) for the assembly. This estimate indicates how many errors may still be present in an assembly and can therefore be used to test whether common assembly polishing tools/pipelines improve quality. In general assemblies with a Phred Quality greater than 30 are considered of high quality, although 45+ is common with more recent long-read chemistries and polishing methods.

Feature 5: Correctness. In the age of reference-quality assemblies, users are not only interested in whether genome content is present *when* it should be, but also if it is present *where* it should be, i.e., assembled correctly at a structural level. CRAQ (Li et al. 2023) is a tool that uses read

139 mapping evidence to highlight regions within an assembly that have potential assembly errors.
140 This provides an estimate for genome-wide structural accuracy, thereby providing evidence to
141 support claims of assembly contiguity. PAQman initially down-samples the provided long-read
142 sequencing data to a maximum of 50X coverage, with a higher weighted selection for read length
143 using Filtlong (https://github.com/rrwick/Filtlong/), and then provides the filtered read dataset to
144 CRAQ in addition to short-reads, if provided. PAQman then focuses on two summary statistics,
145 the R-AQI and S-AQI, that score the assembly quality based on the detection of small regional
146 (R) and large structural (S) errors respectively. This score is a percentage with 100% meaning
147 the absence of detectable errors.
148
149 Feature 6: Coverage. Using read mapping, PAQman also evaluates whether coverage varies in
150 particular regions compared to the genome-wide median. One important use of this evaluation
151 metric is to determine whether multi-copy repeat regions have collapsed into a single copy during
152 assembly, as coverage will noticeably increase in collapsed regions. Coverage can therefore help
153 in understanding and locating more complex rearrangements such as large duplications and
154 aneuploidies. PAQman uses bwa mem (Li 2013) and/or minimap2 (Li 2018) for short- and long-
155 read mapping respectively; samtools (Danecek et al. 2021) and bedtools (Quinlan and Hall 2010)
156 for pre- and post-processing; and custom Rscripts with ggplot2 (Wickham 2016) for plotting the
157 relative coverage normalised by the genome-wide median (Fig. 1). PAQman parameters *-w* and
158 *-s* control the size of the window and slide, respectively, used to average coverage across the
159 genome. For a final statistic, PAQman calculates the percentage of the genome that is within two
160 standard deviations of the genome-wide median coverage. Ideally, larger percentage values are
161 more desirable.
162
163 Feature 7: Telomerality. This term defines a set of statistics that indicate whether an assembly
164 contains assembled telomeric ends, which is a good indication that complex subtelomeric regions
165 have been assembled and that contigs may thus represent full telomere-to-telomere (T2T)
166 chromosomes. In most cases, this statistic will be used to elevate assemblies ranging from very
167 good quality to complete. Organism-specific telomeric repeats are provided through the *-r*
168 parameter, quickly identified using seqkit *locate* (Shen et al. 2016) and processed using bedtools
169 *merge* for overlapping repeats with a max of one missing repeat between. Repeats are considered
170 telomeric if the aggregated repeat regions are at least the size of two repeats. Contig/scaffold
171 ends are considered to be capped by telomeres if the distance of a repeat to an end is less than
172 75% of the size of the entire telomeric repeat region.

The second optional command in PAQman, *paqplot*, takes the summary statistics output by *paqplot* from multiple assemblies and generates radar and lollipop plots to help benchmark multi-assembly comparisons using both the raw and relative values (Fig. 2).

We tested PAQman on a dataset of five publicly available assemblies of the *Saccharomyces cerevisiae* reference strain S288c (GCA_000146045.2, GCA_902192305.1; GCA_022626425.2; GCA_002057635.1; GCA_016858165.1). Each was evaluated using the same ONT dataset (SRR17374240) (Zhang et al. 2022) downsampled from 800x to 100x coverage using Filtlong (*-t 1200000000 --length_weight 5*). *paqman.sh* was run with default options except '*-b saccharomycetaceae*' and '*-r GGTGTG*' for the BUSCO db and telomeric repeat respectively. All summary stats were then evaluated by *paqplots.sh* using default settings (Fig. 2). The full pipeline and results can be found here https://github.com/SAMtoBAM/PAQman/tree/main/pub_examples/cerevisiae. The lower AQI values in GCA022626425, indicative of assembly errors, were confirmed by whole genome alignment (data not shown).

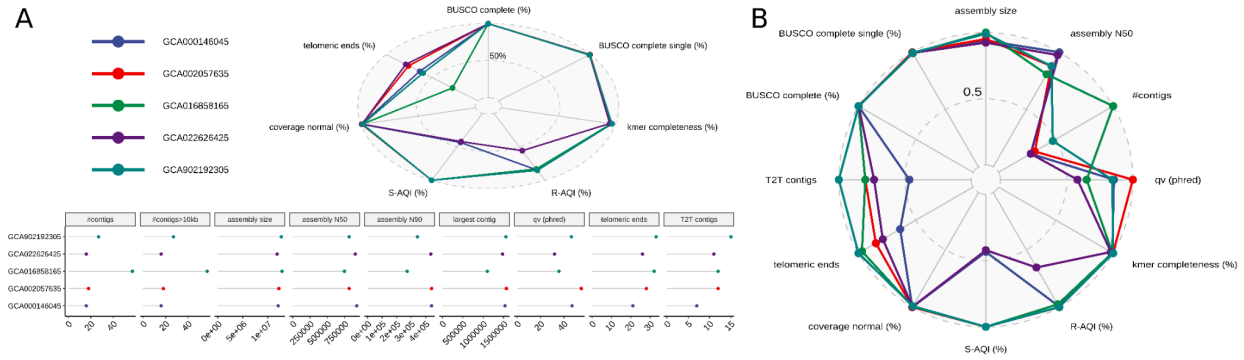


Figure 2. Output plot examples from paqplot using five different assemblies of the *S. cerevisiae* reference strain S288c. (A) Radar and lollipop plots using the raw values of several metrics. (B) Radar plot using relative scales of metrics between assemblies. The individual metrics plotted represent all 7 features of assembly quality measured by PAQman; 1.Contiguity ('assembly size', 'assembly N50', 'assembly N90', 'largest contig', '#contigs' and '#contigs>10kb'); 2.Gene content ('BUSCO complete single (%)' and 'BUSCO complete (%)'); 3.Completeness ('kmer completeness (%)'); 4.Accuracy ('qv (phred)'); 5.Correctness ('R-AQI (%)' and 'S-AQI (%)'); 6.Coverage ('coverage normal (%)'); and 7.Telomerality ('Telomeric ends (%)', 'Telomeric ends' and 'T2T contigs'). The assembly colored in red represents a somewhat ideal assembly quality, exemplified by being close to the outer edges of the radar plots except for the number of contigs ('#contigs') in the relative scaled radar in (B).

Based on the test dataset (5 assemblies; 12Mb genome size; 100X coverage) PAQman required on average 10.5 minutes per assembly to complete using only 8 threads and 16GB RAM (Table 1).

Table 1: Average run time across five assemblies results for paqman.sh using an AMD EPYC 7513 32-Core Processor and x86_64 architecture. Run time was calculated using the *time* command in Linux. Each of the five assemblies is ~12Mb and the long-read ONT data was downsampled to 100X prior to running PAQman.

system specifications		Run times		
threads	RAM (GB)	real (minutes)	user (minutes)	sys (minutes)
8	16	10.5	43.5	1.8
16	16	8.5	41.3	2.1
32	16	7.9	40.5	2.8

Installation, usage and citation

The easiest way to install PAQman is using the conda package manager:

conda config --append channels pwwang
conda install samtobam::paqman

It may also be run as an Apptainer/Singularity image:

docker pull ghcr.io/samtobam/paqman:latest

Both PAQman commands are run as below with default options (additional arguments available for paqman.sh):

paqman.sh -a path/to/assembly.fa -l path/to/longreads.fq
paqplots.sh -s path/to/combined_summary_stats.tsv

The PAQman GitHub (https://github.com/SAMtoBAM/PAQman) contains descriptions of installation/usage, commands, parameters, output files and both the features and their metrics.

PAQman has many dependencies that are stand-alone programs in their own right. Please cite both this manuscript in addition to its core dependencies. For example: “We used PAQman v1.1.0 (O’Donnell et al. 2025) in conjunction with Quast (Mikheenko et al. 2023), BUSCO (Tegenfeldt et al. 2025), Meryl (Miller et al. 2008), Mercury (Rhie et al. 2020), Filtlong (https://github.com/rrwick/Filtlong/), CRAQ (Li et al. 2023), BWA (Li 2013), minimap2 (Li 2018), samtools (Danecek et al. 2021), bedtools (Quinlan and Hall 2010), seqkit (Shen et al. 2016) and ggplot2 (Wickham 2016) to assess and visualize assembly quality.”

Conclusion

As the ease of producing reference-quality assemblies continues to advance, new tools are needed to comprehensively evaluate assembly quality in a high-throughput manner. PAQman provides a simple means of evaluating an assembly's overall quality by streamlining the calculation of seven commonly used features in the absence of a reference assembly. PAQman thus not only helps identify possible issues present within a specific assembly but simplifies the process of benchmarking assembly software to find the set of parameters that produce the “best” assembly for a given sequencing dataset and organism.

Author contributions

Samuel O’Donnell (Conceptualization, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing)
Ningxiao Li (Data curation, Writing – review & editing)
Jacob L. Steenwyk (Conceptualization, Writing – review & editing)
David Geiser (Resources, Writing – review & editing)
Frank N. Martin (Data curation, Resources, Writing – review & editing)
Emile Gluck-Thaler (Conceptualization, Funding acquisition, Resources, Writing – review & editing)

Funding

EGT and SO are supported by the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison with funding from the Wisconsin Alumni Research Foundation and the Department of Plant Pathology at the University of Wisconsin-Madison. JLS is a Howard Hughes Medical Institute Awardee of the Life Sciences Research Foundation.

Conflicts of Interest

JLS is an advisor to ForensisGroup Inc. JLS is a scientific consultant to FutureHouse Inc.

References

Bradnam KR, Fass JN, Alexandrov A, Baranay P, Bechner M, Birol I, Boisvert S, Chapman JA, Chapuis G, Chikhi R, et al. 2013. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. GigaScience 2:10.

Cosma B-M, Shirali Hossein Zade R, Jordan EN, van Lent P, Peng C, Pillay S, Abeel T. 2023. Evaluating long-read de novo assembly tools for eukaryotic genomes: insights and considerations. GigaScience 12:giad100.

Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM, et al. 2021. Twelve years of SAMtools and BCFtools. GigaScience 10:giab008.

Espinosa E, Bautista R, Larrosa R, Plata O. 2024. Advancements in long-read genome sequencing technologies and algorithms. Genomics 116:110842.

Giani AM, Gallo GR, Gianfranceschi L, Formenti G. 2020. Long walk to genomics: History and current approaches to genome sequencing and assembly. Comput. Struct. Biotechnol. J. 18:9–19.

Kolmogorov M, Yuan J, Lin Y, Pevzner PA. 2019. Assembly of long, error-prone reads using repeat graphs. Nat. Biotechnol. 37:540–546.

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997 [Internet]. Available from: http://arxiv.org/abs/1303.3997

Li H. 2018. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 34:3094–3100.

Li K, Xu P, Wang J, Yi X, Jiao Y. 2023. Identification of errors in draft genome assemblies at single-nucleotide resolution for quality assessment and improvement. Nat. Commun. 14:6556.

Manchanda N, Portwood JL, Woodhouse MR, Seetharam AS, Lawrence-Dill CJ, Andorf CM, Hufford MB. 2020. GenomeQC: a quality assessment tool for genome assemblies and gene structure annotations. BMC Genomics 21:193.

Mikheenko A, Saveliev V, Hirsch P, Gurevich A. 2023. WebQUAST: online evaluation of genome assemblies. Nucleic Acids Res. 51:W601–W606.

Miller JR, Delcher AL, Koren S, Venter E, Walenz BP, Brownley A, Johnson J, Li K, Mobarry C, Sutton G. 2008. Aggressive assembly of pyrosequencing reads with mates. Bioinformatics 24:2818–2824.

Quinlan AR, Hall IM. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 26:841–842.

Rhie A, Walenz BP, Koren S, Phillippy AM. 2020. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. Genome Biol. 21:245.

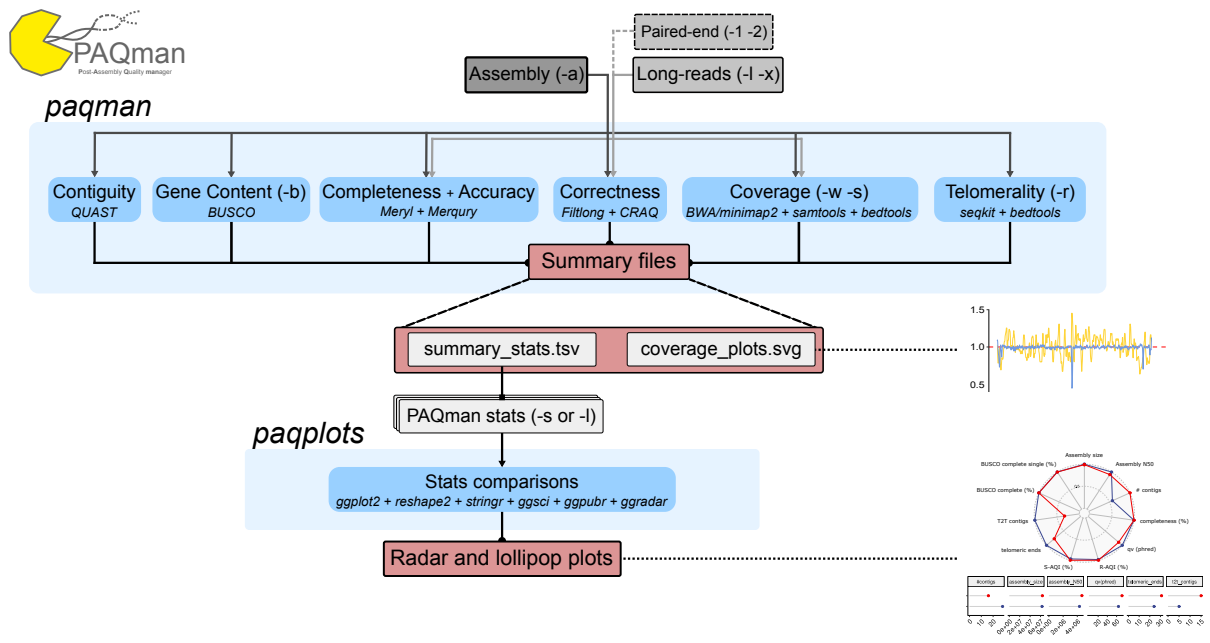
Shen W, Le S, Li Y, Hu F. 2016. SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation. PLOS ONE 11:e0163962.

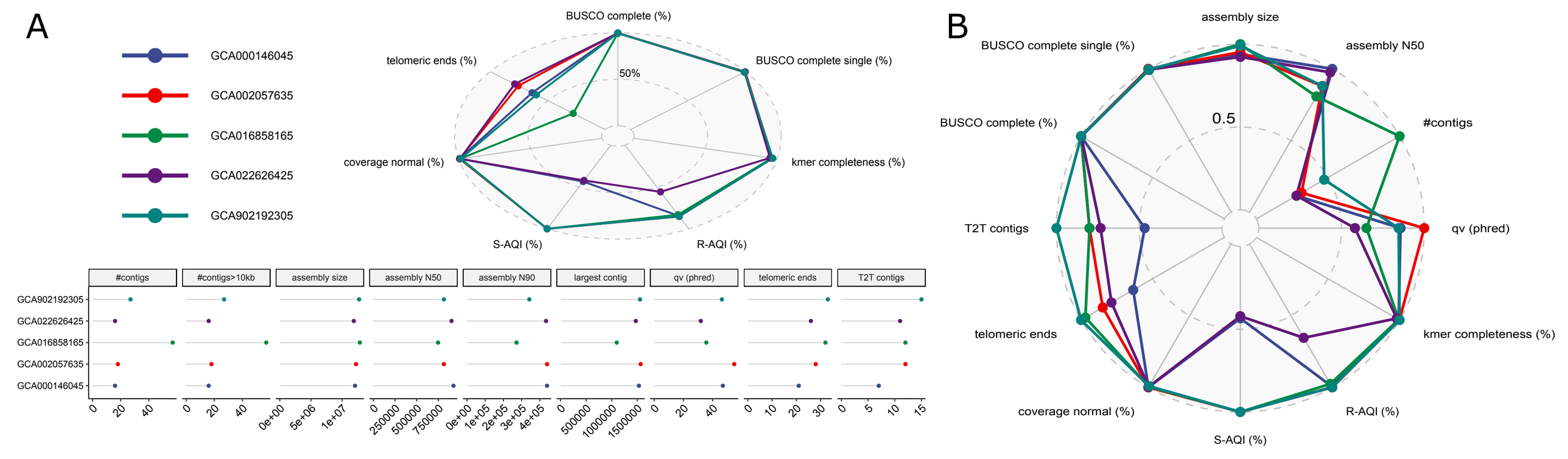
Tegenfeldt F, Kuznetsov D, Manni M, Berkeley M, Zdobnov EM, Kriventseva EV. 2025. OrthoDB and BUSCO update: annotation of orthologs with wider sampling of genomes. Nucleic Acids Res. 53:D516–D522.

Wickham H. 2016. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York Available from: https://ggplot2.tidyverse.org

Zhang X, Liu C-G, Yang S-H, Wang X, Bai F-W, Wang Z. 2022. Benchmarking of long-read sequencing, assemblers and polishers for yeast genome. Brief. Bioinform. 23:bbac146.

Zhang Y, Lu H-W, Ruan J. 2023. GAEP: a comprehensive genome assembly evaluating pipeline. J. Genet. Genomics 50:747–754.


PAQman
Post-Assembly Quality manager
Paired-end (-1 -2)
Assembly (-a)
Long-reads (-l -x)
paqman
Contiguity
QUAST
Gene Content (-b)
BUSCO
Completeness + Accuracy
Meryl + Merqury
Correctness
Filtlong + CRAQ
Coverage (-w -s)
BWA/minimap2 + samtools + bedtools
Telomerality (-r)
seqkit + bedtools
Summary files
summary_stats.tsv
coverage_plots.svg
1.5
1.0
0.5
PAQman stats (-s or -l)
paqplots
Stats comparisons
ggplot2 + reshape2 + stringr + ggsci + ggpubr + ggradar
Radar and lollipop plots
Assembly size
Assembly N50
contigs
completeness (%)
qv (phred)
R-AQI (%)
S-AQI (%)
telomeric ends
T2T contigs
BUSCO complete (%)
BUSCO complete single (%)
#contigs
assembly_size
assembly_N50
qv(phred)
telomeric_ends
t2t_contigs


A
GCA000146045
GCA002057635
GCA016858165
GCA022626425
GCA902192305
BUSCO complete (%)
BUSCO complete single (%)
kmer completeness (%)
R-AQI (%)
S-AQI (%)
coverage normal (%)
telomeric ends (%)
50%
#contigs
#contigs>10kb
assembly size
assembly N50
assembly N90
largest contig
qv (phred)
telomeric ends
T2T contigs
B
assembly size
assembly N50
#contigs
qv (phred)
kmer completeness (%)
R-AQI (%)
S-AQI (%)
coverage normal (%)
telomeric ends
T2T contigs
BUSCO complete (%)
BUSCO complete single (%)
0.5