Chapter 8

Nuclear Genome Assembly and Annotation of Kinetoplastids

Kristína Záhonová, Rajendra Mandage, and Vyacheslav Yurchenko

Abstract

In this chapter, we describe a pipeline for the nuclear genome analysis of kinetoplastids. Our approach relies on a combination of short- and long-sequencing reads and can be universally applied to studies of other kinetoplastids' genomes.

Key words Kinetoplastea, Euglenozoa, Trypanosomatids, Next-generation sequencing

1 Introduction

Kinetoplastea (Euglenozoa) is a group of unicellular eukaryotes with widely variable lifestyles, incorporating free-living species found in marine and freshwater environments, intracellular symbionts of other protists, and medically important parasites of vertebrates [1]. The evolutionary success of kinetoplastids across a wide range of ecological niches makes their genomes a compelling subject of study offering insights into the genomic changes underlying lifestyle transitions within the group [2, 3]. Presently, hundreds of kinetoplastid genomes are available in public databases, with an overwhelming majority of them belonging to members of the obligately parasitic family Trypanosomatidae [4, 5].

Kinetoplastid genomes exhibit considerable variation in size and complexity overlaid over a unique organization mode [6]. The smallest known kinetoplastid genome (~9.5 Mb) belongs to *Perkinsela* sp., an intracellular symbiont of an amoebozoan [7], while the largest (~40 Mb) to the free-living eubodonid *Bodo saltans* [3]. Protein-coding genes in the relatively compact kinetoplastid genomes are arranged in strand-specific clusters, transcribed as polycistronic units [8], and typically lack *cis*-spliceosomal introns, with only three known exceptions [9]. Trypanosomatids are characterized by high levels of synteny, i.e., conservation of gene order [10] that greatly facilitates and simplifies the genome annotation

Paul A. M. Michels et al. (eds.), *Euglenozoa: Methods and Protocols, Volume 1*, Methods in Molecular Biology, vol. 3013,
https://doi.org/10.1007/978-1-0716-5142-1_8,

[11]. Nevertheless, other features of kinetoplastid genomes may undesirably influence this process and must be carefully considered when evaluating the assembly quality. These include: (1) a high proportion of repetitive regions including those associated with the expansion of species- and/or lineage-specific protein families, exemplified by variant surface glycoproteins in *Trypanosoma brucei*, *trans*-sialidases in *Trypanosoma cruzi*, or amastins in *Leishmania* spp. [12–16]; (2) the prevalent presence of pseudogenes, particularly among genes encoding members of large protein families [17–19]; (3) widespread aneuploidy highlighting the need of having fully resolved phased haplotypes for proper analysis [20–22]; (4) presence of the short intergenic regions within polycistronic clusters [23, 24], as well as (5) presence of the clusters of tandemly duplicated genes [25, 26]. Genome assembly and annotation in some kinetoplastid lineages may be particularly challenging. A good example of this are members of the genus *Blastocrithidia*, in which all three universal stop codons have been reassigned to encode amino acids with UAA serving also as a genuine translation terminator [27–29]. All the concerns and restrictions mentioned above justify a standard practice of combining short- and long-sequencing reads for kinetoplastid genome assemblies that we describe in this chapter. We think that it will benefit the community of trypanosomatid researchers and facilitate further genome-guided investigations into the biology of these interesting and important parasites. Some parts of these protocols have been published before [14, 30, 31], but not in the pipeline form presented here.

2 Materials

2.1 Software

FastQC [32], BBDuk [33], Karect [34], SPAdes [35], Platanus [36], GapCloser from SOAPdenovo2 [37], Filtlong [38], proovread [39], Flye [40], BLAST+ [41], DIAMOND [42], QUAST [43], BUSCO [44], Bowtie2 [45], SAMtools [46], Hisat2 [47], Cufflinks [48], Gffread [49], tRNAscan-SE [50], ARAGORN [51], IGV [52], Artemis [53], Companion [54] (*see* **Note 1**).

3 Methods

3.1 Short Read Filtering

1. Check the quality of the raw short, i.e., Illumina paired-end, reads by FastQC using the following command:

```
fastqc -t 5 -o <output_directory> <short_reads_file.fastq.gz>
```

where -**t** is the number of threads.

2. Trim the reads with BBDuk to remove adaptors and low-quality reads. It is not necessary to unpack the read files beforehand. The typical command for trimming of paired-end short reads is:

```
bbduk in1=<forward_1.fastq.gz> in2=<reverse_2.fastq.gz>
out1=<trimmed_1.fastq.gz> out2=<trimmed_2.fastq.gz> ref=a-
dapters.fa usejni=t qtrim=rl trimq=20 ktrim=r k=22 mink=11
hdist=2 tpe tbo t=10 2> <bbduk_report.log>
```

In case of interleaved reads, the command uses one input file (**in=<reads.fastq.gz>**) and produces one output file (**out=<trimmed.fastq.gz>**). The exact description of parameters can be found in the BBDuk manual (*see* **Note 2**).

3. Check the quality of trimmed reads by FastQC using the command from **step 1**.
4. Perform error-correction of reads by Karect (*see* **Note 3**):

```
karect -correct -threads=10 -matchtype=hamming
-celltype=diploid -inputfile=<trimmed_1.fastq.gz> -inputfi-
le=<trimmed_2.fastq.gz> 2> <karect_report.log>
```

3.2 Long Reads Filtering

1. Filter the long reads, i.e., PacBio or Nanopore, with Filtlong using the short reads with high coverage and high depth as a reference data set:

```
filtlong -1 <trimmed_1.fastq.gz> -2 <trimmed_2.fastq.gz> --
min_length 1000 --keep_percent 90 --verbose <long_reads.fastq.
gz> > <filtered_long_reads.fastq>
```

Adding --**verbose** provides information about the read length and quality. The description of other parameters can be found on the Filtlong GitHub page.

2. Perform error-corrections on long reads with proovread using a high coverage short reads dataset.

```
proovread -i <filtered_long_reads.fastq> -s <trimmed_1.fastq.
gz> -s <trimmed_2.fastq.gz>  -o <corrected_long_reads.fastq>
```

where -**o** is a prefix of the output file.

3.3 Genome Assembly of Short Reads

1. Assemble trimmed reads de novo into contigs/scaffolds using SPAdes:

```
spades.py --pe1-1 <trimmed_1.fastq.gz> --pe1-2 <trimmed_2.
fastq.gz> -t 40 -o <spades_output_directory>
```

The description of other parameters can be found in the SPAdes manual. In the output directory, the resulting **contigs.fasta** and **scaffolds.fasta** files can be found. The **contigs.fasta** file should be scaffolded (*see* Subheading 3.3, **step 2** below), alternatively, the **scaffolds.fasta** can be used.

2. Scaffold contigs by Platanus:

```
platanus scaffold -o <platanus_rnd1> -c <contigs.fasta> -IP1
<trimmed_1.fastq> <trimmed_2.fastq> -t 40 2> <platanus_rnd1.
log>
```

where **-o** is a prefix of the output file. Note that the files of trimmed reads must be unpacked for this tool.

3. Use GapCloser module of SOAPdenovo2 for gap filling:

```
GapCloser -b <config_file> -a <platanus_rnd1_scaffold.fasta>
-o <platanus_rnd1_scaffold.gapcloser.fasta> -t 20 2> <plata-
nus_rnd1.gapcloser.log>
```

Note that this step requires a "**config_file**", an example of which can be found in the SOAPdenovo manual and should be modified according to your data.

4. Repeat **steps 2** and **3** once again using the produced scaffolds from the previous round (**platanus_rnd1_scaffold.gapcloser.fasta**).

3.4 Hybrid Assembly Using Both Short and Long Reads

1. Assemble filtered short and long reads into contigs using SPAdes:

```
spades.py --pe1-1 <trimmed_1.fastq.gz> --pe1-2
<trimmed_2.fastq.gz> --pacbio <corrected_long_reads.fastq> -t
40 -o <spades_output_directory>
```

2. Scaffold the contigs and polish the assembly by Flye:

```
Flye --pacbio-hifi <corrected_long_reads.fastq> <SPAdes_con-
tigs.fasta> --out-dir <flye_output_directory> --threads 40 --
iterations 3 --scaffold
```

where --**iterations** specifies assembly polishing number, --**scaffold** performs scaffolding of the assembly (*see* **Notes 4** and **5**).

3.5 Assembly Decontamination and Quality Assessment

1. Discard scaffolds <500 nt.
2. Screen scaffolds in BLASTn searches against the NCBI nucleotide database:

```
blastn -query <assembly.fasta> -db <NCBI_nt_database.fasta>
-out <blast_output.tsv> -outfmt 6 -evalue 1e-05 -num_threads
10
```

Discard scaffolds showing nucleotide identity >95% and query coverage >85% to non-euglenozoan sequences.

3. Screen scaffolds with non-euglenozoan hits below the specified thresholds in the previous step by DIAMOND in sensitive mode against the NCBI non-redundant database:

```
diamond blastx -q <updated_assembly.fasta> -d <NCBI_nr_data-
base.fasta> -o <diamond_output.tsv> -f 6 -evalue 1e-10 --
taxonmap prot.accession2taxid.gz --sensitive -p 10
```

where --**taxonmap** is a mapping file that maps NCBI protein accession numbers to taxon IDs (can be downloaded from NCBI: ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/accession2taxid/prot.accession2taxid.gz). Discard scaffolds with non-euglenozoan sequences retrieved as best hits.

4. Evaluate the completeness of the assembly by BUSCO:

```
busco -i <final_assembly.fasta> -l <lineage> -o <base_name> -m
genome -c 5
```

where -**l** specifies the name of the BUSCO lineage dataset (euglenozoa_odb10 or eukaryota_odb10), and -**m** specifies the assessment mode (note that BUSCO can be run in genome, transcriptome, and proteins mode) (*see* **Note 6**).

5. Map the trimmed reads back to the genome assembly using Bowtie2 and SAMtools. First, index the genome assembly:

```
bowtie2-build <final_assembly.fasta> <base_name>
```

Perform the read mapping:

```
bowtie2 --very-sensitive -p 50 -x <base_name> -1 <trimmed_1.
fastq.gz> -2 <trimmed_2.fastq.gz> --un-gz <unmapped_unpaired.
fastq.gz> --un-conc-gz <unmapped_paired.fastq.gz> --al-conc-
gz <mapped.fastq.gz> -S <mapped_reads.sam> 2> <bowtie2.log>
```

where -**x** specifies the same name from the indexing step, -**1** and -**2** are trimmed reads, --**very-sensitive** is a preset of alignment parameters, -**S** specifies the produced SAM file with read mapping.

Convert the SAM file into BAM file with sorted and indexed reads:

```
samtools view -bS -@ 50 <mapped_reads.sam> > <mapped_reads.
bam>
samtools sort -o <mapped_sorted_reads.bam> -@ 50 <mapped_-
reads.bam>
samtools index -b <mapped_sorted_reads.bam>
```

where <**mapped_reads.sam**> is the SAM file, <**mapped_-reads.bam**> is the BAM file, <**mapped_sorted_reads.bam**> contains the sorted mapped reads. The last command produces indexed BAM file (<**mapped_sorted_reads.bam.bai**>).

6. Inspect read mapping rate in the <**bowtie2.log**> file and visually in a genome viewer program (e.g., IGV or Artemis) after importing the genome assembly and mapped sorted reads.

3.6 Transcriptome Assembly

1. Check quality and filter the RNA-Seq reads as in **steps 1** and **2** of the Subheading 3.1 above (*see* **Note 2**).
2. Map trimmed reads to the genome assembly using Hisat2 and SAMtools. First, index the genome assembly:

```
hisat2-build -p 50 <final_assembly.
fasta> <base_name>
```

Perform the read mapping:

```
hisat2 --very-sensitive --dta --
secondary -p 50 -x <base_name> -1 <RNA_trimmed_1.fastq.gz> -2
<RNA_trimmed_2.fastq.gz> --un-gz <RNA_unmapped_unpaired.
fastq.gz> --un-conc-gz <RNA_unmapped_paired.fastq.gz> -S
<RNA_mapped_reads.sam> 2> <hisat2.log>
```

where --**dta** reports alignments tailored for transcript assemblers, and --**secondary** reports secondary alignments.

Convert the SAM file into BAM file with sorted and indexed reads:

```
samtools view -bS -@
50 <RNA_mapped_reads.sam> > <RNA_mapped_reads.bam>
samtools sort -o
<RNA_mapped_sorted_reads.bam> -@ 50 <RNA_mapped_reads.bam>
samtools index -b
<RNA_mapped_sorted_reads.bam>
```

3. Assemble reads into transcripts using Cufflinks:

```
cufflinks -p 40 -o <RNA_mapped_sorted_reads.bam>
```

This command produces **transcripts.gtf** file, which can be converted into gff file using Gffread:

```
gffread transcripts.gtf -o <organism_cufflinks.gff>
```

Using a perl script from Augustus [55] and awk command produces file with transcripts in fasta format:

```
perl getAnnoFasta.pl --seqfile=<final_assembly.fasta> --protein=off --codingseq=on <organism_cufflinks.gff>
awk 'BEGIN{FS=" "}{if(!/>/){print toupper($0)}else{print $1}}' <organism_cufflinks.mrna> > <organism_cufflinks.fasta>
```

3.7 Genome Annotation

1. Annotate protein-coding genes using Companion. In "Step 1: Basic job properties," specify properties of the run. "Step 2: Target sequence" asks for the genome assembly (<**final_assembly.fasta**>). In "Step 3: Transcript evidence," supply gtf file produced by Cufflinks in a previous step (**transcripts.gtf**; *see* Subheading 3.6 above). In "Step 4: Reference organism," choose a reference organism that is phylogenetically closest to yours (all available kinetoplastids can be found here: https://companion.gla.ac.uk/references). In "Step 5: Pseudochromosome contiguation," keep default settings. In "Step 6: Advanced settings," use the following options:
   - Yes, align reference proteins to target sequence.
   - Yes, perform pseudogene detection.
   - Yes, use RATT with the Species transfer type to transfer reference gene models.
2. Identify rRNA-coding genes by BLASTn searches using reference kinetoplastid sequences:

```
blastn -query <reference_rRNAs.fasta> -db <final_assembly.fasta> -out <blast_output.tsv> -outfmt 6 -num_threads 10
```

3. Identify tRNA-coding genes using tRNAscan-SE:

```
tRNAscan-SE -E -o <organism.tRNAscan_table.tsv> -a <organism.tRNAscan_seqs.fasta> --thread 10 <final_assembly.fasta>
```

where -**E** specifies eukaryotic genome, -**o** is tRNAscan-SE output in a tabular format and -**a** collects the predicted tRNA gene sequences. Alternatively, ARAGORN can be used:

```
aragorn -fo -o <organism.aragorn_seqs.fasta>
<final_assembly.fasta>
```

where -**fo** prints out sequences in fasta format only (without secondary structures).

3.8 Specific Cases

1. The genome annotation for *Blastocrithidia* spp. is more challenging as all three stop codons were reassigned as sense ones [27] and specific pipelines have been developed [29].
2. The genomes of diplonemids, a sister lineage of kinetoplastids [1], possess long repetitive regions, numerous transposable elements, or very long introns, i.e., features that require special caution and tinkering when assembling and annotating. The only high-quality genome of a diplonemid is that of *Paradiplonema papillatum* [56], renamed from *Diplonema papillatum* [57]. This assembly was produced by merging separate assemblies of short Illumina and long PacBio reads rather than using hybrid assembly as we describe in this chapter.

4 Notes

1. Please use latest versions of the software available.
2. When running BBDuk, filtering short reads with length <75 nt or <50 nt for DNA-Seq and RNA-Seq reads using **minlen=75** or **minlen=50** parameter, respectively, is recommended to improve the assembly.
3. Alternative error-correction tools for short reads (e.g., RECKONER [58], CARE [59]) are available.
4. Alternative hybrid assembly tools (e.g., MaSuRCA [60], WENGAN [61]) are also available.
5. Caution should be taken when performing rounds of the assembly scaffolding and polishing as over-polishing may introduce errors.
6. BUSCO can be run in the genome mode since kinetoplastids contain very few introns [9].

Acknowledgments

This work was supported by the European Union Operational Program "Just Transition" (LERCO CZ.10.03.01/00/22_003/

0000003) and the Czech Ministry of Education, Youth and Sports (MEYS CZ) (INTER-EXCELLENCE-LUASK22033) to V.Y. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the MEYS CZ.

References

1. Kostygov AY, Karnkowska A, Votýpka J et al (2021) Euglenozoa: taxonomy, diversity and ecology, symbioses and viruses. Open Biol 11: 200407
2. Butenko A, Hammond M, Field MC et al (2021) Reductionist pathways for parasitism in euglenozoans? Expanded datasets provide new insights. Trends Parasitol 37(2):100–116
3. Jackson AP (2015) Genome evolution in trypanosomatid parasites. Parasitology 142(Suppl 1):S40–S56
4. Yurchenko V, Butenko A, Kostygov AY (2021) Genomics of Trypanosomatidae: where we stand and what needs to be done? Pathogens 10(9):1124
5. Briggs EM, Marques CA, Reis-Cunha J et al (2020) Next-generation analysis of trypanosomatid genome stability and instability. In: Methods Mol Biol, vol 2116, pp 225–262
6. Maslov DA, Opperdoes FR, Kostygov AY et al (2019) Recent advances in trypanosomatid research: genome organization, expression, metabolism, taxonomy and evolution. Parasitology 146(1):1–27
7. Tanifuji G, Cenci U, Moog D et al (2017) Genome sequencing reveals metabolic and cellular interdependence in an amoeba-kinetoplastid symbiosis. Sci Rep 7(1):11688
8. Clayton C (2019) Regulation of gene expression in trypanosomatids: living with polycistronic transcription. Open Biol 9(6):190072
9. Kostygov AY, Skýpalová K, Kraeva N et al (2024) Comprehensive analysis of the Kinetoplastea intron landscape reveals a novel intron-containing gene and the first exclusively *trans*-splicing eukaryote. BMC Biol 22(1):281
10. Ghedin E, Bringaud F, Peterson J et al (2004) Gene synteny and evolution of genome architecture in trypanosomatids. Mol Biochem Parasitol 134(2):183–191
11. Wu F, Mai Y, Chen C et al (2024) SynGAP: a synteny-based toolkit for gene structure annotation polishing. Genome Biol 25(1):218
12. Silva Pereira S, Jackson AP, Figueiredo LM (2022) Evolution of the variant surface glycoprotein family in African trypanosomes. Trends Parasitol 38(1):23–36
13. Pita S, Díaz-Viraqué F, Iraola G et al (2019) The tritryps comparative repeatome: insights on repetitive element evolution in trypanosomatid pathogens. Genome Biol Evol 11(2): 546–551
14. Albanaz ATS, Gerasimov ES, Shaw JJ et al (2021) Genome analysis of *Endotrypanum* and *Porcisia* spp., closest phylogenetic relatives of *Leishmania*, highlights the role of amastins in shaping pathogenicity. Genes 12(3):444
15. Jackson AP (2010) The evolution of amastin surface glycoproteins in trypanosomatid parasites. Mol Biol Evol 27(1):33–45
16. Freitas LM, dos Santos SL, Rodrigues-Luiz GF et al (2011) Genomic analyses, gene expression and antigenic profile of the *trans*-sialidase superfamily of *Trypanosoma cruzi* reveal an undetected level of complexity. PLoS One 6(10):e25914
17. Abrahim M, Machado E, Alvarez-Valin F et al (2022) Uncovering pseudogenes and intergenic protein-coding sequences in TriTryps' genomes. Genome Biol Evol 14(10):evac142
18. Durante IM, Butenko A, Rašková V et al (2020) Large-scale phylogenetic analysis of trypanosomatid adenylate cyclases reveals associations with extracellular lifestyle and host-pathogen interplay. Genome Biol Evol 12(12):2403–2416
19. Santi AMM, Ribeiro JM, Reis-Cunha JL et al (2022) Disruption of multiple copies of the prostaglandin F2alpha synthase gene affects oxidative stress response and infectivity in *Trypanosoma cruzi*. PLoS Negl Trop Dis 16(10): e0010845
20. Reis-Cunha JL, Pimenta-Carvalho SA, Almeida LV et al (2024) Ancestral aneuploidy and stable chromosomal duplication resulting in differential genome structure and gene expression control in trypanosomatid parasites. Genome Res 34(3):441–453
21. Dumetz F, Imamura H, Sanders M et al (2017) Modulation of aneuploidy in *Leishmania donovani* during adaptation to different *in vitro* and *in vivo* environments and its impact on gene expression. mBio 8(3):e00599–e00517

22. Negreira GH, de Groote R, Van Giel D et al (2023) The adaptive roles of aneuploidy and polyclonality in *Leishmania* in response to environmental stress. EMBO Rep 24(9):e57413
23. Waithaka A, Maiakovska O, Grimm D et al (2022) Sequences and proteins that influence mRNA processing in *Trypanosoma brucei*: evolutionary conservation of SR-domain and PTB protein functions. PLoS Negl Trop Dis 16(10): e0010876
24. Novak EM, de Mello MP, Gomes HB et al (1993) Repetitive sequences in the ribosomal intergenic spacer of *Trypanosoma cruzi*. Mol Biochem Parasitol 60(2):273–280
25. Field H, Field MC (1997) Tandem duplication of *rab* genes followed by sequence divergence and acquisition of distinct functions in *Trypanosoma brucei*. J Biol Chem 272(16): 10498–10505
26. Zakharova A, Tashyreva D, Butenko A et al (2023) A neo-functionalized homolog of host transmembrane protein controls localization of bacterial endosymbionts in the trypanosomatid *Novymonas esmeraldas*. Curr Biol 33(13): 2690–2701
27. Záhonová K, Kostygov A, Ševčíková T et al (2016) An unprecedented non-canonical nuclear genetic code with all three termination codons reassigned as sense codons. Curr Biol 26(17):2364–2369
28. Opperdoes FR, Záhonová K, Škodová-Sveráková I et al (2024) *In silico* prediction of the metabolism of *Blastocrithidia nonstop*, a trypanosomatid with non-canonical genetic code. BMC Genomics 25(1):184
29. Kachale A, Pavlíková Z, Nenarokova A et al (2023) Short tRNA anticodon stem and mutant eRF1 allow stop codon reassignment. Nature 613(7945):751–758
30. Albanaz ATS, Carrington M, Frolov AO et al (2023) Shining the spotlight on the neglected: new high-quality genome assemblies as a gateway to understanding the evolution of Trypanosomatidae. BMC Genomics 24(1):471
31. Butenko A, Kostygov AY, Sádlová J et al (2019) Comparative genomics of *Leishmania* (*Mundinia*). BMC Genomics 20(1):726
32. Andrews S (2019) FastQC: a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc. Accessed 25 Feb 2025
33. Bushnell B, Rood J, Singer E (2017) BBMerge – accurate paired shotgun read merging *via* overlap. PLoS One 12(10):e0185056
34. Allam A, Kalnis P, Solovyev V (2015) Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. Bioinformatics 31(21): 3421–3428
35. Bankevich A, Nurk S, Antipov D et al (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol 19(5):455–477
36. Kajitani R, Toshimoto K, Noguchi H et al (2014) Efficient *de novo* assembly of highly heterozygous genomes from whole-genome shotgun short reads. Genome Res 24(8): 1384–1395
37. Luo R, Liu B, Xie Y et al (2012) SOAPdenovo2: an empirically improved memory-efficient short-read *de novo* assembler. Gigascience 1(1):18
38. Wick RR (2017) Filtlong. https://github.com/rrwick/Filtlong. Accessed 25 Feb 2025
39. Hackl T, Hedrich R, Schultz J et al (2014) *Proovread*: large-scale high-accuracy PacBio correction through iterative short read consensus. Bioinformatics 30(21):3004–3011
40. Kolmogorov M, Yuan J, Lin Y et al (2019) Assembly of long, error-prone reads using repeat graphs. Nat Biotechnol 37(5):540–546
41. Camacho C, Coulouris G, Avagyan V et al (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421
42. Buchfink B, Reuter K, Drost HG (2021) Sensitive protein alignments at tree-of-life scale using DIAMOND. Nat Methods 18(4): 366–368
43. Gurevich A, Saveliev V, Vyahhi N et al (2013) QUAST: quality assessment tool for genome assemblies. Bioinformatics 29(8):1072–1075
44. Tegenfeldt F, Kuznetsov D, Manni M et al (2025) OrthoDB and BUSCO update: annotation of orthologs with wider sampling of genomes. Nucleic Acids Res 53(D1):D516–D522
45. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9(4):357–359
46. Danecek P, Bonfield JK, Liddle J et al (2021) Twelve years of SAMtools and BCFtools. Gigascience 10(2):1–4
47. Kim D, Paggi JM, Park C et al (2019) Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat Biotechnol 37(8):907–915
48. Trapnell C, Roberts A, Goff L et al (2012) Differential gene and transcript expression

analysis of RNA-seq experiments with TopHat and cufflinks. Nat Protoc 7(3):562–578

49. Pertea G, Pertea M (2020) GFF utilities: GffRead and GffCompare. F1000Res 9:304
50. Chan PP, Lin BY, Mak AJ et al (2021) tRNAscan-SE 2.0: improved detection and functional classification of transfer RNA genes. Nucleic Acids Res 49(16):9077–9096
51. Laslett D, Canback B (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. Nucleic Acids Res 32(1):11–16
52. Robinson JT, Thorvaldsdottir H, Winckler W et al (2011) Integrative genomics viewer. Nat Biotechnol 29(1):24–26
53. Carver T, Harris SR, Berriman M et al (2012) Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. Bioinformatics 28(4): 464–469
54. Steinbiss S, Silva-Franco F, Brunk B et al (2016) Companion: a web server for annotation and analysis of parasite genomes. Nucleic Acids Res 44(W1):W29–W34
55. Stanke M, Keller O, Gunduz I et al (2006) AUGUSTUS: *ab initio* prediction of alternative transcripts. Nucleic Acids Res 34: W435–W439
56. Valach M, Moreira S, Petitjean C et al (2023) Recent expansion of metabolic versatility in *Diplonema papillatum*, the model species of a highly speciose group of marine eukaryotes. BMC Biol 21(1):99
57. Tashyreva D, Simpson AGB, Prokopchuk G et al (2022) Diplonemids – a review on "new" flagellates on the oceanic block. Protist 173(2): 125868
58. Dlugosz M, Deorowicz S (2017) RECKONER: read error corrector based on KMC. Bioinformatics 33(7):1086–1089
59. Kallenborn F, Hildebrandt A, Schmidt B (2021) CARE: context-aware sequencing read error correction. Bioinformatics 37(7): 889–895
60. Zimin AV, Marcais G, Puiu D et al (2013) The MaSuRCA genome assembler. Bioinformatics 29(21):2669–2677
61. Di Genova A, Buena-Atienza E, Ossowski S et al (2021) Efficient hybrid *de novo* assembly of human genomes with WENGAN. Nat Biotechnol 39(4):422–430