UiT The Arctic University of Norway

The Norwegian College of Fishery Science, Faculty of Biosciences, Fisheries and Economics

Phylogenomic Study of 36 Polychaete Species in a North Norwegian Fjord

GengYing,SHAO

Master thesis in Marine Biotechnology …. BIO-3901 …. May 2020

Table of Contents

1 Abstract 1

2 Introduction 1

2.1 Why do we study marine Polychaeta? 1

2.2 Bioinformatics Pipeline 2

2.2.1 NOVOPlasty 4

2.2.2 MitoZ 4

2.2.3 MITObim 6

2.2.4 MITOS 7

2.2.5 ATGC Bioinformatics Platform 7

2.3 Phylogenomic Study 9

3 Material 13

4 Methods 17

4.1 Assembly 18

4.1.1 NOVOPlasty 18

4.1.2 MitoZ 20

4.1.3 MITObim 21

4.2 Annotation 23

4.2.1 MitoZ 23

4.2.2 MITOS 23

4.3 Phylogenetic Tree 24

4.3.1 Preparation - COI for Tree 24

4.3.2 Preparation – Mitochondrial Genome for Tree 24

4.3.3 ATGC Bioinformatics Platform 25

4.4 Useful Scripts 26

5 Results 26

5.1 The Nature of Our Sample Datasets 26

5.2 Assembly 28
5.3 Annotation 29
5.3.1 COI Sequence 29
5.3.2 Mitochondrial Genome Sequence 30
5.3.3 MitoZ vs MITOS 32
5.4 Mitochondrial Genome Visualization & Gene Order 34
5.5 Phylogenetic Tree 37
5.5.1 Based on Nearly Completed Mitochondrial Genome 38
5.5.2 Based on Multiple Genes from Partial Mitochondrial Genome 39
5.5.3 Based on Single Gene: COI 41
6 Discussion 41
6.1 The Nature of Our Sample Datasets 41
6.2 Bioinformatics Pipeline 42
6.2.1 Assembly Tools 42
6.2.2 Annotation Tools 47
6.2.3 Phylogenetic Tree Tools 49
6.3 Mitochondrial Gene Order 50
6.4 Phylogenomic Study 51
6.4.1 Single Gene, Multiple Genes or Mitochondrial Genome? 51
6.4.2 Polychaete Phylogenetic Relationships 54
6.4.3 Evolutionary Direction 59
7 References 63
Appendix 68

Abbreviations

ND 1-6	NADH dehydrogenase subunits 1-6
COX 1-3	cytochrome oxidase subunits 1-3
CYTB	cytochrome b
ATP 6/8	ATP synthase subunit 6/8
l-rRNA	large ribosomal RNA
s-rRNA	small ribosomal RNA
COI	cytochrome-oxidase subunit 1

Preface

They say, “In law, we are equal”. I say, “In science, we are equal.”

Yes, 1000 is more than 1. However, if one compare to 1 trillion, then 1000 and 1 are basically not much difference. That is how I view scientific world. Compare to the unknown out there, we human beings basically still know nothing. Yet we are trying here very hard to explore this fantastic unknown world, trying to explain it in each of our own way.

People have different visions and I hope you can see mine. Thank you for your time!

1 Abstract

Annelida, the segmented worms, is a highly diverse animal phylum that includes over 15,000 described species. Polychaeta is one of the major groups of Annelida, also called bristle worms. The vast majority of them are marine, including more than 11,000 described species. In this thesis, we created a new bioinformatics pipeline which produced I) 26 novel COI sequences with length between 913bp and 3391bp, II) 12 novel nearly completed mitochondrial genomes with length between 12331bp and 17681bp and III) 9 novel partial mitochondrial genomes. Among the 12 nearly completed mitochondrial genomes, we find out IV) 2 novel Annelida mitochondrial gene orders. Furthermore, we used 1) single gene: COI, 2) multiple genes from partial mitochondrial genomes and 3) nearly completed mitochondrial genomes to perform the phylogenomic study for Polychaete phylogeny. The results show that COI sequences are probably not the best genetic marker whereas partial or nearly completed mitochondrial genomes have high potentials to resolve the phylogenetic relationships among Polychaetes. Moreover, we propose the phylogenetic relationships among Polychaetes are 1) most of the species that come from same Order/Suborder/Family form monophyletic clades, 2) the species that come from different Order/Suborder/Family, evolutionary direction from I) Ampharetidae → Spionidae → Orbiniidae → Aphroditiformia; II) Ampharetidae → Nephtyidae; III) Ampharetidae → Maldanidae → Aphroditiformia. *Owenia fusiformis* is the basal group with high bootstrap support 1000. Finally, Order Sabellida, Family Ampharetidae and Orbiniidae, are probably the important taxa during Polychaete phylogeny that future study could focus more on.

Key words: Annelida, Polychaeta, bioinformatics pipeline, novel COI sequences, novel nearly completed mitochondrial genomes, novel Annelida mitochondrial gene orders, phylogenomic study, Polychaete phylogeny.

2 Introduction

2.1 Why do we study marine Polychaeta?

Annelida, the segmented worms, is a highly diverse animal phylum that includes over 15,000 described species. It makes up the dominant benthic macro-organism from the intertidal zone, shallow waters, down to the deep ocean (Struck, Paul *et al*., 2011).

Polychaeta is one of the major groups of Annelida, also called bristle worms. The vast majority of them are marine, including more than 11,000 described species (Pamungkas, Glasby *et al.*, 2019). Its high diversity is shown in body features, way of living as well as feeding and reproductive strategies (Rouse & Pleijel, 2001) .

Polychaetes are studied for various purposes. For example, 1) used for biomonitoring purpose, to monitor toxic materials and serve as pullution indicators as well as marine environmental quality indicators (Pocklington & Wells, 1992). 2) used for bioprospecting purpose, potential applications are porphyrinoid pigments; luminescence and fluorescent probes; eco-friendly pesticides and anti-foulants; painkillers, anaesthetics and relaxants; drug development for anti-cancer, antimicrobial and antivirals purposes (Rodrigo & Costa, 2019). 3) used for evolutionary developmental studies, usually the researches are focus on Annelida level. To provide the evidences for the even bigger questiones such as the evolution of Bilateria and if it is evolved by miniaturization or progenesis, which means from simple to complex or vice versa or even both (Struck, Golombek *et al.*, 2015; Struck, Westheide *et al.*, 2002).

All of these make Polychaeta a very interesting group of organisms to study. In this thesis, we use bioinformatics as a tool to generate mitochondrial genomes/genes for our Polychaete phylogeny study.

2.2 Bioinformatics Pipeline

In this thesis, 36 polychaete species are sequenced (not by me), only 3 of them have been reported for achieving completed or nearly completed mitochondrial genomes so far (Weigert, Golombek *et al.*, 2016; Zhong, Struck *et al.*, 2008). Furthermore, most of the available COI sequences are quite short between 312bp - 699bp. In addition, 13 species have no records for mitochondrial sequences data in NCBI at all. Details and current state of knowledge showed in Table 1.

Therefore, one of the purposes of this thesis is to set up a bioinformatics pipeline to reconstruct as many (nearly) completed mitochondrial genomes as possible and recover as many genes as possible from our 36 sample datasets to reconstruct phylogenetic trees for the purposes of our phylogenomic study.

Table 1 Overview of current available mitochondrial sequences data in NCBI of our sample species and the current available corresponding literatures.

ID	Species	COI length (bp)	Mitochondrial Genome (bp)	Literatures
1	*Spirobranchus triqueter*			
2	*Pholoe baltica*	660		
3	*Nereimyra punctata*	662		
4	*Melinna cristata*	692		
10	*Neoleanira tetragona*	508		
11	*Nereiphylla lutea*	603		
12	*Phyllodoce groenlandica*	660		
13	*Polycirrus medusa*	662		
14	*Euclymene droebachiensis*			
15	*Chirimia biceps*			
17	*Scoloplos armiger*	660		
18	*Paramphinome jeffreysii*	1282		
19	*Pista cristata*	689	15,894	(Zhong *et al*., 2008)
20	*Nephtys paradoxa*	312		
21	*Eupolymnia nesidensis*			
22	*Microclymene acirrata*			
23	*Polycirrus plumosus*			
24	*Aricidea catherinae*	620		
26	*Aurospio banyulensis*			
27	*Streblosoma intestinale*			
28	*Goniada maculata*	658		
29	*Lumbriclymene cylindricauda*			
32	*Melinna albicincta*	657		
33	*Phyllodoce cf. groenlandica*	627		
34	*Aphrodita aculeata*	601		
35	*Melinna cristata elisabethae*			
36	*Nothria conchylega*	660		
37	*Hyalinoecia tubicola*	699		
38	*Sphaerodorum gracilis*			
39	*Nephtys longosetosa*	620		
40	*Eclysippe vanelli*	1,518	13,749	(Zhong *et al*., 2008)
41	*Owenia fusiformis*	657	16,204	(Weigert *et al*., 2016)
42	*Gattyana cirrhosa*	660		

43	*Spiophanes kroyer*	
44	*Amphictene auricoma*	
47	*Hydroides norvegica*	381

Animal mitochondrial genomes usually contain 37 genes, including 13 protein-coding genes (CDS), 22 tRNAs and 2 rRNAs, with an average length of around 16,000 nucleotides (J. L. Boore, 1999; Wolstenholme, 1992).

In this thesis, several bioinformatics tools are used for the purpose of mitochondrial genome assembly, annotation, and phylogenetic tree reconstruction. The most important 5 tools are introduced here.

2.2.1 NOVOPlasty

NOVOPlasty is a *de novo* assembler, which designed for organelle genome assembly from whole genome sequencing (WGS) data. It is a seed-based tool using seed-and-extend algorithm. The seed sequence can be very "flexible", from related to more distantly related species; from a single read, a conserved gene to a complete organelle genome sequence. The algorithm uses the given seed to retrieve a sequence read from the dataset and subsequently extend it until the targeted organelle genome is circularized (Dierckxsens, Mardulyn *et al*., 2017).

NOVOPlasty is quite easy to use, their paper also claimed it as a quicker assembler with less demand of computer memory (maximum 16GB) and a high accuracy of over 99.99% (Dierckxsens *et al*., 2017). Therefore, NOVOPlasty is chosen as our first assembler.

2.2.2 MitoZ

MitoZ is a "one-click solution" tool, which uses whole genome shotgun sequencing reads as raw data input, through different independent modules to final mitochondrial genome visualization output. These independent modules include "All", "All 2", "Filter", "Assemble", "Annotate" and "Visualize" (Meng, Li *et al.*, 2019). In this thesis, "All" and "All 2" modules

are used to assembly, annotate and visualize mitochondrial genome, while "Annotate" module is used for the contigs that generated from both NOVOPlasty and MITObim.

MitoZ conducts *de novo* assembly by using *de Bruijn* graph (DBG) algorithm, it does not request seeds or reference sequences from closely related species, this "solves" the dilemma when there are no related sequence information available on database or no previous knowledge was known (Meng *et al*., 2019).

The workflow of MitoZ (Meng *et al*., 2019) starts with quality control of raw data by filtering out 1) reads with many 'N's, 2) low quality reads or 3) PCR duplication reads. After that, it performs *de novo* assembly, the *de Bruijn* graph (DBG) algorithm conducts 1) contig assembly and 2) scaffold construction. The assembly outputs have both mitochondrial and nuclear genome sequences. Therefore, the next step is to use a profile Hidden Markov Model (profile HMM) (Durbin, Eddy *et al*., 1998; Krogh, Brown *et al*., 1994) based method to filter out the target mitochondrial genome sequences. Then mitochondrial genome annotation is performed, protein coding genes annotation using an in-house Perl script. By the time their paper has been published, the MitoZ version included protein database of both "Chordate and Arthropods". However, on their Github website, they have claimed, "Actually built some HMM profiles for other clades in the latest version of MitoZ" (linzhi2013, 2019a) and "Annelida-segmented-worms" is one of them. Therefore, in this thesis, we used "Annelida" setting for annotation. Transfer RNA (tRNA) is annotated by MiTFi (Juhling, Putz *et al*., 2012), a covariance model (CM) (Durbin *et al*., 1998) based method. rRNAs (12S rRNA and 16S rRNA) are annotated using infernal-1.1.1 (Nawrocki & Eddy, 2013). After annotation, conducting second quality control to remove potential false positive mitochondrial scaffolds, such as NUMTs and contaminations. Finally, employ Circos (Krzywinski, Schein *et al*., 2009) for mitochondrial genome visualization.

MitoZ is quite convenient and not difficult to use. However, it requests around 100 GB computer memories when using thread_number 16. Their paper claimed 97.33% of the test samples' PCG and rRNA could be recovered by MitoZ, and compared to their sanger sequenced mitochondrial genomes, the recovered genes are of high similarity (≥ 97%) (Meng *et al*., 2019). Therefore, MitoZ is chosen as our second assembler.

2.2.3 MITObim

MITObim is one of the most frequently used tools for mitochondrial genome assembly compared to the previous 2 tools. It is designed for novel mitochondrial genome assembly of non-model organisms from total genomic DNA derived NGS reads (Hahn, Bachmann *et al*., 2013).

MITObim is a seeds and reference sequences based tool, which has 3 different modes: “T1”, “T2” and “T3”. Among these, “T1” and “T2” mode request reference sequences while “T3” mode requests seed sequences. It also has an option of *de novo* assembly. However, we do not consider our sample datasets as "well behaved" datasets (Hahn *et al*., 2013). Therefore, only the 3 modes: “T1”, “T2” and “T3” are performed in this thesis.

“T1” mode is a two-step procedure, which requests a starting reference sequence. The starting reference sequence can be (distantly) related mitochondrial reference genome. Then use a MIRA mapping assembly (Chevreux, Wetter *et al.,* 1999) as a first step. While “T2” mode is a single-step procedure without the initial mapping assembly step, a quick option. It still requests the starting reference sequence that can be (not too distantly) related mitochondrial reference genome. In addition, “T3” mode requests seed sequences. The seed sequence can be the commonly used cytochrome-oxidase subunit 1 (COI), as a starting reference (Hahn *et al*., 2013).

The workflow of MITObim: the mitochondrial reads in our sample datasets are first mapped to this (distantly) related mitochondrial reference genome using MIRA. The mapping result is a new reference sequence with many gaps in between since only the conserved regions are mapped. Then using this new reference sequence as a bait to fish the reads in our sample datasets that either partly or fully overlaps to it, map the subset of reads using MIRA again and thus extend this new reference sequence. MITObim stands for “mitochondrial baiting and iterative mapping”. Therefore, repeat baiting, mapping and extending, ideally until all the gaps are closed or the number of reads remains stationary. The final output is the novel mitochondrial genome (Hahn *et al*., 2013).

MITObim provides a new strategy of assembly by “reference mapping”, compared to the previous 2 tools. In addition, it is also easy to use. Their paper claimed it as superior to existing tools at that time and recovered mitochondrial genomes with an accuracy over 99.5% (Hahn *et al*., 2013). Therefore, MitoZ is chosen as our third assembler.

2.2.4 MITOS

MITOS stands for "MITOchondrial genome annotation Server", which is designed to compute a consistent *de novo* annotation of metazoan mitochondrial genomes (Bernt, Donath *et al*., 2013).

Protein coding genes annotation by aggregating BLASTX-based similarity searches with previously annotated protein sequences in the NCBI RefSeq 39 to identify each CDS. In other words, by conduct BLASTX, there are hits that matching candidate protein coding genes with previous annotated protein sequences in the database. In addition, correctly annotated genes are assumed to aggregate more hits than the erroneous ones at the same locations of the input mitochondrial genome. Therefore, the aggregation process serves as a quality control step to auto filter the previous problematic protein sequences in the database. For each CDS, the hits are aggregated separately to obtain "predictions". This allows for the detection of frame shifts, duplication events, and split genes (Bernt *et al*., 2013).

Transfer RNA (tRNA) is annotated by MiTFi (Juhling *et al*., 2012), a structure-based covariance model (Durbin *et al*., 1998). rRNAs (12S rRNA and 16S rRNA) are annotated using structure-based covariance models that similarly to the tRNA models.

MITOS is a web server, which does not require any prior bioinformatics knowledge, making it one of the easiest tools to use. Furthermore, according to their website, 509 studies have used MITOS so far (MITOSWebServer). Therefore, MITOS is chosen as a comparison approach to MitoZ.

2.2.5 ATGC Bioinformatics Platform

Phylogenetic tree can be made through different evolutionary models and tree-building methods. Here we choose to use SMS model (Lefort, Longueville *et al*., 2017): Smart Model Selection and both PhyML (Guindon & Gascuel, 2003) and PhyML 3.0 (Guindon, Dufayard *et al*., 2010) that based on the maximum-likelihood methods as comparison approach to make phylogenetic trees. Furthermore, Bootstrap (Felsenstein, 1985) 1000 replicates to assess the reliability of the phylogenetic trees.

PhyML is one of the most widely used tool (cited: >10,000) for phylogenetics (ATGC-Bioinformatics-Platform), it is a "simple hill-climbing algorithm that adjusts tree topology and branch lengths simultaneously" (Guindon & Gascuel, 2003). The starting tree is built by distance-based method using BIONJ (Gascuel, 1997). This starting tree is modified to improve its likelihood by simultaneous Nearest Neighbor Interchanges (NNIs) until it reaches an optimum. Moreover, this "simultaneous" strategy contributes to reduce the computing time. (Guindon & Gascuel, 2003)

PhyML 3.0 is a new algorithm using Subtree Pruning and Regrafting (SPR) topological moves, which relies on a parsimony-based filter (Guindon *et al*., 2010). Instead of a distance-based one which is used by PhyML (Guindon & Gascuel, 2003). The starting tree is still built by distance-based method using BIONJ (Gascuel, 1997). This starting tree is modified by using parsimony-based filter to "filter out the least promising topology modifications with respect to the likelihood function" then make the best SPR moves to maximize the likelihood (Guindon *et al*., 2010).

SMS model is fully integrated in PhyML. In this thesis, we use both settings for likelihood-based criteria: Akaike Information Criterion (AIC) (Akaike, 1973) and Bayesian Information Criterion (BIC) (Schwarz, 1978). For DNA sequence, there are 4 Rates Across Sites (RAS) options: +G, +G+I, +I, None. And 4 matrices: GTR (Lanave, Preparata *et al*., 1984), TN93 (Tamura & Nei, 1993), HKY85 (Hasegawa, Kishino *et al*., 1985), and K80 (Kimura, 1980). The basic principle is using a BIONJ tree topology, SMS selects the best RAS option for the best matrix (Lefort *et al*., 2017).

Bootstrap (Felsenstein, 1985) is the most commonly used methods to assess the reliability of the inferred phylogenetic trees. Here we use 1000 replication times to test the reliability of our final inferred phylogenetic trees.

ATGC Bioinformatics Platform is a web server, which also does not require any prior bioinformatics knowledge, making it another easy tool to use. Furthermore, it provides one-step solution: PhyML or PhyML 3.0 + SMS (AIC or BIC) + Bootstrap 1000 for reconstructing phylogenetic trees. Finally, according to their website, their tools are cited between 2500 and 3000 times each year (ATGC-Bioinformatics-Platform). Therefore, ATGC Bioinformatics Platform is chosen for reconstructing our phylogenetic trees.

2.3 Phylogenomic Study

In order to do our phylogenomic study, first of all, we need to have current state of knowledge. According to these two papers (Pamungkas *et al*., 2019; Weigert & Bleidorn, 2016), the big picture of polychaete worms phylogeny becomes slightly clear (compare to the big unknown out there, we still basically know nothing here).

Weigert *et al.* (Weigert & Bleidorn, 2016) suggests 1) based on morphology, Polychaeta consist of Palpata (Canalipalpata and Aciculata) and Scolecida (Rouse & Fauchald, 1997). In addition, Phyllodocida and Eunicida are within Aciculata while Terebellida, Sabellida and Spionida are within Canalipalpata. Furthermore, Annelida are split into Polychaeta and Clitellata. 2) Based on molecular data, Annelida comprise Errantia and Sedentaria and the six basal branching lineages Sipuncula, Amphinomida, Lobatocerebrum, Chaetopteridae, Magelonidae, and Oweniidae. Moreover, Errantia include Aciculata and Protodriliformia, while Sedentaria comprise Canalipalpata and Scolecida, as well as Clitellata.

Pamungkas *et al.* (Pamungkas *et al*., 2019) suggests as of year 2016, in total 11,456 polychaete species (1417 genera, 85 families) have been described on WoRMS. Of which, 6033 species belong to subclass Errantia, whereas 5085 belong to subclass Sedentaria. Additionally, 158 species belong to subclass Echiura and 180 species with unassigned subclass as Polychaeta incertae sedis. Six most studied polychaete families are Syllidae (993 species), Polynoidae (876 species), Nereididae (687 species), Spionidae (612 species), Terebellidae (607 species) and Serpulidae (576 species).

Therefore, we sort our 29 samples into 2 major parts based on subclass Errantia (Table 2) and Sedentaria (Table 3) that described on WoRMS. In addition, 7 samples are not included in below tables due to the cut-off line mentioned in Results section 5.3. Overall, most of our sample species are not from above mentioned 6 families.

In total 12 samples belong to subclass Errantia, and they come from 10 different families. Most of them have been studied or involved (means they are not the target species) in a phylogenetic/phylogenomic study before, except ID 20 - *Nephtys paradoxa* and ID 38 - *Sphaerodorum gracilis*.

Table 2 Overview of our sample species from subclass Errantia that described on WoRMS and some of the current available corresponding literatures (phylogenetic studies). IDs come from same family are highlighted in same colours.

ID	Species	Subclass	Order	Suborder	Family	Literatures
2	*Pholoe baltica*	Errantia	Phyllodocida	Aphroditiformia	Pholoidae	(Aguado & Bleidorn, 2010; Norlinder, Nygren *et al.*, 2012; Wiklund, Nygren *et al.*, 2005)
10	*Neoleanira tetragona*	Errantia	Phyllodocida	Aphroditiformia	Sigalionidae	(Aguado & Bleidorn, 2010; Norlinder *et al.*, 2012; Wiklund *et al.*, 2005)
34	*Aphrodita aculeata*	Errantia	Phyllodocida	Aphroditiformia	Aphroditidae	(Aguado & Bleidorn, 2010; Norlinder *et al.*, 2012; Wiklund *et al.*, 2005)
42	*Gattyana cirrhosa*	Errantia	Phyllodocida	Aphroditiformia	Polynoidae	(Norlinder *et al.*, 2012)
3	*Nereimyra punctata*	Errantia	Phyllodocida	Nereidiformia	Hesionidae	(Aguado & Bleidorn, 2010; Rousset, Pleijel *et al.*, 2007)
28	*Goniada maculata*	Errantia	Phyllodocida	Glyceriformia	Goniadidae	(Aguado & Bleidorn, 2010)
20	*Nephtys paradoxa*	Errantia	Phyllodocida	Phyllodocida incertae sedis	Nephtyidae	
39	*Nephtys longosetosa*	Errantia	Phyllodocida	Phyllodocida incertae sedis	Nephtyidae	(Aguado & Bleidorn, 2010; Struck, Nesnidal *et al.*, 2008)
38	*Sphaerodorum gracilis*	Errantia	Phyllodocida	Phyllodocida incertae sedis	Sphaerodoridae	
36	*Nothria*	Errantia	Eunicida		Onuphidae	(Budaeva,

	conchylega					Schepetov *et al.*, 2016)
37	*Hyalinoecia tubicola*	Errantia	Eunicida		Onuphidae	(Budaeva *et al.*, 2016; Rousset *et al.*, 2007; Struck *et al.*, 2008; Struck & Purschke, 2005; Struck *et al.*, 2002)
18	*Paramphinome jeffreysii*	Errantia	Amphinomida		Amphinomidae	(Lemer, Kawauchi *et al.*, 2015; Rousset *et al.*, 2007; Struck *et al.*, 2008)

On the other hand, 17 samples belong to subclass Sedentaria, and they come from 8 different families. Only 7 of them have been studied or involved in a phylogenetic/phylogenomic study before. The 10 samples that have not been studied based on molecular data before are: ID 1 - *Spirobranchus triquete*, ID 47 - *Hydroides norvegica*, ID 35 - *Melinna cristata elisabethae*, ID 32 - *Melinna albicincta*, ID 23 - *Polycirrus plumosu*, ID 27 - *Streblosoma intestinale*, ID 26 - *Aurospio banyulensis* , ID 43 - *Spiophanes kroyer*, ID 14 - *Euclymene droebachiensis* and ID 15 - *Chirimia biceps*.

Table 3 Overview of our sample species from subclass Sedentaria that described on WoRMS and some of the current available corresponding literatures (phylogenetic studies). IDs come from same family are highlighted in same colours.

ID	Species	Subclass	Infraclass	Order	Suborder	Family	Literature
1	*Spirobranchus triqueter*	Sedentaria	Canalipalpata	Sabellida		Serpulidae	
47	*Hydroides norvegica*	Sedentaria	Canalipalpata	Sabellida		Serpulidae	
4	*Melinna cristata*	Sedentaria	Canalipalpata	Terebellida	Terebellomorpha	Ampharetidae	(Glasby, Hutchings *et al.*, 2004; Rousset *et*

							al., 2007)
35	*Melinna cristata elisabethae*						
32	*Melinna albicincta*	Sedentaria	Canalipalpata	Terebellida	Terebellomorpha	Ampharetidae	
40	*Eclysippe vanelli*	Sedentaria	Canalipalpata	Terebellida	Terebellomorpha	Ampharetidae	(Weigert *et al*., 2016; Zhong *et al*., 2008)
19	*Pista cristata*	Sedentaria	Canalipalpata	Terebellida	Terebellomorpha	Terebellidae	(Patra, Kwon *et al*., 2016; Struck *et al*., 2008; Weigert *et al*., 2016; Zhong *et al*., 2008)
21	*Eupolymnia nesidensis*	Sedentaria	Canalipalpata	Terebellida	Terebellomorpha	Terebellidae	(Rousset *et al*., 2007)
23	*Polycirrus plumosus*	Sedentaria	Canalipalpata	Terebellida	Terebellomorpha	Terebellidae	
27	*Streblosoma intestinale*	Sedentaria	Canalipalpata	Terebellida	Terebellomorpha	Terebellidae	
26	*Aurospio banyulensis*	Sedentaria	Canalipalpata	Spionida	Spioniformia	Spionidae	
43	*Spiophanes kroyer*	Sedentaria	Canalipalpata	Spionida	Spioniformia	Spionidae	
14	*Euclymene droebachiensis*	Sedentaria	Scolecida			Maldanidae	
15	*Chirimia biceps*	Sedentaria	Scolecida			Maldanidae	
17	*Scoloplos armiger*	Sedentaria	Scolecida			Orbiniidae	(Bleidorn, Vogt *et al*., 2003; Struck & Purschke, 2005)
24	*Aricidea catherinae*	Sedentaria	Scolecida			Paraonidae	(Langeneck, Barbieri *et al*., 2019)
41	*Owenia fusiformis*	Polychaeta incertae sedis				Oweniidae	(Bleidorn *et al*., 2003; Kupriyanova & Rouse, 2008; Lehrke, ten Hove *et al*., 2007; Struck *et al*.,

	2008; Struck & Purschke, 2005; Weigert *et al*., 2016; Weigert, Helm *et al*., 2014)

The phylogeny of Annelida has been remained controversial since the recognition of Annelida as a taxon in the 19th century. The earliest molecular study used 18S rRNA sequences (Winnepenninckx, Backeljau *et al*., 1995), followed by using nuclear gene elongation factor -1α (Kojima, 1998; McHugh, 1997), three genes: histone H3, U2 snRNA and 28S rDNA (Brown, Rouse *et al*., 1999), 18S rDNA (Bleidorn *et al*., 2003; Hall, Hutchings *et al*., 2004), then combined of 18S rDNA, 28S rDNA and COI (Jördens, Struck *et al*., 2004), of COI and 18S rDNA were also used (Struck & Purschke, 2005). However, the results are not always encouraging or consistent. Recent years, phylogenomic studies were conducted (Struck *et al*., 2015; Weigert *et al*., 2014). It is said through phylogenomic studies, a well-supported and resolved annelid backbone tree becomes possible (Weigert & Bleidorn, 2016).

Therefore, one of the other purposes of this thesis is through different approaches: 1) single gene: COI, 2) multiple genes from partial mitochondrial genomes and 3) nearly completed mitochondrial genomes to reconstruct the phylogenetic trees of our 29 polychaete species for the study of Polychaete phylogeny.

3 Material

Nextseq shotgun sequencing data for 36 polychaete species, with adapters already removed. (In total, 40 datasets were provided (Fig 1), but according to the excel file that was provided, ID 30 and 31 were duplicated. On the other hand, datasets that were provided, ID 45 and 46 were not showed on the excel file. Later Blast results could not confirm which species they are. Therefore, ID 30, 31, 45 and 46 were excluded from this thesis. In total 36 polychaete species were studied). Details for read number (provided by others) and file size for each sample showed in below Table 4 and Fig 1, respectively.

Table 4 Read number for each sample. The maxim and minimum showed in red.

Work package	Sample name	Read number
WP3	polychaete 10	1,121,164
	polychaete 11	1,340,744
	polychaete 12	1,140,989
	polychaete 13	1,177,668
	polychaete 14	841,436
	polychaete 15	1,045,020
	polychaete 17	1,900,558
	polychaete 18	1,446,888
	polychaete 19	1,085,869
	polychaete 1	1,375,470
	polychaete 20	1,974,628
	polychaete 21	1,063,557
	polychaete 22	1,106,683
	polychaete 23	1,202,915
	polychaete 24	971,786
	polychaete 26	1,542,988
	polychaete 27	1,443,406
	polychaete 28	1,237,129
	polychaete 29	917,717
	polychaete 2'	1,380,007
	polychaete 32	982,897
	polychaete 33	1,090,680
	polychaete 34	973,894
	polychaete 35	1,888,051
	polychaete 36	952,738
	polychaete 37	1,363,922
	polychaete 38	1,020,537
	polychaete 39	972,473
	polychaete 3	1,043,702
	polychaete 40	1,022,948
	polychaete 41	1,771,041
	polychaete 42	1,358,602
	polychaete 43	887,847
	polychaete 44	974,953
	polychaete 47	1,174,112
	polychaete 4	1,202,794

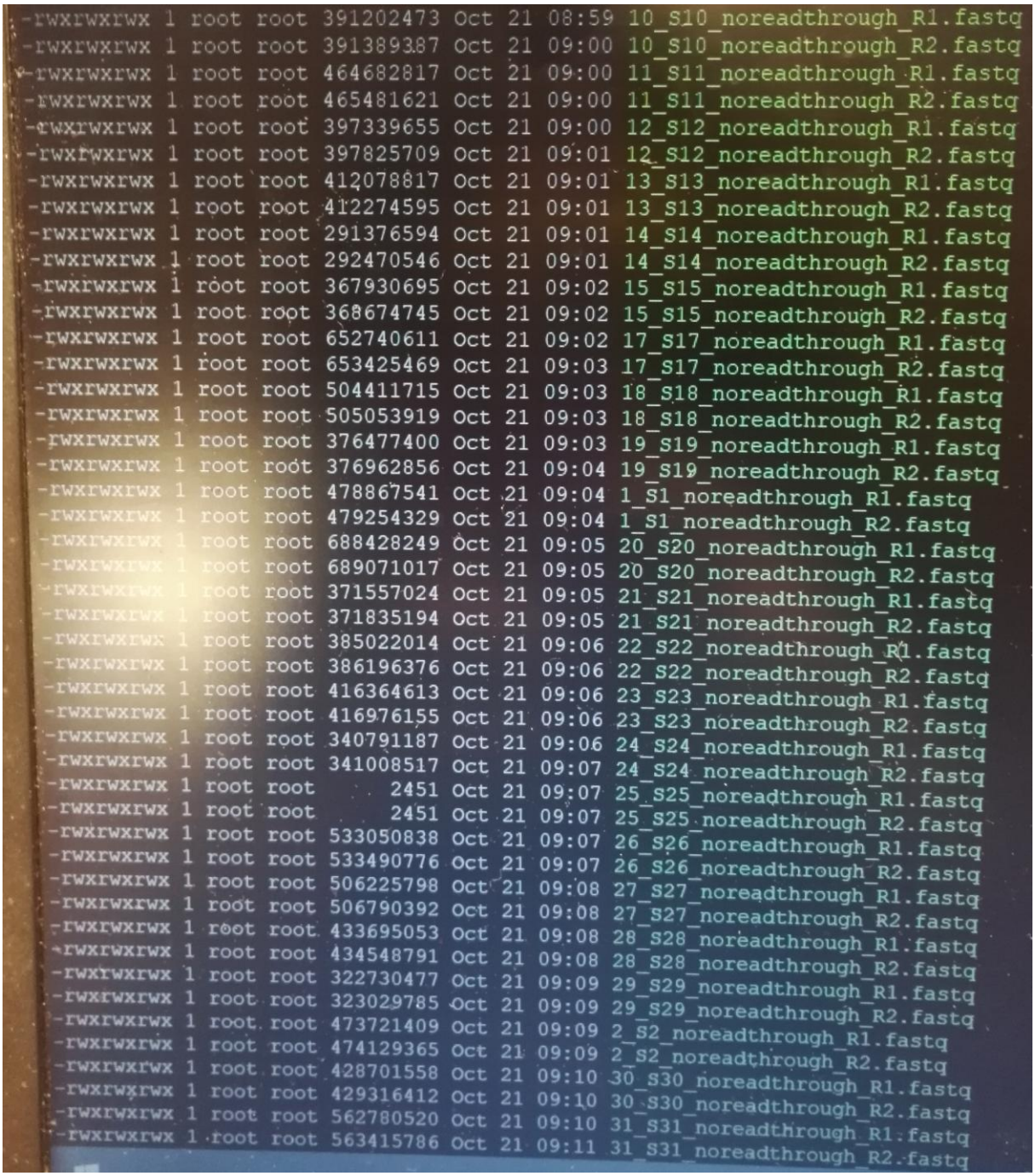


Fig 1 Overview of the datasets that were provided. File size for each sample.

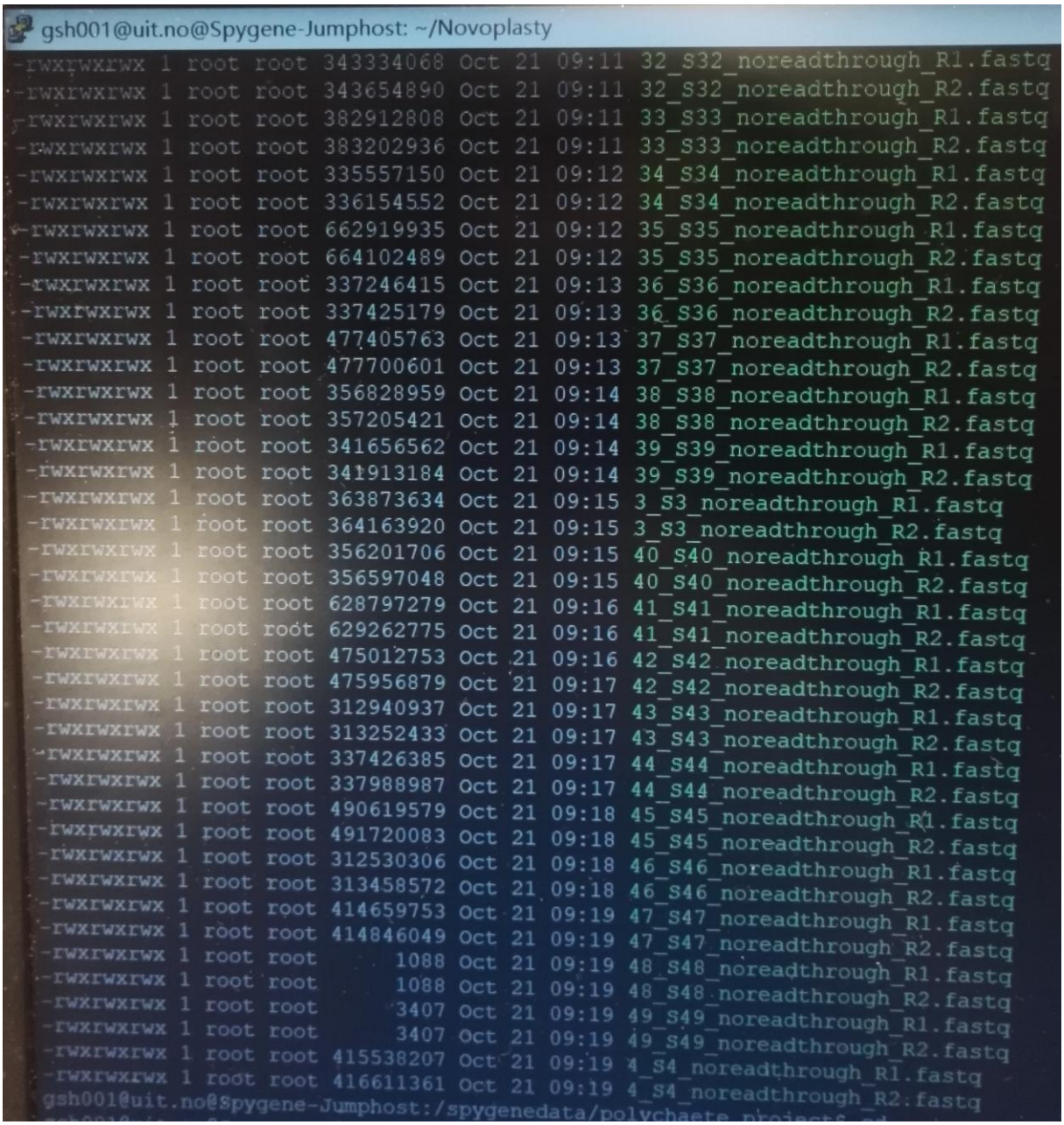


Fig 1-continue Overview of the datasets that were provided. File size for each sample.

4 Methods

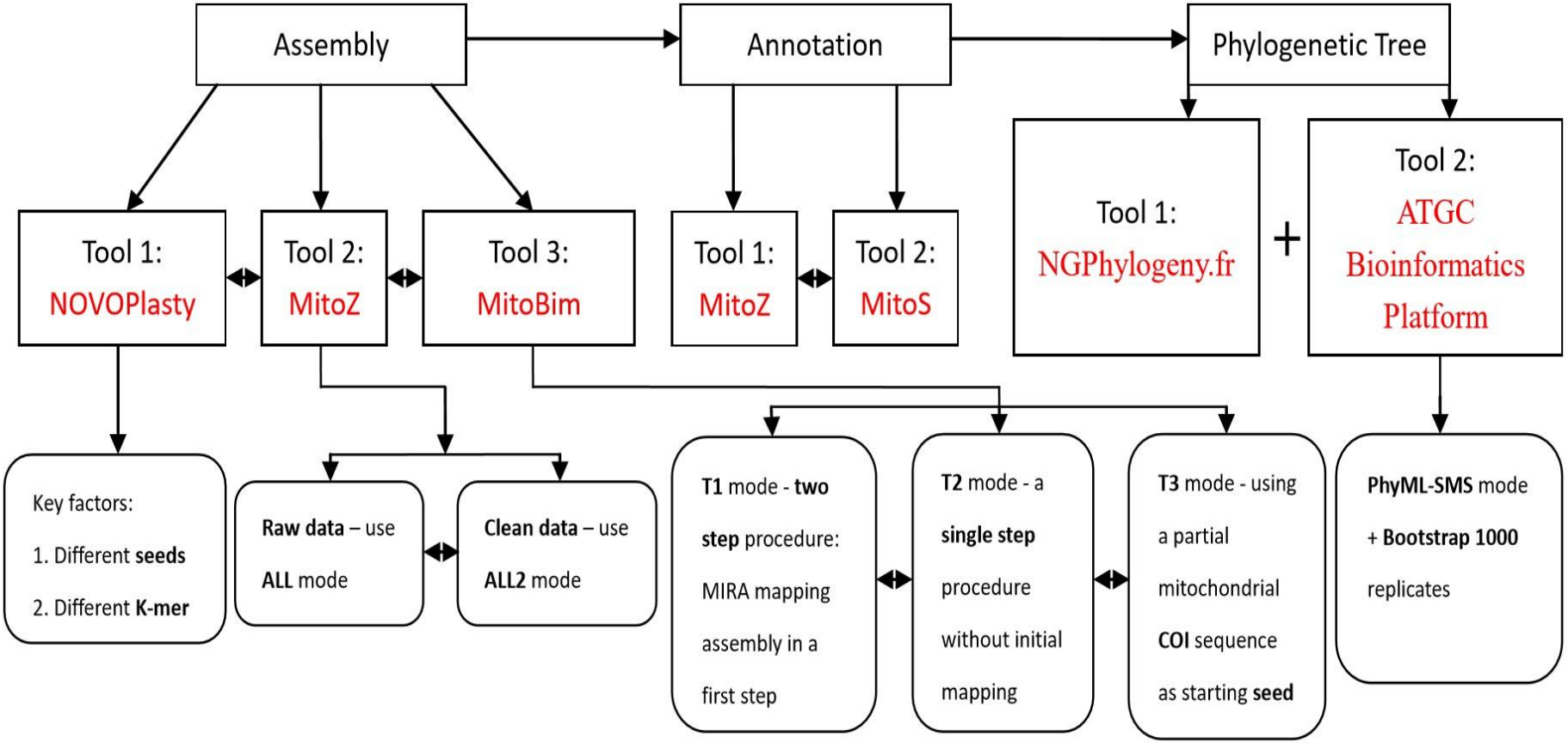


Fig 2 An overview of the bioinformatics pipeline used in this thesis: assembly, annotation and phylogenetic tree reconstruction; different tools were chosen and several modes were used.

4.1 Assembly

4.1.1 NOVOPlasty

NOVOPlasty (version 3.7) was the first tool to be chosen to perform the assembly for the mitochondrial genomes of our 36 polychaete species. In total 14 tests were run in order to optimize the results, details showed in Table 5. Assembly followed the instructions on website: https://github.com/ndierckx/NOVOPlasty. Useful scripts showed in Appendix A.1-A.3.

Tests were conducted by 1) using different mitochondrial sequence data as seeds: a) from as small as single COI to as large as all the available polychaete sequences in database: NCBI；b) from the mitochondrial sequences that available on database to the COI sequences that were generated from the assembly; 2) using different K-mers (from 23 to 39); 3) using two different databases (NCBI and BOLD).

Table 5 In total 14 tests were run using NOVOPlasty, details showed below.

	NOVOPlasty
Test 1	Seeds using **each ID corresponding COI** sequence which downloaded from NCBI
Test 2	Seed using a big dataset of COI sequence (371731mb) downloaded from NCBI, which combined **15 species**
Test 3	Start testing on 2 different **K-mer: 23 & 39**
Test 4	Seed using a big dataset of **16s & 12s rRNA** sequence which downloaded from NCBI
Test 5	Seed using a big dataset of **all rRNA** sequence downloaded from NCBI, which also included 18s rRNA etc.
Test 6	Seed using a super big dataset of **all the available polychaetes** sequence which downloaded from NCBI
Test 7	Seeds using each ID corresponding COI sequence which downloaded from 2 different database: **NCBI & BOLD**
Test 8	Seeds using each ID corresponding COI sequence which downloaded from NCBI test on **clean data** generated from Mitoz
Test 9	Run "**244188 possibilities**" with 57 seeds (30 from NCBI & 27 from BOLD), 7 reference sequences, 17 K-mer (23 – 39), for 36 samples.
Test 10	Use Seeds_everything to build a large database, blast against our samples, testing how many hits matching (COI, 16s rRNA sequences etc.) in our samples, then using the **highest COI hits** as seeds run **K-mer 23, 30 & 39**.
Test 11	Use each COI sequences in COI_for_tree.fasta as a single seed for each corresponding sample and run **K-mer 23-33 & 39**.
Test 12	Use the COI sequences that were cut-off due to too short as a single seed for each corresponding sample and run **K-mer 23-33 & 39**.
Test 13	Use COI_for_tree.fasta as a seed for the samples that have no results so far and run **K-mer 23, 30 & 39**.
Test 14	Use the **improved** results from Test 11-13 (**longer** COI sequences) as seeds to each corresponding samples and run **K-mer 23-33 & 39** again.

4.1.1.1 Assembly Strategy

Test 2, 4, 5 and 6 used big seeds instead of small seeds, this strategy based on mathematics "Probability issues", the more sequences were given, with the hope of the more chances of NOVOPlasty picking up these sequences to retrieve the sequences in our sample datasets. However, the results improved meanwhile computer-running time increased as well. Same effort was done for MITObim "T3" mode, only it turned out MITObim could not work if the seeds were too big.

Test 4 and 5 used rRNAs as seeds, especially Test 5 nearly every sample could be assembled. However, when later annotation, the results mainly showed rRNAs and tRNAs. Therefore, the following tests stopped using this strategy.

In early stage (Test 1-7), all the tests were performed by using raw data while in later stage (Test 8), clean data that generated by the second chosen tool: MitoZ were also tested. However, later annotation step showed that for our low coverage datasets, using clean data would cause 1) lose results (one example: ID 23); 2) lose genes (one example: ID 47), but it would not affect robust datasets like ID 2. Therefore, clean data were stopped using for the following tests.

Test 9 was the "The ultimate weapon" that on theory would run "244188 possibilities" (57 *7*17*36=244188). However, due to the computer running time constrain. For example, in general, small seeds (only 1 COI sequence) run <1 min per test if no results, while big seeds (several hundreds of different COI sequences for species like *Owenia fusiformis*) could take more than 1 hour per test if it can be assembled. On theory, it might take months to years to finish all the possibilities. Therefore, it was stopped right after the pattern was discovered. The key factors for assembly our sample datasets were 1) different seeds and 2) different K-mers. Reference sequences could also be an important factor if the datasets are very robust which could always be assembled, but this was not our case. Rule of thumb is the longer K-mers, the better quality the results are. However, due to the nature of our sample datasets, short K-mer was preferred. Otherwise, there would be no results at all for NOVOPlasty and this strategy was applied in the following tests as well.

After testing with both NOVOPlasty and MitoZ, Test 10 was conducted to figure out the reason why it showed so many "low coverage" and "invalid seeds" for NOVOPlasty and "low abundance" for MitoZ. The strategy here was to build a large local polychaete

mitochondrial sequences database and blast our each sample datasets against it, and then used accession number that belongs to each corresponding COI or rRNAs in NCBI to calculate how many COI or rRNAs hits that matched in our each sample datasets. By doing this to test whether our sample datasets were enriched with mitochondrial sequences especially COI and rRNAs or not. The seed that included all the available polychaete mitochondrial sequences in NCBI called "Seeds_everything" was used to build this large database. After that, blast against each of our sample datasets, and then count how many hits matched by using accession number. Finally, the accession number that got the highest COI hits, its corresponding COI sequence was used as a seed and run NOVOPlasty again on 3 different K-mers. However, the outcome did not improve much.

The final four tests (Test 11-14) were conducted after Methods section 4.3.1, a file with all the chosen longest identical COI sequences for each sample called "26_COI_for_tree.fasta" was generated. The strategy here was to do the final tests for all the samples with the seeds that each came from its own sample datasets. By doing this, the outcome for NOVOPlasty was finally improved compared to the previous tests. However, the overall outcome for assembly did not improve much.

4.1.2 MitoZ

MitoZ (version_2.4-alpha) was the second tool to be chosen to perform the same task. In total 8 tests were run in order to optimize the results, details showed in Table 6. Assembly followed the instructions on website: https://github.com/linzhi2013/MitoZ. Useful scripts showed in Appendix A.4-A.7.

Tests were conducted by 1) using different setting (default and Annelida); 2) using different mode (all, all2 and multi-Kmer); 3) using different K-mers (from 21 to 39); 4) using different min_abundance (10, 5, 1 and 0).

Table 6 In total 8 tests were run using MitoZ, details showed below.

	MitoZ
Test 1	Test on default setting with **all** mode, min_abundance = **10 & 5**
Test 2	Test on default setting with **all2** mode, min_abundance = **1 & 0**
Test 3	Test on **Annelida** setting with **all** mode, min_abundance = **10 & 5**
Test 4	Test on **Annelida** setting with **all2** mode, min_abundance = **1 & 0**
Test 5	Test on **Annelida** setting with **all** mode, **K-mer: 23 & 39**, min_abundance = **5 & 1**
Test 6	Test on **Annelida** setting with **all** mode, **K-mer: 22**, min_abundance = **1**
Test 7	Test on **Annelida** setting with **Multi-Kmer** mode for ID 1 & 47
Test 8	Test on **Annelida** setting with **all** mode, **K-mer: 21 - 39**, min_abundance = **1**

4.1.2.1 Assembly Strategy

MitoZ uses a non-"seeds and reference sequences" algorithm. Therefore, the only two variables that seem important here are k-mer and min_abundance. "All" module used raw data as input while "All 2" used clean data, the raw data filtering as quality control step was time-consuming. Therefore, in order to test whether min_abundance was a key factor here or not, Test 2 and 4 using clean data with "All 2" module aimed to save some computering time. However, it turned out that changing different min_abundance did not produce much useful outcomes. On the other hand, previously what we had learned from NOVOPlasty that k-mer was one of the key factors. Therefore, the only strategy left here was to change different k-mer. By using this strategy, some optimal outcomes were generated.

4.1.3 MITObim

MITObim (version 1.9.1) was the third chosen tool to continue the assembly. In total 14 tests were run in order to optimize the results, details showed in Table 7. Assembly followed the instructions on website: https://github.com/chrishah/MITObim. Useful scripts showed in Appendix A.8-A.11.

Tests were conducted by 1) using different mode (T1, T2 and T3). 2) Using different reference sequences: partial or completed mitochondrial genome sequences that downloaded from NCBI, as well as the results that generated by either NOVOPlasty or MitoZ, the nearly

completed mitochondrial genome sequences. 3) Using different mitochondrial COI sequence data as seeds: a) using own corresponding COI sequences downloaded from NCBI or BOLD; b) using the COI sequences that were generated from the assembly.

Table 7 In total 14 tests were run using MITObim, details showed below.

	MitoBim
Test 1	Test on **T2** quick mode, mismatch **20**, using **13** different **reference sequences** which downloaded from NCBI
Test 2	Test on **T3** mode, mismatch **20**, using **own COI** sequence downloaded from NCBI or BOLD as seeds
Test 3	Test on **T1** two steps (mapping first) mode, mismatch **20**, using **7** different **reference sequences** which downloaded from NCBI for **7** samples
Test 4	Test on **T2** quick mode, mismatch **20**, using **ID 2, 10, 24, 26** as reference sequences
Test 5	Use Seeds_everything to build a large database, blast against our samples, testing how many hits matching (COI, 16s rRNA sequences etc.) in our samples, then using the **highest COI hits** as seeds run **T3** mode, mismatch **20**
Test 6	Test on **T1** two steps (mapping first) mode, mismatch **20**, using **7** different **reference sequences** which downloaded from NCBI for **all** the samples based on their different "**order**"
Test 7	Run **T1** two steps (mapping first) mode for all the samples that have results on **Test 1 & 4** (using **T2** quick mode), mismatching **20**
Test 8	Test on **T1** two steps (mapping first) mode, mismatch **20**, using **2 new reference sequences** which downloaded from NCBI for the samples that have no results so far only
Test 9	Test on **T1** two steps (mapping first) mode, mismatch **20**, using **ID 1, 17, 20, 34, 39, 40, 41 & 47** as reference sequences for the samples that have no results so far only
Test 10	Test on **T1** two steps (mapping first) mode, mismatch **20**, using **1 new reference sequences** which downloaded from NCBI for the samples that have no results so far only
Test 11	Use each COI sequences in COI_for_tree.fasta as a single seed for each corresponding sample and run **T3** mode, mismatch **20**.
Test 12	Use the COI sequences that were cut-off due to too short as a single seed for each corresponding sample and run **T3** mode, mismatch **20**.
Test 13	Use COI_for_tree.fasta as a seed for the samples that have no results so far and run **T3** mode, mismatch **20**.
Test 14	Use the **improved** results from Test 11-13 (**longer** COI sequences) as seeds to each corresponding samples and run **T3** mode, mismatch **20** again.

4.1.3.1 Assembly Strategy

Test 5 was conducted at the same time as Test 10 in NOVOPlasty using the same approach. The highest COI hits were chosen as seeds and run MITObim on T3 mode. Only this time, MITObim showed great power to capture the COI sequences in our sample datasets once the "correct" single COI seeds were given. Therefore, the outcomes were finally improved.

Same improvements could also be seen by using T1 two steps (mapping first) mode. More nearly completed mitochondrial genome sequences were generated in this way. The overall strategy here was to test using as many reference sequences as possible with T2 quick mode (Test 1 & 4), then narrow down to look into the ones with outcomes by using T1 two steps mode (Test 7). Test 8-10 focused on the samples that either had no results or very poor results. However, the outcome did not improve much.

The final four tests (Test 11-14) were conducted at the same time as Test 11-14 in NOVOPlasty using the same strategy. Only this time, the overall outcome for assembly did have some improvement.

4.2 Annotation

4.2.1 MitoZ

MitoZ was chosen to perform the annotation for the contigs that generated from both NOVOPlasty and MITObim. Moreover, MitoZ itself is a "one-click solution" thus self-annotated. "Annelida-segmented-worms" setting was used for annotation. Annotation followed the instructions on website: https://github.com/linzhi2013/MitoZ. Useful scripts showed in Appendix A.3, A.6 and A.12.

4.2.2 MITOS

MITOS (6b33f95) was chosen as a comparison approach to MitoZ. Upload the right fasta file to its web server and pick the right genetic code: 05-Invertebrate, wait for about 1.5 hours, then collect the results.

4.3 Phylogenetic Tree

4.3.1 Preparation - COI for Tree

All the generated contigs for each ID were compared with contig name first, the identical ones were considered as one contig. Then COI sequences were extracted from each different contigs by using bedtools (version: v2.26.0) based on “start and end” position of “COX1” in a file named “summary.txt” that generated by MitoZ. However, due to computer counted position starting from “0” instead of “1”. Therefore, the positions for extracting COI sequences were adjusted to “n-1” for both start and end. The extracted COI sequence for each ID from different assembly tools but same annotation tool were compared with each other by using MEGA X (version 10.0.5) default setting to perform the pairwise distance analyses. Due to the nature of our sample datasets, the longest identical COI sequences were chosen for reconstructing the phylogenetic tree.

All the chosen COI sequences were adjusted to the same direction to “+”. For example, if the original sequence was generated in the direction of “+” then the forward strand would be picked, while if the original sequence was generated in the direction of “-” then the reverse strand would be picked.

In the end, all these COI sequences were saved in one fasta file named “26_COI_for_tree.fasta”, ready for reconstructing the phylogenetic tree based on COI sequences.

4.3.2 Preparation – Mitochondrial Genome for Tree

The methods applied to the above COI sequences were also used for mitochondrial genome sequences, but without the COI sequences extracting step. Due to the nature of our sample datasets, the longest identical mitochondrial genome sequences were chosen for reconstructing the phylogenetic tree.

All the chosen mitochondrial genome sequences were also adjusted to the same direction to “+”. Seqtk (Version: 1.2-r94) was used to get the reverse complement sequences if the original sequence was generated in the direction of “-”.

In the end, all these mitochondrial genome sequences were saved in two fasta files named "12_Genome_for_tree.fasta" and "21_Genome_for_tree.fasta", together with out-group: Katharina tunicata mitochondrion, complete genome (U09810.1), ready for reconstructing the phylogenetic trees based on mitochondrial genome sequences.

4.3.3 ATGC Bioinformatics Platform

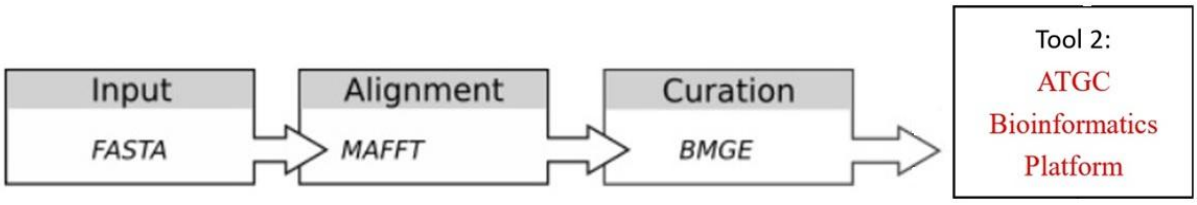


Fig 3 Workflow of NGPhylogeny.fr process the alignment and curation step while ATGC Bioinformatics Platform reconstruct the phylogenetic tree: PhyML or PhyML3.0, SMS (AIC or BIC) and Bootstrap 1000 replicates to test the reliability of the inferred phylogenetic trees.

Three fasta files "26_COI_for_tree.fasta", "12_Genome_for_tree.fasta" and "21_Genome_for_tree.fasta" were uploaded to NGPhylogeny.fr to process the alignment and curation step. Alighment was performed by MAFFT (version 7.407_1) (Katoh & Standley, 2013) and curation was conducted by BMGE (version 1.12_1) (Criscuolo & Gribaldo, 2010). After that, the 3 output BMGE Cleaned sequences Phylip (file type: PHYLIP) were then uploaded to ATGC Bioinformatics Platform, to reconstruct the phylogenetic tree using 4 different methods and models. For maximum-likelihood methods: PhyML for NNI algorithm and PhyML3.0 for SPR algorithm. For SMS model, both AIC and BIC criterions were tested. Bootstrap 1000 replicates were chosen to assess the reliability of the inferred phylogenetic trees for all the tests. Table 8 showed details about each tests and the corresponding hours that took.

The generated phylogenetic trees were subsequently improved to be more trustworthy (higher bootstrap value) by removing certain species from the clade that had lowest bootstrap value.

Table 8 Details about each tests and the corresponding hours that took.

Bootstrap 1000 for all	PhyML-SMS-AIC	PhyML -SMS-BIC	PhyML3.0-SMS-AIC	PhyML3.0-SMS-BIC
26_COI_for_tree.fasta	14 hour(s), 1 minute(s)	14 hour(s), 3 minute(s)	14 hour(s), 3 minute(s)	14 hour(s), 4 minute(s)
12_Genome_for_tree.fasta	1 day(s), 1 hour(s)	1 day(s), 1 hour(s)	1 day(s), 1 hour(s)	1 day(s), 0 hour(s)
21_Genome_for_tree.fasta	4 day(s), 4 hour(s)	4 day(s), 3 hour(s)	4 day(s), 4 hour(s)	4 day(s), 5 hour(s)

4.4 Useful Scripts

Besides the web server ones, all of the above-mentioned tests were conducted by using virtual machine, some useful scripts can be found in Appendix A.1 - A.12.

The overall principles are: 1) Run test datasets manually first; 2) Run single test with our sample datasets manually; 3) Create shell scripts to run tests in bulk; 4) Test the scripts with 2-3 our samples; 5) Run tests with all our samples in bulk; 6) Always manual check the results.

5 Results

5.1 The Nature of Our Sample Datasets

The first half part of Test 10 in NOVOPlasty and Test 5 in MITObim were conducted to figure out whether our sample datasets were enriched with mitochondrial sequences, especially COI and rRNAs sequences or not. Results showed in below Table 9. All the recorded numbers here were the maxim number. For example, ID 41 *Owenia fusiformis* have around 300 COI sequences in NCBI, which means 300 accession numbers were processed to get a maxim number of hits. The results showed 1090 COI hits of ID 2 as the highest amount,

followed by 148 hits of ID 47. Besides three 0 hits, the lowest number of hits was 2, almost 500 times less compared to ID 2. Most of the COI hits were less than 100 while ID 2 seemed like "coming from another world" with 1000+ hits. Furthermore, in most cases, 16s rRNA got higher number of hits than COI, which means more rRNAs than COI sequences in our sample datasets.

The results suggested our sample datasets were not enriched with mitochondrial sequences except ID 2 was a robust dataset.

Table 9 Results of first half part of Test 10 in NOVOPlasty and Test 5 in MITObim. "NA" means no COI sequence available in NCBI.

Blast results		Hits	
Data ID	COI	16s rRNA	12s rRNA
1	NA	NA	NA
2	1090	1034	
3	12	18	
4	70	74	
10	52	120	
11	12	20	
12	32	44	
13	2		
14	NA	NA	NA
15	NA	NA	NA
17	66	42	
18	68	78	
19	26	28	
20	38	86	
21	NA	NA	NA
22	NA	NA	NA
23	NA	NA	NA
24	0		
26	NA	NA	NA
27	NA	NA	NA
28	2	6	
29	NA	NA	NA
32	62	50	
33	0		
34	0	74	
35	NA	NA	NA
36	38	46	
37	56	76	
38	NA	NA	NA
39	6		
40	82	76	98
41	94	74	
42	42	44	
43	28	16	
44	NA	NA	NA
47	148		

5.2 Assembly

In total, 470 outputs that contained COI sequences were generated from 3 different assembly tools but same annotation tool. Some of the outputs contained multiple short contigs, but for simple calculation, here we consider 470 outputs equal to 470 contigs. Among these, run 14 tests for NOVOPlasty generated 231 contigs, 8 tests for MitoZ produced 83 contigs and 14 tests for MITObim obtained 156 contigs. Among all the samples, ID 2 was the most robust sample dataset that worked on all the 3 tools and thus generated 59 contigs being the highest amount. On the contrary, there were 4 samples, ID 13, 22, 33 and 44 did not work on any of the 3 tools thus with no results at all. Details for each ID showed below (Table 10), details of the contigs can be found in Appendix.

Table 10 Total number of outputs generated from 3 different assembly tools but same annotation tool for each tools as well as for each ID.

Data ID	Novoplasty	MitoZ	MitoBim	Total (All contigs)
1	11	22	1	34
2	27	28	4	59
3	0	0	7	7
4	15	0	7	22
10	8	3	3	14
11	0	0	7	7
12	0	0	9	9
13	0	0	0	0
14	0	0	3	3
15	0	0	7	7
17	3	0	4	7
18	4	0	4	8
19	0	0	9	9
20	6	0	6	12
21	23	0	7	30
22	0	0	0	0
23	25	0	4	29
24	11	6	1	18
26	12	12	2	26
27	5	0	4	9
28	0	0	5	5
29	0	0	3	3
32	0	0	6	6
33	0	0	0	0
34	11	0	4	15
35	28	0	6	34
36	1	0	4	5
37	11	0	3	14
38	0	0	3	3
39	0	0	7	7
40	0	0	7	12
41	5	0	5	5
42	6	0	6	12
43	0	0	5	5
44	0	0	0	0
47	19	12	3	34
Total	231	83	156	470

In total, NOVOPlasty generated results for 19 sample datasets, while MitoZ only gave results to 6 samples, and MITObim being the best tool here produced results for 32 samples (Table 11).

Table 11 Total number of results generated by each tool

Summary	NOVOPlasty	MitoZ	MitoBim
Sample Qty	19	6	32

5.3 Annotation

5.3.1 COI Sequence

In total, 32 novel COI sequences were generated out of 36 samples. Among these, 21 of them were in decent size around 1500bp, 4 of them around 1000bp and 1 of them was 3391bp, possible duplicated later confirmed by MITOS. In addition, 6 of them were too short around 196bp – 763bp thus cut off (summarized in Table 12).

Therefore, 26 novel COI sequences were ready for reconstructing the phylogenetic tree based on COI sequences. Their length showed in below Table 13.

Table 12 Summary table for generated COI sequences.

Summary	Qty
Decent size of COI (around 1500bp)	21
Around 1000bp	4
Too long, possible duplicated COI	1
Total	26
Cut off line	
Too short	6
Without results	4
Total	36

Table 13 Length of COI sequence for each ID.

Data ID	COI sequence length (bp)
1	1510
2	1549
3	1546
4	1559
10	1538
15	1099
17	1534
18	1105
19	1537
20	3391
21	1228
23	1558
24	1534
26	1537
27	1351
32	1565
34	1551
35	1559
36	1545
37	1546
38	1552
40	1526
41	1552
42	1537
43	913
47	1403
Cut off line	
11	763
12	620
13	–
14	421
22	–
28	400
29	292
33	–
39	196
44	–

5.3.2 Mitochondrial Genome Sequence

In total, 5 sample datasets: ID 2, 10, 20, 24, 26 had achieved the optimal results with 13 CDS, 22 tRNA and 2 rRNA, the nearly completed novel mitochondrial genomes with all the genes presented (Table 14). Among these samples, ID 2 and 24 were achieved by both NOVOPlasty and MitoZ. ID26 was achieved by both MtioZ and MITObim. ID 10 was achieved by MitoZ only and ID 20 by MITObim only. In another words, NOVOPlasty achieved ID 2 and 24. MitoZ achiteved ID 2, 10, 24 and 26. MITObim achieved ID 20 and 26.

Table 14 Optimal results - 5 samples generated by corresponding assembly tools.

Optimal results with 13 CDS, 22 tRNA and 2 rRNA.			
	NOVOPlasty	MitoZ	MitoBim
ID 2	√	√	
ID 10		√	
ID 20			√
ID 24	√	√	
ID 26		√	√

In total, 6 sample datasets: ID 1, 17, 34, 40, 41 and 47 had achieved nearly optimal results with either 11 or 13 genes, high number of tRNA and rRNA, the nearly completed novel mitochondrial genomes with most of the genes presented (Table 15). Among these samples, ID 1 was achieved by both NOVOPlasty and MitoZ. ID 17, 34, 40 and 41 were achieved by MITObim only and ID 47 by NOVOPlasty only. In another words, NOVOPlasty achieved ID 1 and 47. MitoZ achiteved ID 1. MITObim achieved 17, 34, 40 and 41.

Table 15 Nearly optimal results - 6 samples generated by corresponding assembly tools.

Nearly optimal results with either 11 or 13 genes, high number of tRNA and rRNA			
	NOVOPlasty	MitoZ	MitoBim
ID 1	√	√	
ID 17			√
ID 34			√
ID 40			√
ID 41			√
ID 47	√		

In summary, in total 35 out of 36 samples had at least 1 CDS. Among these, 5 of them had achieved the nearly completed novel mitochondrial genome with all the genes presented. 6 of them had either 11 or 13 CDS, high number of tRNA and rRNA. 1 of them had achieved 11 short CDS. 9 of them got partial mitochondrial genome less than 10 CDS but more than 5 CDS. 14 of them got less than 5 CDS thus cut off. In addition, ID 44 was the only sample yielded no result (summarized in Table 16).

Therefore, 12 nearly completed novel mitochondrial genome sequences were ready for reconstructing the phylogenetic tree based on mitochondrial genome sequences. Furthermore, 21 (12 nearly completed + 9 partial mitochondrial genome) were ready for reconstructing the phylogenetic tree based on multiple gene sequences. Details for each ID showed in Table 16.

Table 16 Summary table for annotated mitochondrial genome sequences information for each ID.

Data ID	CDS	tRNA	rRNA	Length (bp)	
1	11	24	2	15354	
2	13	22	2	15209	
10	13	22	2	14848	
17	13	15	1	12331	
20	13	22	2	17681	
24	13	22	2	16218	
26	13	22	2	15868	
34	13	20	2	14857	
39	11	13	1	17221	
40	13	17	2	14053	
41	13	20	2	16348	
47	11	23	2	11650 + 5092	
Cut off line for 12_Genome_for_tree.fasta					
Data ID	CDS	tRNA	rRNA	Length (bp)	Note
4	8	7	2	16357	mapping results
14	5	7	2	15537	mapping results
19	5	4	1	15970	mapping results
21	7	3	1	15908	mapping results
23	8	7	0	6308	
28	7	6	0	15366	mapping results
32	6	10	1	17473	mapping results
35	9	10	0	13747	mapping results
42	9	8	0	17132	mapping results
Cut off line for 21_Genome_for_tree.fasta					
3	3	1	0	15325	mapping results
11	2	0	0	1460	
12	2	3	0	14048	mapping results
12	2	1	2	17309	mapping results
13	2	6	1	16118	mapping results
15	3	3	1	16589	mapping results
18	3	6	0	4071	
22	4	3	1	15538	mapping results
23	6	16	2	9352	mapping results
27	4	11	2	15860	mapping results
29	1	0	1	14860	mapping results
33	2	0	0	14053	mapping results
36	4	4	0	3299	
37	3	2	0	3017	
38	2	0	0	16901	mapping results
43	1	0	0	933	
44	0	0	0	0	

5.3.3 MitoZ vs MITOS

The annotation results from MitoZ and MITOS were not exactly the same but similar. Except ID 1 showed one different missing gene and MITOS annotated one more gene for ID 47 compared to MitoZ. Table 17 showed a pattern of generally shorter COI sequences generated by MITOS than MitoZ, while the positions of COI sequences were within the similar region.

In addition, MITOS tended to annotate more tRNA but less rRNA than MitoZ. Finally, as for mitochondrial gene orders, MitoZ and MITOS showed the same results.

Table 17 Similar annotation results from MitoZ and MITOS for each sample datasets.

	MitoZ				MitoS			
	COI Position Length (bp)	CDS	tRNA	rRNA	COI Position Length (bp)	CDS	tRNA	rRNA
ID1	2280 | 3789 1510bp	11 Genes not found: atp6, atp8	24 duplicated genes: trnM, trnT	2	2382 | 3815 1434bp	11 Genes not found: atp8, nad4l	22 duplicated genes: trnM, trnT	2
ID2	3183 | 4717 1535bp	13	22	2	3183 | 4706 1524bp	13 duplicated genes: atp6	22	1
ID10	10586 | 12123 1538bp	13	22	2	10599 | 12122 1524bp	13	22	2
ID17	<3083 | 4616 1534bp	13	15	1	3083 | 4549 1467bp	13 duplicated genes: nad1	17	0
ID20	11520 | 14920 3391bp	13	22	2	11520 | 12218 699bp 14068 | 14889 822bp	13 duplicated genes: COX1	22	2
ID24	1940 | 3473 1534bp	13	22	2	1940 | 3451 1512bp	13	22	2
ID26	11071 | 12607 1537bp	13	22	2	11071 | 12594 1524bp	13 Split genes: cob, nad5	22	2
ID34	10549 | 12099 1551bp	13	20	2	10575 | 12098 1524bp	13	21	1
ID39	11481 | 11614 134bp	11	13	1	11481 | 11612 132bp	11	15	2
ID40	4780 | 6305 1526bp	13	17	2	4780 | 6297 1518bp	13	20	2
ID41	168 | 1701 1534bp	13	20	2	168 | 1193 1026bp	13 duplicated genes: cox3, atp6, nad1, nad5, nad6, trnK	18	2
ID47	8820 | 10222 1403bp	11 Genes not found: nd4l, atp8	23 duplicated genes: trnM	2	8793 | 10220 1428bp	12 Genes not found: atp8	22	2

5.4 Mitochondrial Genome Visualization & Gene Order

Table 18 Mitochondrial gene orders for the chosen 12 sample datasets. Conserved gene clusters are indicated by different colours.

Mitochondrial Gene Order																
ID2	ND1	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND5	ND4L	ND4	s-rRNA	l-rRNA
ID10	ND1	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND5	ND4L	ND4	s-rRNA	l-rRNA
ID17	ND1	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND5	ND4L	ND4		l-rRNA
ID20	ND1	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND5	ND4L	ND4	s-rRNA	l-rRNA
ID26	ND1	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND5	ND4L	ND4	s-rRNA	l-rRNA
ID34	ND1	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND5	ND4L	ND4	s-rRNA	l-rRNA
ID39	ND1	ND3	ND2	COX1	COX2	ATP8		ND6		ATP6	ND5	ND4L	ND4		l-rRNA
ID40	ND1	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND4L	ND4	ND5	s-rRNA	l-rRNA
ID24	ND3	ND2	COX1	COX2	ATP8	COX3	ND6	CYTB	ATP6	ND1	ND5	ND4L	ND4	s-rRNA	l-rRNA
ID41	ATP6	COX3	COX1	COX2	ATP8	ND3	ND5	ND2	ND1	ND6	CYTB	ND4L	ND4	s-rRNA	l-rRNA
ID1	ND6	ND3	ND4	ND5	COX3	COX2	ND1	COX1	CYTB	ND4L	ND2			s-rRNA	l-rRNA
ID47	ND1	COX1	ND5	ND4	CYTB	ATP6	ND2	ND6	COX3	ND3	COX1			s-rRNA	l-rRNA

The results showed the mitochondrial gene orders generated by 3 different assembly tools for each sample datasets were exactly the same. ID2 as one sample showed in Fig 4-6. (The rest of the generated mitochondrial genomes can be found in Appendix A.13-A.28.) However, this was not the case among different sample datasets. ID 2, 10, 17, 20, 26, 34 and 39 were in same gene orders, while ID 24 and 40 were slightly different as above-mentioned 7 sample datasets, each with one different gene in different order. Followed by ID 41 with seven genes in same order as above-mentioned 9 samples. Nevertheless, ID 1 and 47 were in very different gene orders compared to all the rest of sample datasets. Details showed in Table 18.

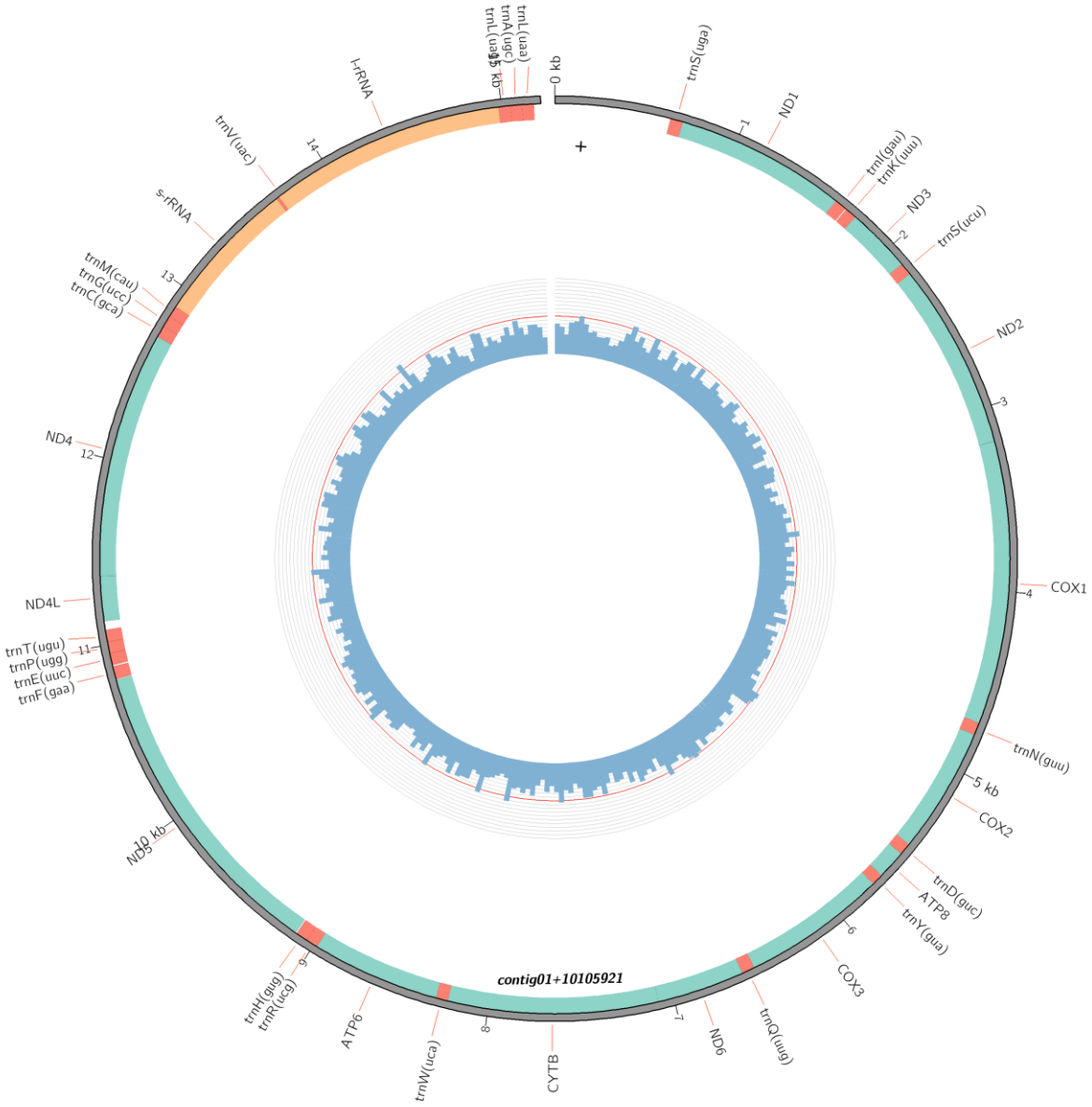


Fig 4 Nearly completed mitochondrial genome generated by NOVOPlasty for ID2. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.

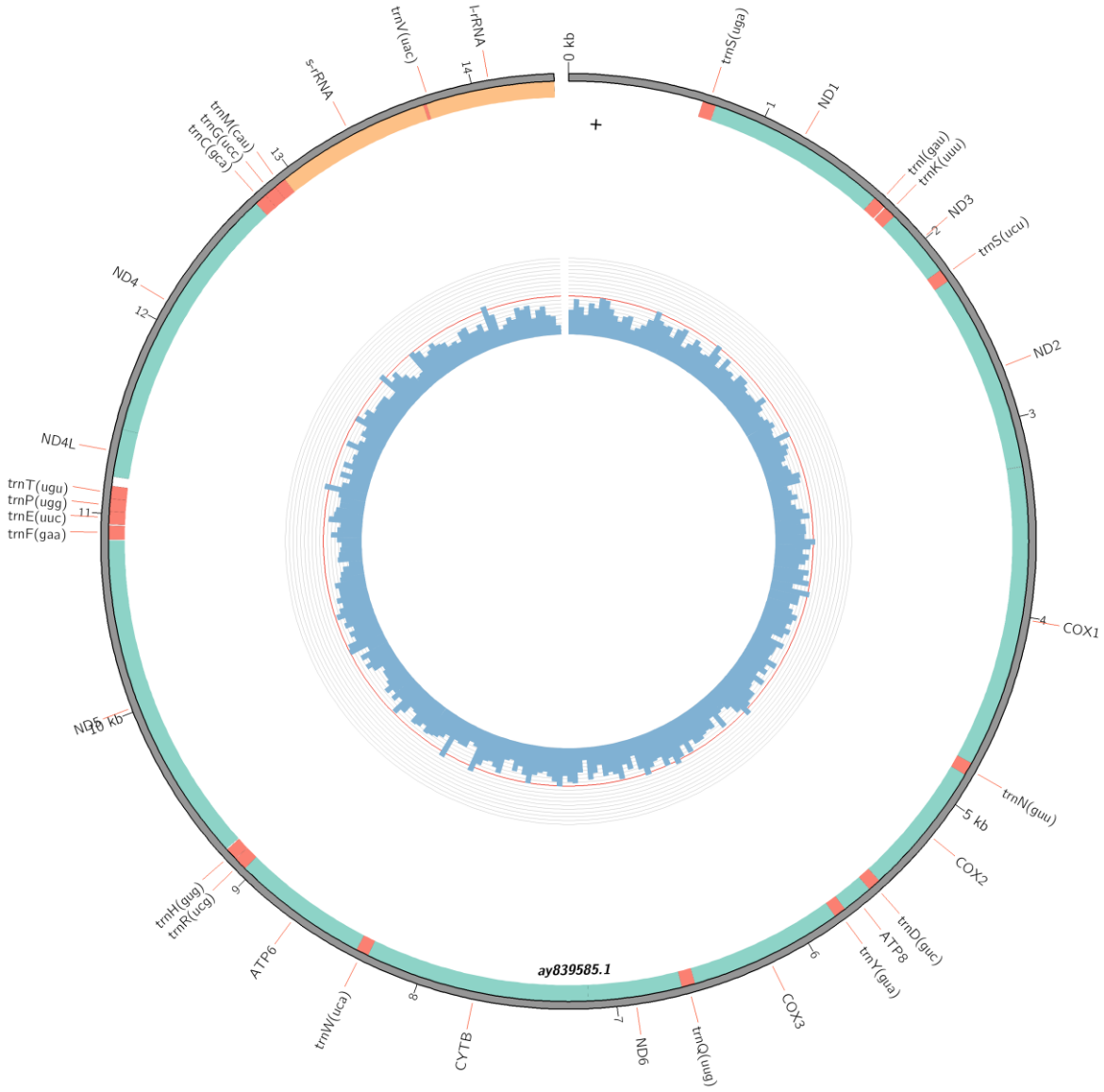


Fig 5 Nearly completed mitochondrial genome generated by MITObim for ID2. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.

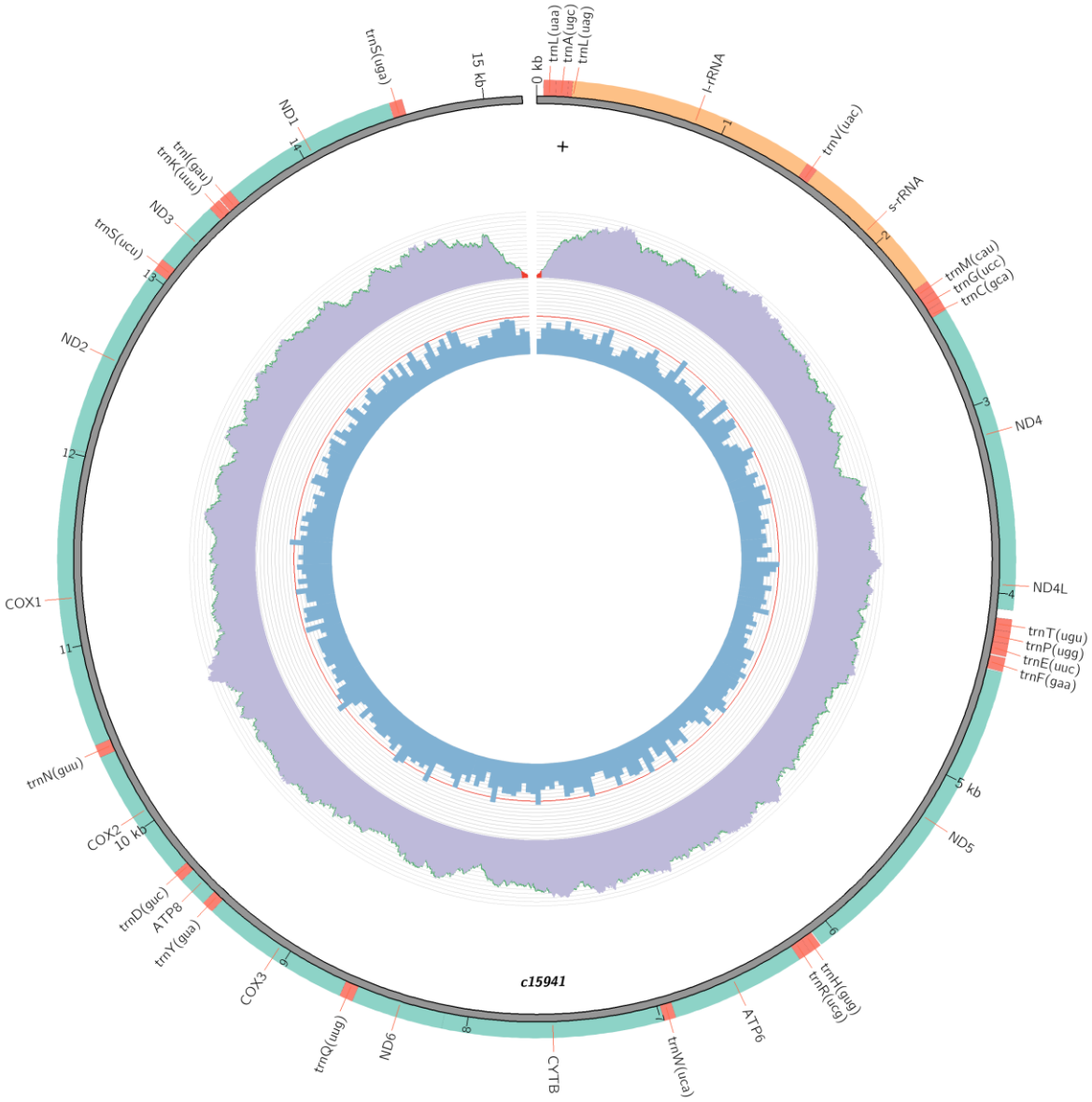


Fig 6 Nearly completed mitochondrial genome generated by MitoZ for ID2. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution with green color as the outline, depth lower than the minimum value (default 20) showed in red, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.

5.5 Phylogenetic Tree

More than hundreds of phylogenetic trees had been made in order to find the most trustworthy ones. Below phylogenetic trees were the ones with the highest minimum bootstrap values.

5.5.1 Based on Nearly Completed Mitochondrial Genome

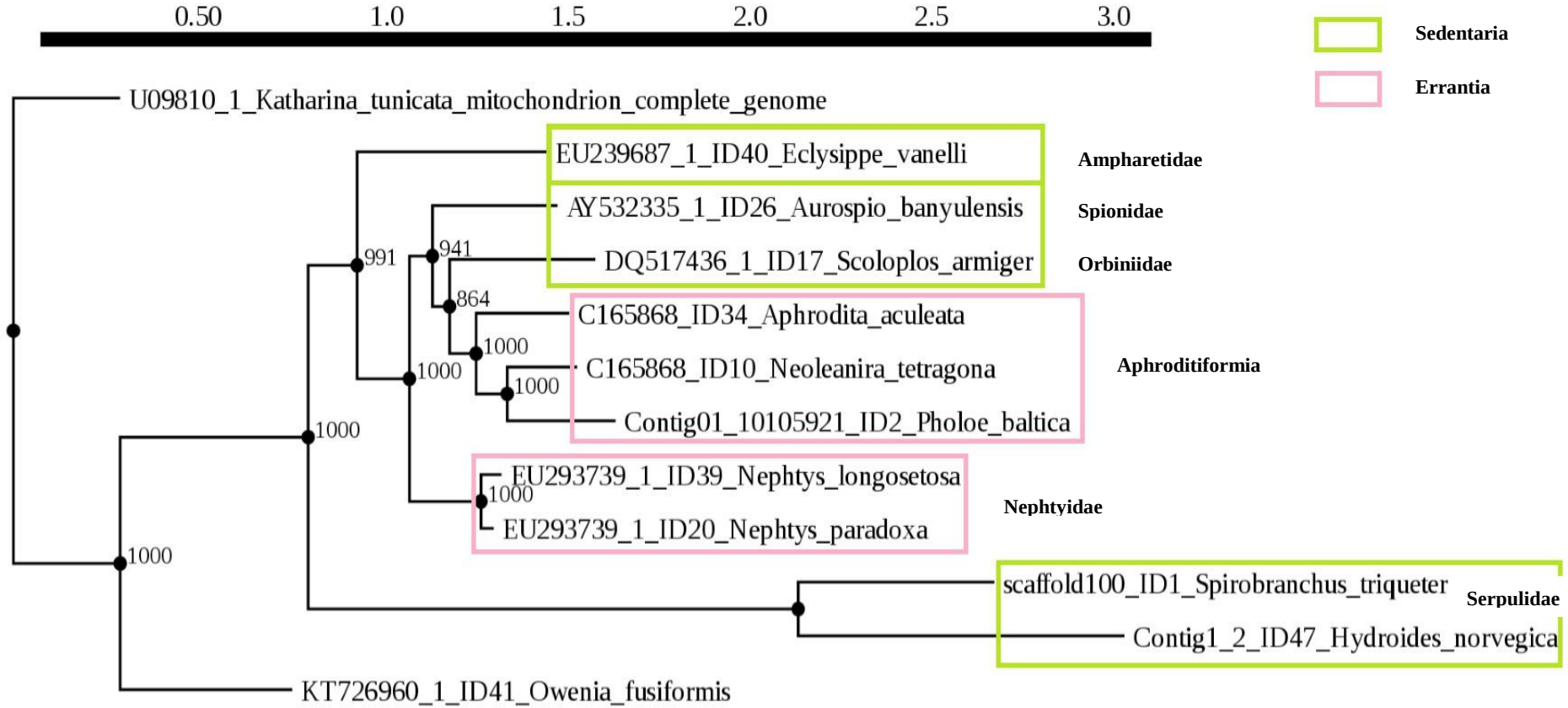


Fig 7 Phylogenetic tree inferred from 11 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML maximum-likelihood methods + BIC (Bayesian Information Criterion) Best model: GTR + G and Bootstrap 1000 replicates. The minimum Bootstrap value is 864.

The initial generated trees were unrooted, in order to find out whether ID 41 was the basal group or not, we introduced an out-group: Katharina tunicata mitochondrion, complete genome (U09810.1). The results (Fig 7) showed that besides out-group Katharina tunicate, ID 41 was the basal group here. The relationships in the basal part of the tree were robust with bootstrap value 1000.

The phylogenetic relationships of the samples that coming from the same family/suborder were quite consistent with the sorted tables (Table 2 and 3) that we presented in the Section 2.3. They all formed monophyletic groups. Furthermore, Spionidae, Orbiniidae and Aphroditiformia formed a monophyletic clade. Nephtyidae are sister to this clade. Ampharetidae are sister to the monophyletic clade formed by Nephtyidae and Spionidae, Orbiniidae and Aphroditiformia. Finally, Serpulidae are sister to the monophyletic clade formed by Ampharetidae and Nephtyidae, Spionidae, Orbiniidae and Aphroditiformia.

5.5.2 Based on Multiple Genes from Partial Mitochondrial Genome

Fig 8 showed that the phylogenetic relationships of the samples that coming from the same family/suborder were relatively consistent with the sorted tables (Table 2 and 3) that we presented in the Section 2.3. Most of them formed monophyletic groups. Except ID 40 *Eclysippe vanelli* from family Ampharetidae, was not along with its own family members, but closer to the basal part of the tree. Furthermore, 4 different families in Suborder Aphroditiformia: Aphroditidae, Polynoidae, Pholoidae, Sigalionidae together with family Maldanidae from Infraclass Scolecida formed a monophyletic clade A. In addition, 3 Terebellidae formed a monophyletic clade B. Moreover, 2 Ampharetidae are sisters to the monophyletic clade formed by 2 Nephtyidae and 1 Ampharetidae, together they formed a monophyletic clade C. Clade B and C formed a monophyletic clade and clade A is sister to this clade. Together, they formed a monophyletic clade D and sister to ID 40 thus form a monophyletic clade E. Finally, Serpulidae are sister to this clade E. ID 41 is the basal group while Katharina unicata is the outgroup.

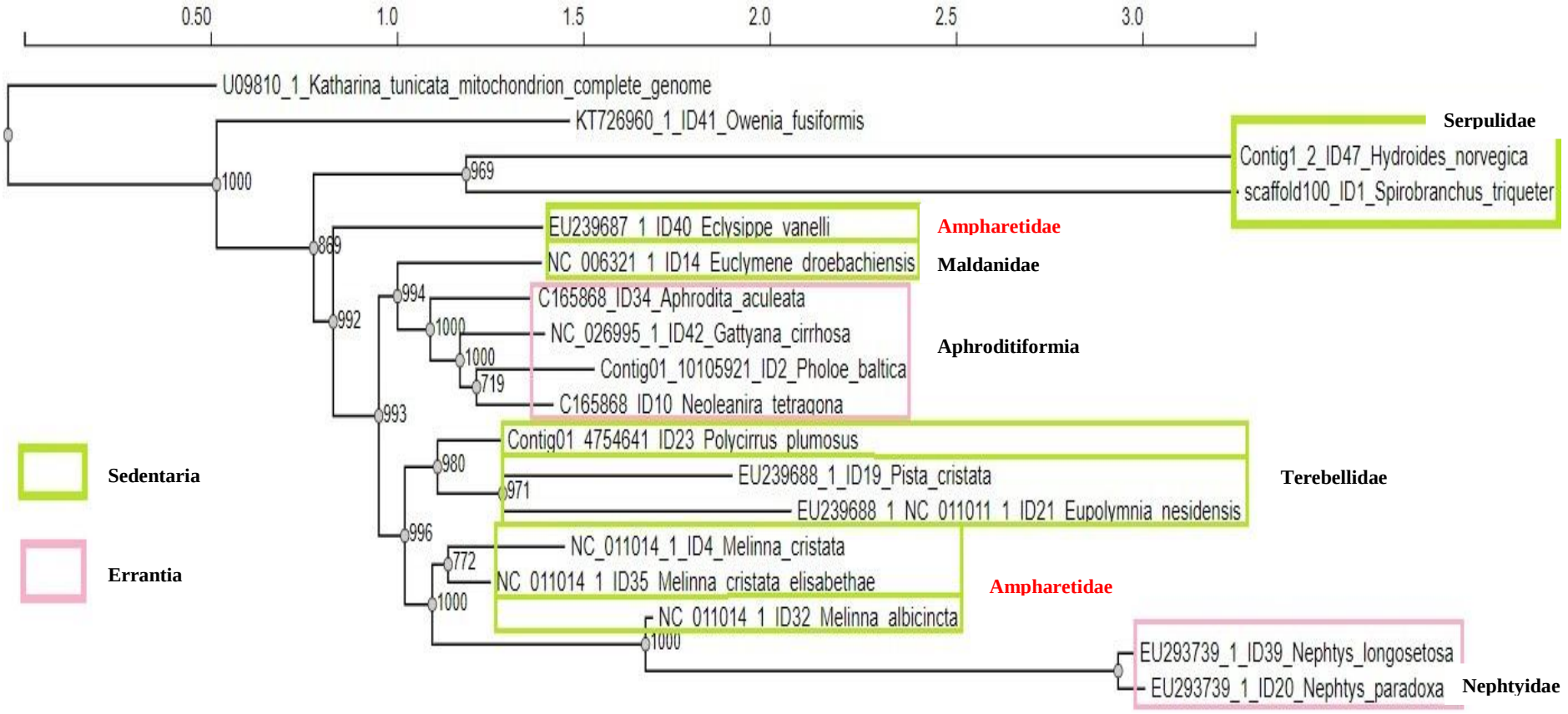


Fig 8 Phylogenetic tree inferred from 17 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML3.0 maximum-likelihood methods + BIC (Bayesian Information Criterion) Best model: GTR + G and Bootstrap 1000 replicates. The minimum Bootstrap value is 719.

5.5.3 Based on Single Gene: COI

The phylogenetic relationships of the samples seemed relatively consistent with the sorted tables (Table 2 and 3) that we presented in the Section 2.3. The samples coming from the same family/suborder formed monophyletic groups. However, the minimum bootstrap value was as low as 13 (out of bootstrap 100 replicates) (Appendix A.38 showed one of the results). Therefore, we are uncertain about the phylogenetic relationships here.

6 Discussion

6.1 The Nature of Our Sample Datasets

Our first assembly tool is NOVOPlasty, in the early stage of the tests, there were only 2 samples can be assembled: ID 2 and 47. The rest either showed "COVERAGE IS TOO LOW, SHOULD BE MORE THAN 10X" or "INVALID SEED, PLEASE TRY AGAIN WITH A NEW ONE". This leads to a doubt of our sample datasets to be low coverage.

Our second assembly tool is MitoZ, ID 1, 2, 10, 24, 26 and 47 were assembled with COI sequences while ID 23 and 41 assembled without COI sequences. In total, 8 out of 36 samples had results. The rest showed "All sequences are low abundance (<10X)". This further supports the doubt, the nature of our sample datasets is probably low coverage.

In order to be more certain of this, the first half part of Test 10 in NOVOPlasty and Test 5 in MITObim were conducted. All of the results that mentioned in Results section 5.1: 1) suggest ID 2 is the most robust datasets among the entire sample datasets. 2) Explain why ID2 could be easily assembled by all of the 3 assembly tools while the rest of the sample datasets all failed by one or more tools. 3) Explain why NOVOPlasty can assembly better when using rRNAs as seeds than COI as seeds.

Last but not least important, NOVOPlasty suggests ID 2 have a coverage of 114X. Then simple mathematics calculation, any number of hits that lower than 100 are below 10X coverage. This 1) consistents with NOVOPlasty initial tests with only two sample datasets assembled: ID 2 (1090 COI hits) and 47 (148 COI hits). 2) Explains why NOVOPlasty showed so many "COVERAGE IS TOO LOW, SHOULD BE MORE THAN 10X".

Furthermore, the number of hits recorded was the maxim number, so some of the numbers of hits could be “0” which might 3) explain why it also showed “INVALID SEED, PLEASE TRY AGAIN WITH A NEW ONE”.

Overall, all of above suggest the nature of our sample datasets is low coverage (<10X) except ID 2 is a robust dataset. However, the reason why ID 2 is enriched with mitochondrial sequences such as COI sequences is still unknown. 1) It could be this species contains more mitochondrion than the rest of the species we studied here, and then the next question would be “which part of the tissue we used during wet-lab steps? (This was done by others)” or 2) most likely, it could be just the probability issues: Coincidence. Since the rest of the datasets are behaved similarly.

This also shows the chosen bioinformatics tools are sensitive enough to tell the nature of datasets. We discuss them in below section.

6.2 Bioinformatics Pipeline

Here we propose a bioinformatics pipeline for mitochondrial genome assembly, annotation and phylogenetic tree reconstruction (summarized in Fig 2). This pipeline can be reused in the future with more polychaete species to achieve a bigger picture of Polychaete phylogeny.

6.2.1 Assembly Tools

Table 19 is a summary table for assembly tools. We discuss them below in details.

Table 19 Summary table for Assembly Tools.

Assembly Tools	NOVOPlasty	MitoZ	MITObim
Algorithm	*de novo* assemblers	*de novo* assemblers	baiting and mapping strategy
	seeds or reference sequences based tools	*de Bruijn* graph (DBG) algorithm	seeds or reference sequences based tools
	seed sequences are used to **retrieve** a sequence read		either seeds or reference sequences are used as a **starting** reference
Assembly Time	quick	most convenient from raw data to final visualized mitochondrial genome only took about 30-40 min	slowest
Computer Memory Requirement	maximum memory 16 GB	100 GB when using thread_number 16	standard laptop or desktop computer with at least 2 GB RAM
Datasets Requirement	30X	1.5 to 3G base pair (bp)	least

6.2.1.1 Algorithm

In this thesis, we used 3 different assembly tools. NOVOPlasty and MitoZ are *de novo* assemblers while MITObim using baiting and mapping strategy for assembly. From another perspective, NOVOPlasty and MITObim are seeds or reference sequences based tools, while MitoZ do not request any of these by using the *de Bruijn* graph (DBG) algorithm. Furthermore, NOVOPlasty use seed-and-extend algorithm, thus the seed sequences are used to "retrieve" a sequence read from the dataset and subsequently extend it. While MITObim use baiting and iterative mapping algorithm, thus either seeds or reference sequences are used as a "starting" reference to initiate the assembly and through repeat baiting and mapping to close the gaps.

6.2.1.2 Assembly Time

MitoZ is the most convenient tool among 3 tools. For example, for ID 2, from raw data to final visualized mitochondrial genome only took about 30 - 40 min. No seeds or reference sequences are needed and only one script can get the job done. NOVOPlasty is a quick assembler indeed. If using NOVOPlasty for assembly and MitoZ for annotation, the computer running time maybe shorter. However, it takes time to download and try out different seeds to optimize the results. In addition, these request 2 different scripts to get both job done. Therefore, overall it would be longer than 40 min. MITObim usually is the slowest among 3 tools due to its baiting and iterative mapping algorithm. The required number of iterations for successfully completing the mitochondrial genome is directly affected by the genetic distance between the available reference sequences and targeted species (Hahn *et al*., 2013). For example, for ID 2, Test 11 took 83 iterations, this may take 30 min already, not to mention the later annotation step.

6.2.1.3 Computer Memory Requirement

NOVOPlasty has a maximum memory requirement of 16 GB being the least computer-demanding tool. MitoZ requests around 100 GB when using thread_number 16 maybe the most computer-demanding tool. MITObim depends on the chosen mode. “T1” mode needs more computer memory than “T2” and “T3” mode because of the initial mapping assembly. However, any standard laptop or desktop computer with at least 2 GB RAM can perform all subsequent steps (Hahn *et al*., 2013).

6.2.1.4 Datasets Requirement

MITObim is the most “successful” assembler among 3 tools, able to assembly 32 out of 36 samples, despite the fact that it is slow in assembly and requires certain computer memory. It is said that COI sequence as a seed is sufficient to retrieve a complete mitochondrial genome (Hahn *et al*., 2013). However, in our case, “T3” mode using COI sequence as a seed is indeed super efficient at fishing out our targeted COI sequences, most of the results coming from this strategy. Nevertheless, not the complete or nearly complete mitochondrial genome though. Overall, consider the nature of our sample datasets and the results we have, MITObim has the least datasets requirement compared to the other 2 tools.

NOVOPlasty has many advantages compared to the other 2 tools used here. However, datasets requirement is not one of them. It requests sufficient coverage as 30X for the organelle genome (Dierckxsens *et al*., 2017). This explains in the early stage of testing, the only 2 samples that can be assembled, ID 2 has a nice contig of 15202bp with optimal results: 13 CDS + 22 tRNA + 2 rRNA, the nearly completed novel mitochondrial genome with all the genes presented, while ID 47 has a short contig of 11066bp. Because the entire 36 sample datasets, only ID 2 has 114X coverage over 30X. Test 10 in Novoplasty, even if using the highest COI hits as seeds, the results still have not improved much. Only until Test 11, using the COI sequences that coming from each datasets itself, finally the results have improved to be able to assembly 19 samples. This very much proves that if one wants to use a quite easy tool with the aim at saving computer-running time and only has a standard computer with limited computer memory, as long as their datasets have sufficient coverage as 30X, NOVOPlasty would be an ideal tool. However, in our case, it probably is not the best tool here considering the nature of our sample datasets. Because one are looking at trying out different seeds to optimize the results with much efforts have been made while little return would be given. Overall, it is wise to have NOVOPlasty as first assembler to explore how robust the datasets are. Knowing the nature of datasets first, and then one can make a better assembly strategy, thus save lots of time and troubles in the future.

Similar story with MitoZ, it is a very convenient one-step solution tool. However, it is said, "About 1.5 to 3G base pair (bp) is enough for mitochondrial genome assembly." (linzhi2013, 2019b) simple mathematics: Gb=giga base pairs=1,000,000,000bp. Thus 1.5G-3G =1,500,000,000bp-3,000,000,000bp. Our sample datasets arrange from ID 14 total 583,847,140bp to ID 20 total 1,377,499,266bp (showed in Fig 1, the real numbers should be much less than this since we look at fastq file instead of fasta file, but for simple mathematics calculation to have an idea about our sample datasets, this is enough to prove our point). Therefore, ID 20 being the sample with most base pair is still less than data size requirement. This explains why MitoZ being the most "unsuccessful" tool in this thesis, only be able to assembly 6 samples. However, 4 out of 6 samples are optimal results, the nearly completed novel mitochondrial genome with all the genes presented. This means MitoZ itself is a great tool, the algorithm is no wrong. The only problem here is again, due to the nature of our sample datasets, it is not the best tool here. Still, it is not a bad choice to choose it as a second tool to be used, thanks to MitoZ's annotation mode. Especially, one script (Appendix A.3) is made to connect NOVOPlasty and MitoZ together, this smooth the whole process for keeping

trying out different seeds for NOVOPlasty to get the optimized assembly and annotation results. Lots of time can be saved by using this strategy. From another perspective, MitoZ does not request seed sequences that makes itself as a tool either works on one's datasets or not working at all. Changing k-mer helps optimizing the results in some way. However, if the datasets are robust, usually MitoZ produced the (nearly) optimal results straightaway, best example here ID 2. Of course, not just MitoZ, this is true for all the 3 assembly tools used here. That is the reason why ID 2 worked on all 3 tools easily which all produced optimal or nearly optimal results. For ID 10 and 26, MitoZ produced optimal results straightaway while it took 11 tests for NOVOPlasty finally got nearly optimal results (only obtained 10 CDS for both IDs). In this case, MitoZ seems to have lower datasets requirement than NOVOPlasty.

6.2.1.5 Assembly Pipeline and Strategy

In summary, we propose an assembly pipeline/strategy: if one is given or received a set of data, one can use NOVOPlasty to do "quality control" of the datasets first, to know the coverage of each sample datasets. Since it is quite easy to use and a quick assembler with less computer memory requirement. If NOVOPlasty works perfectly, then one probably only needs it to finish the whole assembly, unless one wants to do comparison tests as if we do here in this thesis. If NOVOPlasty works half well, then MitoZ maybe a quick fix. However, one needs to arrange for a better computer with ideal 100GB+ memory. If both NOVOPlasty and MitoZ work not very well. Then one is probably looking at the situation we have here in this thesis. In order to save lots of time that we spent here meaninglessly, one should probably give up NOVOPlasty and MitoZ immediately and turn to the third assembly tool. At the same time with the expectation that the final results would be incomplete mitochondrial genome that broken into several contigs. Furthermore, put the final generated results in doubt. Since probably not many assembly tools would work on these datasets, thus not enough contigs could be produced in the first place to rule out assembly and annotation errors.

Therefore, we propose that the best strategy would be 1) enrich or isolate the mitochondrion during "wet lab" step, thus give more mitochondrial derived reads in the sample datasets. 2) Sequence deeper, with higher coverage and more reads, the chances to get (nearly) completed mitochondrial genome would be higher. Unless one is very familiar with the species they are dealing with, knowing that the species are enriched with mitochondrion, then sequence accordingly to save the budget.

6.2.1.6 Future Perspective

One paper showed datasets at the sizes of 1 million reads, the results tend to be incomplete mitochondrial genome, which broken into several contigs (Richter, Schwarz *et al*., 2015). This consistent with our results since our samples are between <1 - 2 million reads (showed in Table 4), ID 14 have 841,436 reads being the least while ID 20 has 1,974,628 reads being the most (reads data provided by others). It also mentioned when 4 million reads, over 95% mitochondrial genome could be recovered in most cases (Richter *et al*., 2015). Therefore, if we are ever to aim at getting all the 36 (nearly) completed mitochondrial genome in the future, one thing we could try is to sequence deeper.

6.2.2 Annotation Tools

The 2 annotation tools we used here: MitoZ and MITOS have their own advantages and disadvantages, summarized in Table 20.

Table 20 Summary table for Annotation Tools

Annotation Tools	**MitoZ**	**MITOS**
advantages	1) relatively quick 2) users get fully control 3) results are auto-saved and can be kept as long as possible 4) annotate multiple sequences in one input file 5) with one script, perform the annotation in bulk 6) output is a human readable circle mitochondrial genome	1) web server that quite easy to use 2) reliable and consistent
disadvantages	1) computer requirement at least 100GB memory 2) requests certain level of bioinformatics skills 3) MitoZ claimed working well mainly for "arthropods" and "mammals" for the protein coding genes annotation	1) takes about 1.5 hour for annotating 1 sample 2) share open resource, users get no control, they are put in a queue. 3) results are deleted automatically after 1 month 4) only handle one sequence in one input file

MITOS is widely used by more than 500 papers (MITOSWebServer) mainly because 1) it is a web server that quite easy to use without the need of any prior bioinformatics knowledge, makes it an easy access for anyone; 2) at some point it is probably reliable and consistent as it claimed. Therefore, keep using by more and more papers. However, its disadvantages are quite obvious as well, 1) it takes MITOS about 1.5 hour (MITOSWebServer) for annotating a single average length mitochondrial genome and we have 36 sample datasets here; 2) it is a shared open resource. Therefore, users get no control of using MITOS. If multiple users submit jobs at the same period or one user submit multiple jobs at a time, they are put in a queue. Below two experiences happened at the same week. One experience encountered was that after submission, it showed "Your job (xMaCqiop) is on position 181 in the queue", thus the job was submitted on one Saturday and it finished until the next Thursday. Another experience was that the job was on the queue of 29, but it stuck for the whole weekend by showing the same queue number non-change; 3) the results are deleted automatically after 1 month. Therefore, manually save all the results are needed; 4) it can only handle one sequence in one input file. For example, if two contigs are generated in one fasta file, in order to annotate both, it has to be separated into 2 input files. Therefore, it is not very convenient for our broken mitochondrial genomes with multiple contigs in one output file. Therefore, in this thesis, only the 12 nearly completed mitochondrial genomes have been annotated by MITOS as a comparison approach to MitoZ.

On the other hand, MitoZ has certain advantages such as 1) relatively quick, usually take less than 30min to finish one annotation depends on the length of mitochondrial genome. 2) Users get full control of MitoZ, since it is a closed terminal-based tool. 3) The results are auto-saved by computer and results can be kept as long as possible. 4) It can annotate multiple sequences in one input file. 5) With one script, it can perform the annotation in bulk. 6) The output is a human readable circle mitochondrial genome. However, the disadvantages make it can only be used by certain users, because 1) as mentioned earlier, the computer requirement, at least 100GB memory. 2) It requests certain level of bioinformatics skills to manage the tool, at the very least it cannot be zero knowledge like MITOS. 3) Most importantly, MITOS claimed it as an annotation tool for "metazoan" mitochondrial genomes (Bernt *et al*., 2013), while MitoZ claimed working well mainly for "arthropods" and "mammals" for the protein coding genes annotation (Meng *et al*., 2019).

6.2.2.1 Future Perspective

In this thesis, we used “Annelida” setting as mentioned earlier, since the annotation results from both tools showed similar but not exactly the same results. Therefore, if we are ever to aim at getting all the 36 well-annotated (nearly) completed mitochondrial genome in the future, another thing we could do is to look into the annotation difference closer.

6.2.3 Phylogenetic Tree Tools

The 2 phylogenetic tree reconstruction tools we used here are: NGPhylogeny.fr and ATGC Bioinformatics Platform. Both of them are web servers. Easy to use, no previous bioinformatics knowledge needed. Simply uploaded the right file format (fasta for NGPhylogeny.fr and PHYLIP for ATGC Bioinformatics Platform) and wait for the results.

Overall, NGPhylogeny.fr is more convenient than ATGC Bioinformatics Platform, since it is a one-click solution: from raw data to final phylogenetic tree. However, the main disadvantage for NGPhylogeny.fr is that it is not always working, this means sometimes the task would finish with errors while ATGC Bioinformatics Platform always works/more stable, in a sense more reliable. We want our results consistently coming from one tool. Therefore, we used ATGC Bioinformatics Platform to produce the final phylogenetic trees, while using NGPhylogeny.fr to test different tree-making strategies since it is more convenient.

Finally, Bootstrap testing is more sensitive than PhyML3.0’s new algorithm: Fast likelihood-based method: aLRT SH-like. We discover this by forgetting to choose the bootstrap option once. Appendix A.31 shows 1.0 value for all the nodes while Fig 7 shows the minimum value is 864 in one node. This suggests the relationships within the tree are not equal. Some are weaker compared to others. In other words, some of the nodes should be put in question. However, the main disadvantage for Bootstrap testing is taking too much time. Bootstrap 1000 replicates take about 4 days for “21_Genome_for_tree.fasta” while 100 replicates only take about 7 hours. PhyML3.0’s new algorithm: aLRT SH-like takes even less. Therefore, we use aLRT SH-like for testing in NGPhylogeny.fr while Bootstrap 1000 replicates in ATGC Bioinformatics Platform for the final phylogenetic trees.

In summary, we propose a phylogenetic tree reconstruction pipeline/strategy: if the datasets are not robust enough, using NGPhylogeny.fr with aLRT algorithm to test out which nodes are the weakest and remove the species from the corresponding clade, repeat until the phylogenetic tree becomes more trustworthy then use Bootstrap 1000 or 100 replicates to confirm it. This could save lots of computering time.

6.3 Mitochondrial Gene Order

We compare our 6 nearly completed mitochondrial genome sequences with the current available corresponding sequences in the database NCBI and find out that the mitochondrial gene orders are the same or similar. Details as follows: 1) for ID 40 and 41, the mitochondrial gene orders are the same (Weigert *et al*., 2016; Zhong *et al*., 2008). Furthermore, 2) for ID 20 and 39, they are both *Nephtys* sp. Therefore, we compare with *Nephtys* sp. 'San Juan Island' YV-2008 mitochondrion, complete genome sequence (EU293739.1) in NCBI and find out that the mitochondrial gene orders are the same for ID 20 and similar for ID 39 (since we are missing 2 CDS here) (Valles, Halanych *et al*., 2008). In addition, 3) for ID 1 - *Spirobranchus triqueter*, since there is no corresponding sequences in NCBI. Therefore, we compare it with *Spirobranchus giganteus* mitochondrion, complete genome sequence (NC_032055.1) and find out that the mitochondrial gene orders are similar (since we are missing 2 CDS here) (Seixas, Russo *et al*., 2017). Similarly, 4) for ID 17 - *Scoloplos armiger*, we compared with *Scoloplos cf. armiger* CB-2006 mitochondrion, partial genome sequence (DQ517436.1) and find out that the mitochondrial gene orders are the same (Bleidorn, Kruse *et al*., 2006).

Furthermore, for ID 2, 10, 26 and 34 so far there are no available corresponding sequences in NCBI. Therefore, we compare our results with each other and found out 5) for ID 2, 10, 17, 20, 26, 34 and 39, in total 7 out of our 12 nearly completed mitochondrial genomes have the same mitochondrial gene orders. Moreover, they have the same mitochondrial gene orders as Pleistoannelida (which includes Errantia and Sedentaria) that described in (Weigert *et al*., 2016). This also consistent with previous studies regarding Annelida mitochondrial gene orders are relatively conserved (Jeffrey L Boore & Brown, 2000; Jennings & Halanych, 2005; Li, Kocot *et al*., 2015; Weigert & Bleidorn, 2016; Weigert *et al*., 2016; Zhong *et al*., 2008).

However, 6) for ID 24 and 47, we find out two novel Annelida mitochondrial gene orders. Their mitochondrial gene orders are different from our own results as well as those of all

other reported mitochondrial genomes in Annelida (Weigert *et al*., 2016). In this thesis, actually in total 5 out of our 12 nearly completed mitochondrial genomes: ID 1, 24, 40, 41 and 47 have different mitochondrial gene orders as Pleistoannelida. This suggests that 1) Annelida mitochondrial gene orders are only "relatively" conserved.

What more interesting is that (Zhang, Sun *et al*., 2018) suggested, "Mitochondrial gene order in the shallow-water (1-301m) families/genera is conserved, but in the deep-sea (1122-2800m) polynoids there are two patterns of gene order rearrangement". However, in this thesis, the above-mentioned 5 samples that have different mitochondrial gene orders are coming from: 19.2m, 100m, 57m, 21.2m and 21.2m, respectively (depth data provided by others). Therefore, this suggests 2) novel Annelida mitochondrial gene orders could happen in different depth, either shallow - intermediate shallow or deep-sea. More research needs to be done to see the full picture and the mechanisms of such mitochondrial genome evolution.

Finally, due to the nature of our sample datasets, not many nearly completed mitochondrial genomes are generated (only 12 out of 36). Moreover, the generated broken mitochondrial genomes with multiple contigs for the rest of 24 sample datasets, making it hard to be certain of the actual mitochondrial gene orders. Therefore, these will not be discussed further.

6.4 Phylogenomic Study

6.4.1 Single Gene, Multiple Genes or Mitochondrial Genome?

6.4.1.1 Mitochondrial Genome

We first use 12 nearly completed mitochondrial genomes + outgroup: Katharina unicata mitochondrion, complete genome (U09810.1) to reconstruct our phylogenetic tree. However, the minimum bootstrap value is only 607 (showed in Appendix A.29). In order to make our tree more trustworthy, we try to remove IDs from the most unstable clade and find out to remove ID 24 would generate the most trustworthy phylogenetic tree. We also generate 4 phylogenetic trees (without ID 24) using 4 different methods and models: the combination of PhyML and PhyML 3.0 methods plus AIC and BIC criterion for SMS models, and the results show the same. This proves that: although using different algorithms, the results are still consistent with each other, thus our phylogenetic trees are trustworthy.

The phylogenetic tree (Fig 7) with ID 24 removed, its minimum bootstrap value goes up to 864 being the highest minimum number (other 3 phylogenetic trees can be found in Appendix A.30-A.32). This shows the relationships in the monophyletic group of ID 17 - *Scoloplos armiger* from family Orbiniidae and ID 24 - *Aricidea catherinae* from family Paraonidae are relatively weak compared to other relationships, thus lower the whole phylogenetic tree's bootstrap value, thus place ID 24's position in the tree in question.

6.4.1.2 Multiple Genes

We first use 21 partial mitochondrial genomes + outgroup: Katharina unicata mitochondrion, complete genome (U09810.1) to reconstruct our phylogenetic tree. However, the minimum bootstrap value is only 302 (showed in Appendix A.33). In order to make our tree trustworthy, we use same strategy and find out to remove ID 17, 24, 26, 28 would generate the most trustworthy phylogenetic tree. The subsequently generated 4 phylogenetic trees using 4 different methods and models show the same results (bootstrap reduced from 1000 to 100 replicates in order to reduce computering time from about 3-4 days to 7-8 hours).

The phylogenetic tree (Fig 8) with ID 17, 24, 26, 28 removed, its minimum bootstrap value goes up to 719 and the bootstrap values of deeper nodes are all above 869. (Other 3 phylogenetic trees can be found in Appendix A.34-A.36). This shows the relationships for ID 17, 24, 26, 28 to their sister taxa are relatively weak compared to other relationships, thus lower the whole phylogenetic tree's bootstrap value, thus place ID 17, 24, 26, 28's positions in the tree in question.

6.4.1.3 Single Gene

We first use 26 COI sequences + outgroup: Katharina unicata mitochondrion, complete genome (U09810.1) to reconstruct our phylogenetic tree. Unlike using mitochondrial genomes, the results generated by using COI sequences are 1) not the same when using the 4 different methods and models. Furthermore, the minimum bootstrap value is 2) as low as 58 when using 1000 replicates! (Appendix A.37 showed one of the results.) These show that the

phylogenetic tree inferred from COI sequences are highly unstable, thus the results would be highly untrustworthy.

We then use the same strategy, try to make a trustworthy tree. We remove ID 17, 18, 24, 26 and the subsequently generated 4 phylogenetic trees using 4 different methods and models 1) show the same results (bootstrap reduced from 1000 to 100 replicates in order to reduce computering time). However, the minimum bootstrap value is 2) as low as 13 (Appendix A.38 showed one of the results). These show that we managed to stable the results yet failed to make it trustworthy. Therefore, we are uncertain about the phylogenetic relationships when using COI sequences.

6.4.1.4 Single Gene vs Multiple Genes and Mitochondrial Genome

We also compared the results that generated by COI sequences with that generated by partial or nearly completed mitochondrial genomes. In order to compare with nearly completed mitochondrial genomes, we use 10 corresponding COI sequences (ID 39 is not included due to the cut-off line mentioned in Results section 5.3), then we apply the 4 different methods and models, and this 1) generates 3 different results (Appendix A.39 showed one of the results). Furthermore, in order to compare with partial mitochondrial genomes, we use 15 corresponding COI sequences (ID 14 and 39 are not included due to the cut-off line mentioned in Results section 5.3), then we apply the same approach, and this 2) generates 2 different results (Appendix A.40 showed one of the results).

It turns out all the results that generated by COI sequences are very different from that generated by partial or nearly completed mitochondrial genomes. Whereas the result that generated by partial mitochondrial genomes is not only consistent with but also is the extension of that generated by nearly completed mitochondrial genomes. In addition, all the results that generated by COI sequences, the minimum bootstrap values are 3) as low as 7 and 3, respectively (out of bootstrap 100 replicates, bootstrap reduced from 1000 to 100 replicates in order to reduce computering time). Whereas the minimum bootstrap values for partial or nearly completed mitochondrial are 719 and 864, respectively (out of bootstrap 1000 replicates).

All of above suggest that single gene - COI sequences are probably not the best genetic marker for molecular phylogeny study of polychaetes, it alone is insufficient to resolve the phylogenetic relationships among polychaetes. Whereas partial or nearly completed mitochondrial genomes not only showed consistent results despite of 4 different tree reconstruction algorithms but also showed high bootstrap support even for deeper nodes, with high potentials to resolve this problem. This consistent with what we have mentioned in the Introduction section 2.3, previously many studies have been conducted to use different genetic markers to try to resolve the phylogenetic relationships among Annelida. However, this problem still remains controversial today. But with phylogenomic study, to finally resolve this problem might become possible.

6.4.1.5 Future Perspective

With the help of high-throughput next generation sequencing technology, we believe that we are now at the beginning of an era to use phylogenomic study to unreveal Polychaeta as well as Annelida phylogeny. Our proposed bioinformatics pipeline could be of use to fulfill this task in an effiective and reliable way. Furthermore, we use several comparsion approaches to make sure our results are trustworthy.

6.4.2 Polychaete Phylogenetic Relationships

6.4.2.1 Terebellomorpha

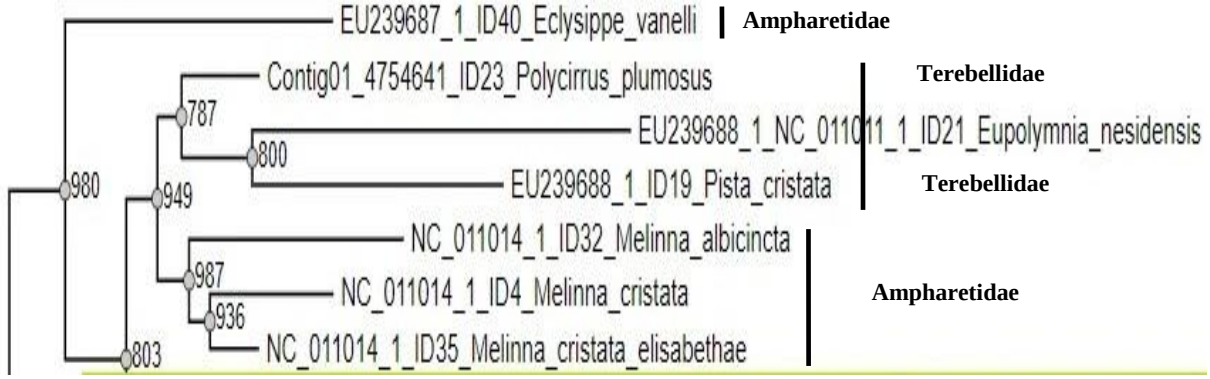


Fig 9 Phylogenetic relationships within suborder Terebellomorpha by using mitochondrial genomes (from Appendix A.33)

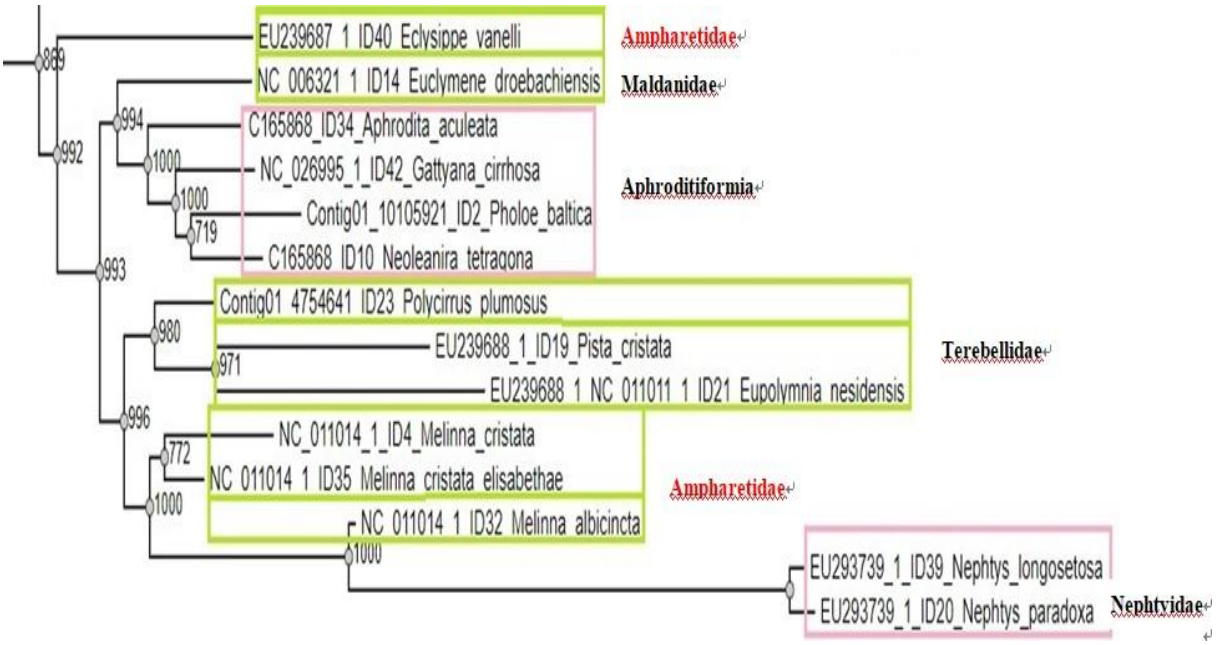


Fig 10 Phylogenetic relationships within suborder Terebellomorpha by using mitochondrial genomes (from Fig 8)

Fig 11 Phylogenetic relationships within suborder Terebellomorpha by using COI sequences (from Appendix A.38)

Our results by using mitochondrial genomes show that besides ID 40 *Eclysippe vanelli*, Family Ampharetidae and Terebellidae form a monophyletic clade (Fig 9). However, after removing the species that unstable the phylogenetic tree, besides ID 40, Terebellidae become non-monophyletic to Ampharetidae (Fig 10). On the other hand, when using COI sequences, ID 40 is with other Ampharetidae family members together with Terebellidae form a monophyletic clade (Fig 11).

We have 8 species that come from 2 different families: Ampharetidae and Terebellidae but same Suborder Terebellomorpha. Both our results by using mitochondrial genomes (Fig 9) and by using COI sequences (Fig 11) show that Ampharetidae and Terebellidae have been successfully recovered into one monophyletic clade, although one with high bootstrap values (Fig 9) while the other (Fig 11) is not, and ID 40 *Eclysippe vanelli*'s position is different. Nevertheless, this consistent with previous study showed that Order Terebellida Suborder Terebellomorpha including taxa from Ampharetidae and Terebellidae (Dales, 1962; Glasby *et al*., 2004; Holthe, 1986; Rousset, Rouse *et al*., 2003).

Furthermore, we have successfully reconstructed the phylogenetic relationships between these 2 families: Ampharetidae and Terebellidae with high bootstrap values (Fig 9 and 10). The phylogenetic relationships within Terebellomorpha remain debatable. Our two conflicting results reflect this. Traditional views: Family Ampharetidae and Terebellidae form a monophyletic clade (Rousset *et al*., 2003). But one study also showed Ampharetidae and Terebellidae are non-monophyletic (Colgan, Hutchings *et al.*, 2001). One reasonable explaination for these conflicting results could be people used different methods with different species thus naturally the results would be different. In our case, when the methods are the same, but results Fig 9 using 22 species, while results Fig 10 using only 18 species, this suggests certain species might affect the results greatly. Moreover, when the methods are also different, Fig 11 using 22 COI sequences, the third result comes up. Overall, we suspect that Terebellomorpha (especially Family Ampharetidae) are important taxa during polychaete phylogeny, it maybe one of the ancestor version of polychaeta thus its phylogenetic position within Annelid as well as the phylogenetic reationships within itself remain so controversial. This suspicion is supported by ID 40 *Eclysippe vanelli* is closer to the basal part of the tree. The reason for this might be its mitochondrial gene order is slightly different as mentioned in Section 5.4 and 6.3. Be noted that both of the phylogenetic trees (Fig 9 and 10) with high bootstrap support.

Although both of our conflicting results with high bootstrap support, the second one (Fig 10) comes from a more stabilized phylogenetic tree, thus it is more convincing. Therefore, we propose the phylogenetic relationships within Suborder Terebellomorpha: Terebellidae maybe nested within Ampharetidae, in another words, Ampharetidae evolve towards/split into Ampharetidae and Terebellidae.

6.4.2.2 Phyllodocida

In this thesis, we have 9 species from Order Phyllodocida. Among these, 4 from Suborder Aphroditiformia, and 2 from Family Nephtyidae. The rest 3 species, ID 3 *Nereimyra punctata*, ID 28 *Goniada maculata* and ID 38 *Sphaerodorum gracilis*, we are uncertain about their positions in the phylogenetic tree, thus will not be discussed further.

6.4.2.2.1 Aphroditiformia

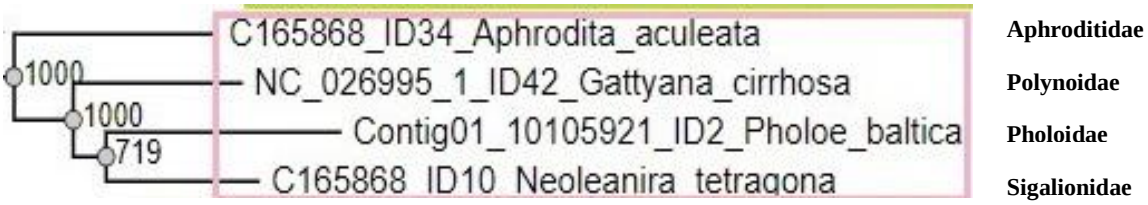


Fig 12 Phylogenetic relationships within suborder Aphroditiformia (from Fig 8)

Our results show Suborder Aphroditiformia form a monophyletic clade (Fig 12). We have 4 species that come from 4 different families but same Suborder Aphroditiformia, and they have been successfully recovered into one monophyletic clade with high bootstrap values. This consistent with previous study showed that scale-worms including taxa from Aphroditidae, Pholoidae, Polynoidae and Sigalionidae (Aphroditiformia) (Norlinder *et al*., 2012). Furthermore, we have successfully reconstructed the phylogenetic relationships among these 4 families with high bootstrap values. This also consistent with previous studies that 1) Aphroditidae are the basal group within Aphroditiformia. 2) Polynoidae are sister to the monophyletic clade formed by Pholoidae and Sigalionidae. 3) Pholoidae and Sigalionidae are most closely related (Norlinder *et al*., 2012; Wiklund *et al*., 2005). However, in this thesis, we only have one species from both Pholoidae and Sigalionidae. Therefore, it is not possible to tell whether Pholoidae are nested within or a distinct group from Sigalionidae. Since previous studies have showed conflicting results, first included Pholoidae in Sigalionidae (Fauchald, 1977), but later re-erected it as a distinct group from Sigalionidae (Pettibone, 1992). Then Wiklund *et al.* again showed that Pholoidae are nested within Sigalionidae (Wiklund *et al*., 2005).

Overall, our results make both mathematics (high bootstrap support) and biological sense. Therefore, we propose the phylogenetic relationships within Suborder Aphroditiformia:

Aphroditidae are the basal group within Aphroditiformia, and Polynoidae are sister to the monophyletic clade formed by Pholoidae and Sigalionidae.

6.4.2.3 Serpulidae

The two species ID 1 *Spirobranchus triqueter* and ID 47 *Hydroides norvegica* come from same Family Serpulidae have been successfully recovered into one monophyletic clade with high bootstrap values (Fig 7 and 8). We have discussed in Section 6.3, ID 1 has similar mitochondrial gene orders as one published species that comes from the same family, whereas ID 47 shows novel Annelida mitochondrial gene orders. Nevertheless, both of them have very different mitochondrial gene orders. Therefore, we would expect that they might be more separated or distantly related to other species that studied here. Fig 7 and 8 indeed show they are closer to the basal part of the tree with a long branch. However, since their mitochondrial gene orders are so different from each other, some questions might be raised: are they really closely related to each other? Could it be long branch attraction (LBA) artifacts?

Serpulidae and Oweniidae were formerly combined in a clade called Sabellida (Rouse & Fauchald, 1997), the composition of which have been debated for years. Now that Oweniidae are considered as the basal group (Weigert *et al*., 2016), our results (Fig 7 and 8) also support this with high bootstrap values. However, Serpulidae are still placed in Sabellida (Rouse & Pleijel, 2001). In this thesis, our results (Fig 7 and 8) show that Serpulidae also close to the basal part of the phylogenetic tree with high bootstrap support. This makes us wondering Order Sabellida might be just like Suborder Terebellomorpha (especially Family Ampharetidae) are another important taxa during polychaete phylogeny, since some current and former members of Sabellida closer or belong to the basal group.

6.4.2.4 Future Perspective

In summary, what we have seen in this thesis is that the species that close to the basal part of the phylogenetic tree all show different mitochondrial gene orders (ID 41, ID 1, ID 47 and ID 40). In another words, the species with different mitochondrial gene orders all form the basal part of the phylogenetic tree. Previous study showed that ID 41*Owenia fusiformis* has a

different mitochondrial gene order which is confirmed by our results and now it is considered as the basal group (Weigert *et al*., 2016) which is also confirmed by our results. However, could this also be the bioinformatics tool problem? it puts the species with same mitochondrial gene orders together, the ones are different in the basal part of the tree (ID 41 and ID 40) and finally the ones are very different with each other (ID 1 and ID 47), suggested by Fig 7 and Table 18. Due to the nature of our sample datasets, we cannot be certain about this. Since we only achieved 12 nearly completed mitochondrial genomes, the rest of them are from partial mitochondrial genomes that we are uncertain about their mitochondrial gene orders. Nevertheless, this is a very interesting discovery that could be further confirmed.

Another pattern we have seen in this thesis is that some Order/Suborder/Family maybe the important taxa during polychaete phylogeny. Usually, they have certain traits: 1) members with different mitochondrial gene orders, 2) near to the basal part of the phylogenetic tree, 3) previous studies showed controversial results. We believe all of these suggest that the species in these groups due to various reasons decide to evolve to a better version to better cope with the living environment. It is because they are in the crossroad of evolution that their mitochondrial gene orders show differently. It is also because they are in the crossroad of evolution that the species could show different morphologically traits which could be wrongly identified, which would cause the following study come to a very different conclusion, thus produce controversial results. Plus we have discussed that people used different methods with different species thus naturally the results would be different. Nevertheless, if future polychaete phylogeny study could focus more on those important taxa using phylogenomic approach would be more effectively resolve this problem.

6.4.3 Evolutionary Direction

We have just discussed the species that come from same Order/Suborder/Family form monophyletic clades. Now we look at the phylogenetic relationships among the species that come from different Order/Suborder/Family.

6.4.3.1 Ampharetidae, Spionidae, Orbiniidae and Aphroditiformia

Fig 7 shows Spionidae, Orbiniidae and Aphroditiformia formed a monophyletic clade with high bootstrap support. We find one previous study showing Spionida and Orbiniida form a clade as well as Spionida derived from Terebellida (McHugh, 2000), these consistent with our results. Spionidae are known close to taxa: Sabellariidae that formerly belong to Sabellida (Weigert & Bleidorn, 2016) and Spionida close to Sabellida (Struck *et al*., 2015). Furthermore, we have just discussed that some current and former members of Sabellida closer or belong to the basal group. Therefore, there is a possibility that Spionidae also close to the basal part of the phylogenetic tree, and this would consistent with our results. In addition, previous studies have also showed that Orbiniidae are sister group to Siboglinidae (belong to Sabellida) together with Cirratuliformia (belong to Terebellida) (Weigert & Bleidorn, 2016; Weigert *et al*., 2014) and Orbiniidae are sister group to Phyllodocida (Colgan, Hutchings *et al*., 2006; Struck *et al*., 2008; Struck *et al*., 2011; Struck, Schult *et al*., 2007). Considering Ampharetidae belong to Terebellida and Aphroditiformia belong to Phyllodocida, according to WoRMS. All of these consistent with our results.

Therefore, our results make not only mathematics but also biological sense. We propose that 1) evolution direction from Ampharetidae → Spionidae → Orbiniidae → Aphroditiformia; 2) taxa like Orbiniidae in the crossroad of evolution, which were also regarded as intermediate forms of errant and sedentary living polychaetes (Fauvel, 1927) maybe are another important taxa during polychaete phylogeny since Orbiniidae are sister group to so many different taxa.

6.4.3.2 Ampharetidae and Nephtyidae

The two species ID 20 *Nephtys paradoxa* and ID 39 *Nephtys longosetosa* come from same Family Nephtyidae have been successfully recovered into one monophyletic clade with high bootstrap support (Fig 7 and 8). Surprisingly, this clade is not alongside with the Aphroditiformia group that also belong to same Order Phyllodocida. Instead, Fig 7 and 8 show the same pattern that this clade is always derived from Order Terebellida Family Ampharetidae. We could not find any previous study to support this pattern, but following the logic we have just discussed, if Ampharetidae are one ancestor version of polychaeta, then this is not that surprising anymore. In addition, considering both of the species belong to unassigned Suborder as Phyllodocida incertae sedis according to WoRMS. Then there is a

reason to believe that they are not that closely related to the Aphroditiformia group. Therefore, if no other closer species around, the most reasonable way would be derived from its closest ancestor. And this could be an explaination for what have showed in Fig 7 and 8 and the reason with a long branch.

6.4.3.3 Ampharetidae, Maldanidae and Aphroditiformia

Fig 8 shows Maldanidae and Aphroditiformia formed a monophyletic clade with high bootstrap support. Some studies have suggested that Maldanidae derived from Terebelliformia (Weigert & Bleidorn, 2016; Weigert *et al*., 2016), other studies have also showed that Maldanidae are associated with Terebellida (Brown *et al*., 1999; Hall *et al*., 2004; Rousset *et al*., 2007). Our results consistent with both above findings. As to the phylogenetic relationships between Maldanidae and Aphroditiformia, we find previous studies showing Scolecida are associated with Phyllodocida (Colgan *et al*., 2006; Struck *et al*., 2008). For example, Orbiniidae belong to Scolecida. Therefore, these also consistent with our results.

Therefore, our results make not only mathematics but also biological sense. We propose that evolution direction from Ampharetidae → Maldanidae → Aphroditiformia.

6.4.3.4 Sedentaria → Errantia?

We consider our phylogenetic trees (Fig 7 and 8) are quite robust with bootstrap values above 864 and 719. The whole phylogenetic relationships show a pattern of evolutionary direction from Sedentaria towards Errantia.

Due to the nature of our sample datasets, we only able to obtain 12 nearly completed mitochondrial genomes and 9 partial mitochondrial genomes. In addition, we have also removed 4 IDs in order to make our phylogenetic trees more trustworthy. Therefore, the evolutionary pattern we have discovered comes from a small sample size.

However, we find out one study (its Fig. 3 (B)) also shows this pattern. Nevertheless, due to the current major agreement based on 3 papers that monophyletic Errantia and Sedentaria have been recovered (Andrade, Novo *et al*., 2015; Struck *et al*., 2015; Weigert *et al*., 2014).

Therefore, this study describes the pattern as monophyletic Sedentaria could not be resolved (Weigert *et al*., 2016).

Furthermore, Orbiniidae are controversial taxa, in one study it shows up in the clade of Errantia, and this study named this clade as Errantia due to it is characterized by adaptation to a more errant life (Struck *et al*., 2011). Later studies show Orbiniidae are within Sedentaria clade (Struck *et al*., 2015; Weigert *et al*., 2014). One review paper describes its definite position as "still an open question" (Weigert & Bleidorn, 2016). In this thesis, Orbiniidae, the intermediate forms of errant and sedentary living polychaetes are indeed between Spionidae (Sedentaria) and Aphroditiformia (Errantia) (Fig 7). Be noted the phylogenetic tree with high bootstrap support. This result seems to resolve this controversial problem. Nevertheless, this result is in disagreement with the current major agreement based on above-mentioned 3 papers.

6.4.3.5 Future Perspective

We believe that there is a problem in either current classification of "Sedentaria and Errantia" or "monophyletic recovered Errantia and Sedentaria system", thus in certain way caused controversial results.

However, due to the nature of our sample datasets and a small sample size, there is a chance that this pattern only applies to this thesis.

As we mentioned earlier, we believe that we are now at the beginning of an era to use phylogenomic study to unreveal Polychaeta as well as Annelida phylogeny. Therefore, we believe that future studies will find the solutions.

7 References

Aguado, M. T., & Bleidorn, C. (2010). Conflicting signal within a single gene confounds syllid phylogeny (Syllidae, Annelida). *Molecular Phylogenetics and Evolution, 55*(3), 1128-1138. doi:10.1016/j.ympev.2010.01.012

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csaki (Eds.), *2nd International Symposium on Information Theory* (pp. 267-281). Budapest, Hungary: Akadémiai Kiadó.

Andrade, S. C., Novo, M., Kawauchi, G. Y., Worsaae, K., Pleijel, F., Giribet, G., & Rouse, G. (2015). Articulating "archiannelids": Phylogenomics and annelid relationships, with emphasis on meiofaunal taxa. *Molecular Biology and Evolution, 32*(11), 2860-2875.

ATGC-Bioinformatics-Platform. Citations. Retrieved from http://www.atgc-montpellier.fr/index.php?type=st

Bernt, M., Donath, A., Juhling, F., Externbrink, F., Florentz, C., Fritzsch, G., . . . Stadler, P. F. (2013). MITOS: Improved de novo metazoan mitochondrial genome annotation. *Molecular Phylogenetics and Evolution, 69*(2), 313-319. doi:10.1016/j.ympev.2012.08.023

Bleidorn, C., Kruse, I., Albrecht, S., & Bartolomaeus, T. (2006). Mitochondrial sequence data expose the putative cosmopolitan polychaete Scoloplos armiger (Annelida, Orbiniidae) as a species complex. *BMC evolutionary biology, 6*, 47. doi:10.1186/1471-2148-6-47

Bleidorn, C., Vogt, L., & Bartolomaeus, T. (2003). New insights into polychaete phylogeny (Annelida) inferred from 18S rDNA sequences. *Molecular Phylogenetics and Evolution, 29*(2), 279-288. doi:10.1016/S1055-7903(03)00107-6

Boore, J. L. (1999). Animal mitochondrial genomes. *Nucleic Acids Research, 27*(8), 1767-1780. doi:10.1093/nar/27.8.1767

Boore, J. L., & Brown, W. (2000). Mitochondrial genomes of Galathealinum, Helobdella, and Platynereis: sequence and gene arrangement comparisons indicate that Pogonophora is not a phylum and Annelida and Arthropoda are not sister taxa. *Molecular Biology and Evolution, 17*(1), 87-106.

Brown, S., Rouse, G., Hutchings, P., & Colgan, D. (1999). Assessing the usefulness of histone H3, U2 snRNA and 28S rDNA in analyses of polychaete relationships. *Australian Journal of Zoology, 47*(5), 499-516. doi:10.1071/Zo99026

Budaeva, N., Schepetov, D., Zanol, J., Neretina, T., & Willassen, E. (2016). When molecules support morphology: Phylogenetic reconstruction of the family Onuphidae (Eunicida, Annelida) based on 16S rDNA and 18S rDNA. *Molecular Phylogenetics and Evolution, 94*(Part B), 791-801. doi:10.1016/j.ympev.2015.10.011

Chevreux, B., Wetter, T., & Suhai, S. (1999). *Genome sequence assembly using trace signals and additional sequence information.* Paper presented at the Computer science and biology: Proceedings of the German Conference on Bioinformatics (GCB) 1999.

Colgan, D. J., Hutchings, P. A., & Braune, M. (2006). A multigene framework for polychaete phylogenetic studies. *Organisms Diversity & Evolution, 6*(3), 220-235. doi:10.1016/j.ode.2005.11.002

Colgan, D. J., Hutchings, P. A., & Brown, S. (2001). Phylogenetic relationships within the Terebellomorpha. *Journal of the Marine Biological Association of the United Kingdom, 81*(5), 765-773. doi:10.1017/S002531540100457X

Criscuolo, A., & Gribaldo, S. (2010). BMGE (Block Mapping and Gathering with Entropy): a new software for selection of phylogenetic informative regions from multiple sequence alignments. *BMC evolutionary biology, 10*(1), 210. doi:10.1186/1471-2148-10-210

Dales, R. P. (1962). THE POLYCHAETE STOMODEUM AND THE INTER-RELATIONSHIPS OF THE FAMILIES OF POLYCHAETA. *Proceedings of the Zoological Society of London, 139*(3), 389-428. doi:10.1111/j.1469-7998.1962.tb01837.x

Dierckxsens, N., Mardulyn, P., & Smits, G. (2017). NOVOPlasty: de novo assembly of organelle genomes from whole genome data. *Nucleic Acids Research, 45*(4), e18. doi:10.1093/nar/gkw955

Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*: Cambridge university press.

Fauchald, K. (1977). The polychaete worms. Definitions and keys to the orders, families and genera. *Natural History Museum of Los Angeles County, Science Series*.

Fauvel, P. (1927). Polychètes sédentaires, addenda aux errantes, archiannélides, myzostomaires. *Faune de France, 16*, 1-494.

Felsenstein, J. (1985). Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution, 39*(4), 783-791. doi:10.1111/j.1558-5646.1985.tb00420.x

Gascuel, O. (1997). BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol, 14*(7), 685-695. doi:10.1093/oxfordjournals.molbev.a025808

Glasby, C. J., Hutchings, P. A., & Hall, K. (2004). Assessment of monophyly and taxon affinities within the polychaete clade Terebelliformia (Terebellida). *Journal of the Marine Biological Association of the United Kingdom, 84*(5), 961-971. doi:DOI 10.1017/S0025315404010252h

Guindon, S., Dufayard, J. F., Lefort, V., Anisimova, M., Hordijk, W., & Gascuel, O. (2010). New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst Biol, 59*(3), 307-321. doi:10.1093/sysbio/syq010

Guindon, S., & Gascuel, O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol, 52*(5), 696-704. doi:10.1080/10635150390235520

Hahn, C., Bachmann, L., & Chevreux, B. (2013). Reconstructing mitochondrial genomes directly from genomic next-generation sequencing reads--a baiting and iterative mapping approach. *Nucleic Acids Research, 41*(13), e129. doi:10.1093/nar/gkt371

Hall, K. A., Hutchings, P. A., & Colgan, D. J. (2004). Further phylogenetic studies of the polychaeta using 18S rDNA sequence data. *Journal of the Marine Biological Association of the United Kingdom, 84*(5), 949-960. doi:10.1017/S0025315404010240h

Hasegawa, M., Kishino, H., & Yano, T. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution, 22*(2), 160-174. doi:10.1007/BF02101694

Holthe, T. (1986). Evolution, systematics, and distribution of the Polychaeta Terebellomorpha, with a catalogue of the taxa and a bibliography. *Gunneria, 55*, 1-236.

Jennings, R. M., & Halanych, K. M. (2005). Mitochondrial genomes of Clymenella torquata (Maldanidae) and Riftia pachyptila (Siboglinidae): evidence for conserved gene order in Annelida. *Molecular Biology Evolution, 22*(2), 210-222.

Juhling, F., Putz, J., Bernt, M., Donath, A., Middendorf, M., Florentz, C., & Stadler, P. F. (2012). Improved systematic tRNA gene annotation allows new insights into the evolution of mitochondrial tRNA structures and into the mechanisms of mitochondrial genome rearrangements. *Nucleic Acids Research, 40*(7), 2833-2845. doi:10.1093/nar/gkr1131

Katoh, K., & Standley, D. M. (2013). MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution, 30*(4), 772-780. doi:10.1093/molbev/mst010

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution, 16*(2), 111-120. doi:10.1007/BF01731581

Kojima, S. (1998). Paraphyletic status of Polychaeta suggested by phylogenetic analysis based on the amino acid sequences of elongation factor-1α. *Molecular Phylogenetics and Evolution, 9*(2), 255-261.

Krogh, A., Brown, M., Mian, I. S., Sjolander, K., & Haussler, D. (1994). Hidden Markov models in computational biology. Applications to protein modeling. *Journal of Molecular Biology, 235*(5), 1501-1531. doi:10.1006/jmbi.1994.1104

Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., . . . Marra, M. A. (2009). Circos: an information aesthetic for comparative genomics. *Genome Research, 19*(9), 1639-1645. doi:10.1101/gr.092759.109

Kupriyanova, E. K., & Rouse, G. W. (2008). Yet another example of paraphyly in Annelida: Molecular evidence that Sabellidae contains Serpulidae. *Molecular Phylogenetics and Evolution, 46*(3), 1174-1181. doi:10.1016/j.ympev.2007.10.025

Lanave, C., Preparata, G., Saccone, C., & Serio, G. (1984). A new method for calculating evolutionary substitution rates. *J Mol Evol, 20*(1), 86-93. doi:10.1007/bf02101990

Langeneck, J., Barbieri, M., Maltagliati, F., & Castelli, A. (2019). Molecular phylogeny of Paraonidae (Annelida). *Molecular Phylogenetics and Evolution, 136*, 1-13. doi:10.1016/j.ympev.2019.03.023

Lefort, V., Longueville, J. E., & Gascuel, O. (2017). SMS: Smart Model Selection in PhyML. *Molecular Biology and Evolution, 34*(9), 2422-2424. doi:10.1093/molbev/msx149

Lehrke, J., ten Hove, H. A., Macdonald, T. A., Bartolomaeus, T., & Bleidorn, C. (2007). Phylogenetic relationships of Serpulidae (Annelida: Polychaeta) based on 18S rDNA sequence data, and implications for opercular evolution. *Organisms Diversity & Evolution, 7*(3), 195-206. doi:10.1016/j.ode.2006.06.004

Lemer, S., Kawauchi, G. Y., Andrade, S. C., Gonzalez, V. L., Boyle, M. J., & Giribet, G. (2015). Re-evaluating the phylogeny of Sipuncula through transcriptomics. *Molecular Phylogenetics and Evolution, 83*, 174-183. doi:10.1016/j.ympev.2014.10.019

Li, Y., Kocot, K. M., Schander, C., Santos, S. R., Thornhill, D. J., & Halanych, K. M. (2015). Mitogenomics reveals phylogeny and repeated motifs in control regions of the deep-sea family Siboglinidae (Annelida). *Molecular Phylogenetics and Evolution, 85*, 221-229. doi:10.1016/j.ympev.2015.02.008

linzhi2013. (2019a). --clade option for "Annelida-segmented-worms". Retrieved from https://github.com/linzhi2013/MitoZ/issues/25

linzhi2013. (2019b). Requirment of data size and insert size. Retrieved from https://github.com/linzhi2013/MitoZ

McHugh, D. (1997). Molecular evidence that echiurans and pogonophorans are derived annelids. *Proceedings of the National Academy of Sciences of the United States of America, 94*(15), 8006-8009. doi:10.1073/pnas.94.15.8006

McHugh, D. (2000). Molecular phylogeny of the Annelida. *Canadian Journal of Zoology, 78*(11), 1873-1884. doi:10.1139/cjz-78-11-1873

Meng, G., Li, Y., Yang, C., & Liu, S. (2019). MitoZ: a toolkit for animal mitochondrial genome assembly, annotation and visualization. *Nucleic Acids Research, 47*(11), e63. doi:10.1093/nar/gkz173

MITOSWebServer. Studies using MITOS. Retrieved from http://mitos.bioinf.uni-leipzig.de/help.py#example

Nawrocki, E. P., & Eddy, S. R. (2013). Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics, 29*(22), 2933-2935. doi:10.1093/bioinformatics/btt509

Norlinder, E., Nygren, A., Wiklund, H., & Pleijel, F. (2012). Phylogeny of scale-worms (Aphroditiformia, Annelida), assessed from 18SrRNA, 28SrRNA, 16SrRNA, mitochondrial cytochrome c oxidase subunit I (COI), and morphology. *Molecular Phylogenetics and Evolution, 65*(2), 490-500. doi:10.1016/j.ympev.2012.07.002

Pamungkas, J., Glasby, C. J., Read, G. B., Wilson, S. P., & Costello, M. J. (2019). Progress and perspectives in the discovery of polychaete worms (Annelida) of the world. *Helgoland Marine Research, 73*(1), 4. doi:10.1186/s10152-019-0524-z

Patra, A. K., Kwon, Y. M., Kang, S. G., Fujiwara, Y., & Kim, S. J. (2016). The complete mitochondrial genome sequence of the tubeworm Lamellibrachia satsuma and structural conservation in the mitochondrial genome control regions of Order Sabellida. *Marine Genomics, 26*, 63-71. doi:10.1016/j.margen.2015.12.010

Pettibone, M. (1992). Contribution to the polychaete family Pholoidae Kinberg. *Smithsonian Contributions to Zoology*.

Pocklington, P., & Wells, P. G. (1992). Polychaetes Key taxa for marine environmental quality monitoring. *Marine Pollution Bulletin, 24*(12), 593-598. doi:10.1016/0025-326X(92)90278-E

Richter, S., Schwarz, F., Hering, L., Boggemann, M., & Bleidorn, C. (2015). The Utility of Genome Skimming for Phylogenomic Analyses as Demonstrated for Glycerid Relationships (Annelida, Glyceridae). *Genome Biology and Evolution, 7*(12), 3443-3462. doi:10.1093/gbe/evv224

Rodrigo, A. P., & Costa, P. M. (2019). The hidden biotechnological potential of marine invertebrates: The Polychaeta case study. *Environmental Research, 173*, 270-280. doi:10.1016/j.envres.2019.03.048

Rouse, G. W., & Fauchald, K. (1997). Cladistics and polychaetes. *Zoologica Scripta, 26*(2), 139-204. doi:10.1111/j.1463-6409.1997.tb00412.x

Rouse, G. W., & Pleijel, F. (2001). *Polychaetes*: Oxford University Press: Oxford, UK

Rousset, V., Pleijel, F., Rouse, G. W., Erseus, C., & Siddall, M. E. (2007). A molecular phylogeny of annelids. *Cladistics, 23*(1), 41-63. doi:10.1111/j.1096-0031.2006.00128.x

Rousset, V., Rouse, G. W., Feral, J. P., Desbruyeres, D., & Pleijel, F. (2003). Molecular and morphological evidence of Alvinellidae relationships (Terebelliformia, Polychaeta, Annelida). *Zoologica Scripta, 32*(2), 185-197. doi:DOI 10.1046/j.1463-6409.2003.00110.x

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics, 6*(2), 461-464.

Seixas, V. C., Russo, C. A., & Paiva, P. C. (2017). Mitochondrial genome of the Christmas tree worm Spirobranchus giganteus (Annelida: Serpulidae) reveals a high substitution rate among annelids. *Gene, 605*, 43-53. doi:10.1016/j.gene.2016.12.024

Struck, T. H., Golombek, A., Weigert, A., Franke, F. A., Westheide, W., Purschke, G., . . . Halanych, K. M. (2015). The Evolution of Annelids Reveals Two Adaptive Routes to the Interstitial Realm. *Current Biology, 25*(15), 1993-1999. doi:10.1016/j.cub.2015.06.007

Struck, T. H., Nesnidal, M. P., Purschke, G., & Halanych, K. M. (2008). Detecting possibly saturated positions in 18S and 28S sequences and their influence on phylogenetic reconstruction of Annelida (Lophotrochozoa). *Molecular Phylogenetics and Evolution, 48*(2), 628-645. doi:10.1016/j.ympev.2008.05.015

Struck, T. H., Paul, C., Hill, N., Hartmann, S., Hosel, C., Kube, M., . . . Bleidorn, C. (2011). Phylogenomic analyses unravel annelid evolution. *Nature, 471*(7336), 95-98. doi:10.1038/nature09864

Struck, T. H., & Purschke, G. (2005). The sister group relationship of Aeolosomatidae and Potamodrilidae (Annelida: "Polychaeta") — a molecular phylogenetic approach based on 18S rDNA and cytochrome oxidase I. *Zoologischer Anzeiger - A Journal of Comparative Zoology, 243*(4), 281-293. doi:10.1016/j.jcz.2005.01.001

Struck, T. H., Schult, N., Kusen, T., Hickman, E., Bleidorn, C., McHugh, D., & Halanych, K. M. (2007). Annelid phylogeny and the status of Sipuncula and Echiura. *BMC evolutionary biology, 7*(1), 57. doi:10.1186/1471-2148-7-57

Struck, T. H., Westheide, W., & Purschke, G. (2002). Progenesis in Eunicida ("Polychaeta," Annelida)—separate evolutionary events? Evidence from molecular data. *Molecular Phylogenetics and Evolution, 25*(1), 190-199. doi:10.1016/S1055-7903(02)00231-2

Tamura, K., & Nei, M. (1993). Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Molecular Biology and Evolution, 10*(3), 512-526. doi:10.1093/oxfordjournals.molbev.a040023

Valles, Y., Halanych, K. M., & Boore, J. L. (2008). Group II introns break new boundaries: presence in a bilaterian's genome. *PloS one, 3*(1), e1488-e1488. doi:10.1371/journal.pone.0001488

Weigert, A., & Bleidorn, C. (2016). Current status of annelid phylogeny. *Organisms Diversity & Evolution, 16*(2), 345-362. doi:10.1007/s13127-016-0265-7

Weigert, A., Golombek, A., Gerth, M., Schwarz, F., Struck, T. H., & Bleidorn, C. (2016). Evolution of mitochondrial gene order in Annelida. *Molecular Phylogenetics and Evolution, 94*, 196-206. doi:10.1016/j.ympev.2015.08.008

Weigert, A., Helm, C., Meyer, M., Nickel, B., Arendt, D., Hausdorf, B., . . . Struck, T. H. (2014). Illuminating the Base of the Annelid Tree Using Transcriptomics. *Molecular Biology and Evolution, 31*(6), 1391-1401. doi:10.1093/molbev/msu080

Wiklund, H., Nygren, A., Pleijel, F., & Sundberg, P. (2005). Phylogeny of Aphroditiformia (Polychaeta) based on molecular and morphological data. *Molecular Phylogenetics and Evolution, 37*(2), 494-502. doi:10.1016/j.ympev.2005.07.005

Winnepenninckx, B., Backeljau, T., & De Wachter, R. (1995). Phylogeny of protostome worms derived from 18S rRNA sequences. *Molecular Biology and Evolution, 12*(4), 641-649. doi:10.1093/oxfordjournals.molbev.a040243

Wolstenholme, D. R. (1992). Animal Mitochondrial DNA: Structure and Evolution. In D. R. Wolstenholme & K. W. Jeon (Eds.), *International Review of Cytology* (Vol. 141, pp. 173-216): Academic Press.

Zhang, Y., Sun, J., Rouse, G. W., Wiklund, H., Pleijel, F., Watanabe, H. K., . . . Qiu, J. W. (2018). Phylogeny, evolution and mitochondrial gene order rearrangement in scale worms (Aphroditiformia, Annelida). *Molecular Phylogenetics and Evolution, 125*, 220-231. doi:10.1016/j.ympev.2018.04.002

Zhong, M., Struck, T. H., & Halanych, K. M. (2008). Phylogenetic information from three mitochondrial genomes of Terebelliformia (Annelida) worms and duplication of the methionine tRNA. *Gene, 416*(1), 11-21. doi:https://doi.org/10.1016/j.gene.2008.02.020

Appendix

```
#for testing
#for number in 2 11


for number in  2 3 4 10 11 12 13 17 19 20 24 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 26 27 29 33 35 36 38 44

do
        seed=Seeds_CO1_Goniada_japonica.fasta


#Seeds_CO1_Aricidea_catherinae.fasta
#Seeds_CO1_Aphrodita_aculeata.fasta
#Seeds_CO1_Owenia_fusiformis
#Seeds_everything.fasta
#Seed_RNA_all.fasta
#Seed_RNA.fasta
#Seeds_a_trimmed

        r1=$number'_S'$number'_noreadthrough_R1.fastq'
        r2=$number'_S'$number'_noreadthrough_R2.fastq'

        sed 's/Seed_1/Seeds_CO1_Goniada_japonica/g' config_mito_test.txt > output_1

        sed 's/1_S1_noreadthrough_R1.fastq/'"$r1"'/g' output_1 > config_$number

#        sed 's/1_S1_noreadthrough_R1.fastq/'"$r1"'/g' output_1 > config_BOLD_$number

#        sed 's/1_S1_noreadthrough_R1.fastq/'"$r1"'/g' output_1 > config_NCBI_$number

#       sed 's/1_S1_noreadthrough_R1.fastq/'"$r1"'/g' output_1 > configs_$number

        sed -i 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' config_$number

#        sed -i 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' config_BOLD_$number

#       sed -i 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' config_NCBI_$number

#        sed -i 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' configs_$number

done
```

config2.sh

A.1 Shell script for generating configuration files for NOVOPlasty in bulk

```
for number in  2 3 4 10 11 12 13 17 19 20 24 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 26 27 29 33 35 36 38 44

do
#       perl NOVOPlasty3.7.pl -c config_BOLD_$number

#        perl NOVOPlasty3.7.pl -c config_NCBI_$number

         perl NOVOPlasty3.7.pl -c config_$number

#        perl NOVOPlasty3.7.pl -c configs_$number
done
```

assembly.sh (END)

A.2 Shell script for assembly for NOVOPlasty in bulk

```
for Seed in My_n
do
        seed=$Seed

        sed 's/Seed_1/'"$seed"'/g' config_mito_test.txt > $Seed'_output'



        for reference in Ow_C
        do
                reference=$reference
                sed 's/Reference_1/'"$reference"'/g' $Seed'_output' > $Seed'_re_'$reference'_fer'


                for k_mer in 23
# 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
                do
                        k_mer=$k_mer

                        sed 's/k_mer/'"$k_mer"'/g' $Seed'_re_'$reference'_fer' >  $Seed'_refer_'$reference'_k_'$k_mer'_mer'


#                       for number in 2 24 10 26
                        for number in 47
# 3 4 11 12 13 17 19 20  28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
                        do
                                r1=$number'_S'$number'_noreadthrough_R1.fastq'
                                r2=$number'_S'$number'_noreadthrough_R2.fastq'


                                sed 's/1_S1_noreadthrough_R1.fastq/'"$r1"'/g'  $Seed'_refer_'$reference'_k_'$k_mer'_mer'  >  $Seed'_refer_'$reference'_k_'$k_mer'_mer_''config_'$number

                                sed -i 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' $Seed'_refer_'$reference'_k_'$k_mer'_mer_''config_'$number

                                sed -i 's/test_2/'$Seed'_ref_'$reference'_k_'$k_mer'_config_'$number'/g' $Seed'_refer_'$reference'_k_'$k_mer'_mer_''config_'$number


                         perl NOVOPlasty3.7.pl -c $Seed'_refer_'$reference'_k_'$k_mer'_mer_''config_'$number

                        mv Contigs_1_$Seed'_ref_'$reference'_k_'$k_mer'_config_'$number.fasta topology=circular_$Seed'_'$k_mer'_'$reference'_'$number.fasta

                        mv Uncircularized_assemblies_1_$Seed'_ref_'$reference'_k_'$k_mer'_config_'$number.fasta topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_U1'.fasta

                        mv Uncircularized_assemblies_2_$Seed'_ref_'$reference'_k_'$k_mer'_config_'$number.fasta topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_U2'.fasta

                        mv Uncircularized_assemblies_3_$Seed'_ref_'$reference'_k_'$k_mer'_config_'$number.fasta topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_U3'.fasta

                        mv Circularized_assembly_1_$Seed'_ref_'$reference'_k_'$k_mer'_config_'$number.fasta topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_C'.fasta
happiness shao 2.sh
                         outprefix=$Seed'_'$k_mer'_'$reference'_'$number
                         r1=topology=circular_$Seed'_'$k_mer'_'$reference'_'$number.fasta

                         sed 's/Seed_1/'"$outprefix"'_Annelida/g' mitoz_annotate_config.txt > output_1

                         sed 's/1_S1_noreadthrough_R2.fastq/'"$r1"'/g' output_1 > $Seed'_'$k_mer'_config_annotate_'$reference'_'$number


                        python3 MitoZ_Annelida.py annotate --config $Seed'_'$k_mer'_config_annotate_'$reference'_'$number




                         outprefix=$Seed'_'$k_mer'_'$reference'_'$number'_U1'
                         r2=topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_U1'.fasta

                         sed 's/Seed_1/'"$outprefix"'_Annelida/g' mitoz_annotate_config.txt > output_2

                         sed 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' output_2 > $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_U1'


                        python3 MitoZ_Annelida.py annotate --config $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_U1'




                         outprefix=$Seed'_'$k_mer'_'$reference'_'$number'_U2'
                         r2=topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_U2'.fasta

                         sed 's/Seed_1/'"$outprefix"'_Annelida/g' mitoz_annotate_config.txt > output_3

                         sed 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' output_3 > $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_U2'


                        python3 MitoZ_Annelida.py annotate --config $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_U2'



                         outprefix=$Seed'_'$k_mer'_'$reference'_'$number'_U3'
                         r2=topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_U3'.fasta

                         sed 's/Seed_1/'"$outprefix"'_Annelida/g' mitoz_annotate_config.txt > output_4

                         sed 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' output_4 > $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_U3'


                        python3 MitoZ_Annelida.py annotate --config $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_U3'

:
```

```
                        outprefix=$Seed'_'$k_mer'_'$reference'_'$number'_C'
                        r2=topology=circular_$Seed'_'$k_mer'_'$reference'_'$number'_C'.fasta

                        sed 's/Seed_1/'"$outprefix"'_Annelida/g' mitoz_annotate_config.txt > output_5

                        sed 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' output_5 > $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_C'


                       python3 MitoZ_Annelida.py annotate --config $Seed'_'$k_mer'_config_annotate_'$reference'_'$number'_C'

                       done

               done
       done

done

rm $Seed'_output'

rm $Seed'_re_'$reference'_fer'

rm $Seed'_refer_'$reference'_k_'$k_mer'_mer'

rm -f $Seed'_refer_'$reference'_k_'$k_mer'_mer_''config_'$number

rm Assembled_reads_*

rm contigs_tmp_*
```

A.3 Shell script for Test 9 the "The ultimate weapon" which run "244188 possibilities"

First use NOVOPlasty for assembly then use MitoZ for annotation. This script connected 2 tools together.

```
#for testing
#for number in 2 11


for number in 10
#2 3 4 10 11 12 13 17 19 20 24 28 32 34 37 39 40 41 42  43 47 1 14 15 18 21 22 23 26 27 29 33 35 36 38 44


do
        outprefix=$number'_1x_Annelida'
        r1=$number'_S'$number'_topology=circular_R1.fastq'
        r2=$number'_S'$number'_topology=circular_R2.fastq'

        sed 's/Seed_1/'"$outprefix"'/g' mitoz_all_config.txt > output_1

        sed 's/1_S1_noreadthrough_R1.fastq/'"$r1"'/g' output_1 > config_$number

        sed -i 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' config_$number

done
MitoZ config all.sh (END)
```

A.4 Shell script for generating configuration files for MitoZ all mode in bulk

```
#for testing
#for number in 2 11


for number in 2 3 4 10 11 12 13 17 19 20 24 28 32 34 37 39 40 41 42  43 47 1 14 15 18 21 22 23 26 27 29 33 35 36 38 44


do
        outprefix=$number'_1x_clean'
        r1=$number'_S'$number'_clean.1.fq.gz'
        r2=$number'_S'$number'_clean.2.fq.gz'

        sed 's/Seed_1/banana_'"$outprefix"'/g' mitoz_all2_config.txt > output_1

        sed 's/1_S1_noreadthrough_R1.fastq/'"$r1"'/g' output_1 > configs_$number

        sed -i 's/1_S1_noreadthrough_R2.fastq/'"$r2"'/g' configs_$number

done
MitoZ_config_all2.sh (END)
```

A.5 Shell script for generating configuration files for MitoZ all2 mode in bulk

```
for number in 10
# 2 3 4 10 11 12 13 17 19 20 24 28 32 34 37 39 40 41 42  43 47 1 14 15 18 21 22 23 26 27 29 33 35 36 38 44
do
        outprefix=$number'_S10_GJ_clean'
        r1=topology=circular_$number'_S10_Gj_clean'.fasta

        sed 's/Seed_1/'"$outprefix"'_Annelida/g' mitoz_annotate_config.txt > output_1

        sed  's/1_S1_noreadthrough_R2.fastq/'"$r1"'/g' output_1 > config_annotate_$number'_S10_Gj_clean'


done



MitoZ_config_annotiation.sh (END)
```

A.6 Shell script for generating configuration files for MitoZ annotation mode in bulk

```
for number in 2 3 4 10 11 12 13 17 19 20 24 28 32 34 37 39 40 41  42  43 47 1 14 15 18 21 22   23 26 27  29 33 35 36 38 44

do

#       python3 MitoZ_Annelida.py all2 --config configs_$number

#       python3 MitoZ_Annelida.py all --config config_BOLD_$number

         python3 MitoZ_Annelida_37.py all --config config_$number


#        python3 MitoZ_Annelida.py annotate --config config_annotate_$number'_39'

done
annotation.sh (END)
```

A.7 Shell script for assembly for MitoZ in all and all 2 mode as well as for annotation mode in bulk

```
for number in 30 31 45 46
#2 3 4 10 11 12 13 17 19 20 24 28 32 34 37 39 40 41  42  43 47 1 14 15 18 21 22   23 26 27  29 33 35 36 38 44
do
      /usr/bin/python interleave-fastqgz-MITOBIM.py "$number"_S"$number"_noreadthrough_R1.fastq  "$number"_S"$number"_noreadthrough_R2.fastq >  "$number"_S"$number"_noreadthrough_interleav
ed.fastq
done


#interleave-fastqgz-MITOBIM.py xxxxx_R1.fastq xxxxx_R2.fastq > xxxxx_interleaved.fastq
R1R2_1.sh (END)
```

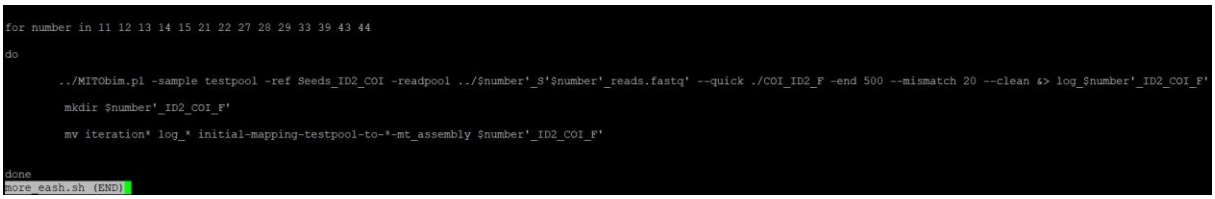
A.8 Shell script for interleaving the forward R1and reverse R2 reads into one fastq file in bulk

```
for number in   3 4 11 12 13 17 19 20 28 32 34 37 39 40 41  42  43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
do
       cp ../$number'_S'$number'_reads.fastq' ./
       cp ../$1 ./

       mv $1 reference.fa

#      echo -e "\n#manifest file for basic mapping assembly with illumina data using MIRA 4\n\nproject = initial-mapping-testpool-to-Salpinus-mt\n\njob=genome,mapping,accurate\n\nparameter
s = -NW:mrnl=0 -AS:nop=1 SOLEXA_SETTINGS -CO:msr=no\n\nreadgroup\nis_reference\ndata = reference.fa\nstrain = Salpinus-mt-genome\n\nreadgroup = reads\ndata = $number_S$number_reads.fastq\nt
echnology = solexa\nstrain = testpool\n" > manifest.conf_$number

       sed -i -e 's/Salpinus/'"$2"'/g' -e 's/banana/'"$number"'_S'"$number"'_reads.fastq/g' manifest.conf

       export LC_ALL=C

       mira manifest.conf

       ../MITObim.pl -start 1 -end 200 -sample testpool -ref $2_mt_genome -readpool $number'_S'$number'_reads.fastq' -maf initial-mapping-testpool-to-$2-mt_assembly/initial-mapping-testpoo
l-to-$2-mt_d_results/initial-mapping-testpool-to-$2-mt_out.maf --mismatch 20 --clean  &> log_$number'_T1'

       mkdir $number'_T1_Of'
       mv iteration* log_* initial-mapping-testpool-to-*-mt_assembly  $number'_T1_Of'
done
more_easy.sh (END)
```

A.9 Shell script for running T1 mode for MITObim in bulk

```
#for number in 2 10 24 26
for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
#       ./MITObim.pl -start 1 -end 100 -sample testpool -ref Owenia_fusiformis_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Owenia_fusiformis_mitochondrion_complete_geno
me.fasta --mismatch 20  --clean &> log_$number

#       ./MITObim.pl -start 1 -end 100 -sample testpool -ref Goniada_japonica_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Goniada_japonica.fasta
--mismatch 20  --clean &> log_$number'_Gj'
#               mkdir Gj_done_$number
#                       mv iteration* Gj_done_$number
#done

#do
 #      ./MITObim.pl -start 1 -end 100 -sample testpool -ref Owenia_fusiformis_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Owenia_fusiformis_mitochondrion_complete_geno
me.fasta --mismatch 20  --clean &> log_$number'_Of'
#               mkdir Of_done_$number
 #                      mv iteration* Of_done_$number
#done

#do
#       ./MITObim.pl -start 1 -end 100 -sample testpool -ref Orbinia_latreillii_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Orbinia_latreillii.fa
sta --mismatch 20  --clean &> log_$number'_Ol'
#               mkdir Ol_done_$number
 #                      mv iteration* Ol_done_$number
#done

#do
#       ./MITObim.pl -start 1 -end 100 -sample testpool -ref Pista_cristata_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Pista_cristata.fasta --mis
match 20  --clean &> log_$number'_Pc'
#               mkdir Pc_done_$number
 #                      mv iteration* Pc_done_$number
#done

#do
#       ./MITObim.pl -start 1 -end 100 -sample testpool -ref Spirobranchus_giganteus_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Spirobranchus_gi
ganteus.fasta --mismatch 20  --clean &> log_$number'_Sg'
#               mkdir Sg_done_$number
 #                      mv iteration* Sg_done_$number
#done

#do
#       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Terebellides_stroemii_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Terebellides_stroe
mii.fasta  --mismatch 20  --clean &> log_$number'_Ts'
#               mkdir Ts_done_$number
 #                      mv iteration* Ts_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
mitobim II.sh
#do
#       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Nephtys_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Nephtys.fasta  --mismatch 20  --c
lean &> log_$number'_Nep'
 #              mkdir Nep_done_$number
  #                     mv iteration* Nep_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
 #       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Eclysippe_vanelli_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_partial_genome_Eclysippe_vanelli.fasta
  --mismatch 20  --clean &> log_$number'_Ev'
  #             mkdir Ev_done_$number
   #                    mv iteration* Ev_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
 #       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Scoloplos_armiger_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_partial_genome_Scoloplos_armiger.fasta
  --mismatch 20  --clean &> log_$number'_Sa'
  #             mkdir Sa_done_$number
   #                    mv iteration* Sa_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
 #       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Namalycastis_abiuma_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Namalycastis_abiuma.
fasta  --mismatch 20  --clean &> log_$number'_Na'
  #             mkdir Na_done_$number
   #                    mv iteration* Na_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
 #       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Tylorrhynchus_heterochaetus_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Tylorrhynchu
s_heterochaetus.fasta  --mismatch 20  --clean &> log_$number'_Th'
  #             mkdir Th_done_$number
   #                    mv iteration* Th_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
 #       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Trypanosyllis_Trypanobia_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Trypanosyllis_T
rypanobia.fasta  --mismatch 20  --clean &> log_$number'_Tt'
  #             mkdir Tt_done_$number
   #                    mv iteration* Tt_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
#do
#
 #       ./MITObim.pl -start 1 -end 200 -sample testpool -ref Eurythoe_complanata_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./Seeds_complete_genome_Eurythoe_complanata.
fasta  --mismatch 20  --clean &> log_$number'_Ec'
  #             mkdir Ec_done_$number
   #                    mv iteration* Ec_done_$number
#done

#for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
do
       ./MITObim.pl -start 1 -end 200 -sample testpool -ref ID2_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./ID2  --mismatch 20  --clean &> log_$number'_ID2'
              mkdir ID2_done_$number
                     mv iteration* ID2_done_$number
done


for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
do
       ./MITObim.pl -start 1 -end 200 -sample testpool -ref ID10_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./ID10  --mismatch 20  --clean &> log_$number'_ID10'
              mkdir ID10_done_$number
                     mv iteration* ID10_done_$number
done


for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
do
       ./MITObim.pl -start 1 -end 200 -sample testpool -ref ID24_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./ID24  --mismatch 20  --clean &> log_$number'_ID24'
              mkdir ID24_done_$number
                     mv iteration* ID24_done_$number
done


for number in 3 4 11 12 13 17 19 20 28 32 34 37 39 40 41 42 43 47 1 14 15 18 21 22 23 27 29 33 35 36 38 44
do
       ./MITObim.pl -start 1 -end 200 -sample testpool -ref ID26_mt_genome -readpool ./$number'_S'$number'_'reads.fastq --quick ./ID26  --mismatch 20  --clean &> log_$number'_ID26'
              mkdir ID26_done_$number
                     mv iteration* ID26_done_$number
done

(END)
```

A.10 Shell script for running Test 1 & 4 using T2 mode for MITObim in bulk

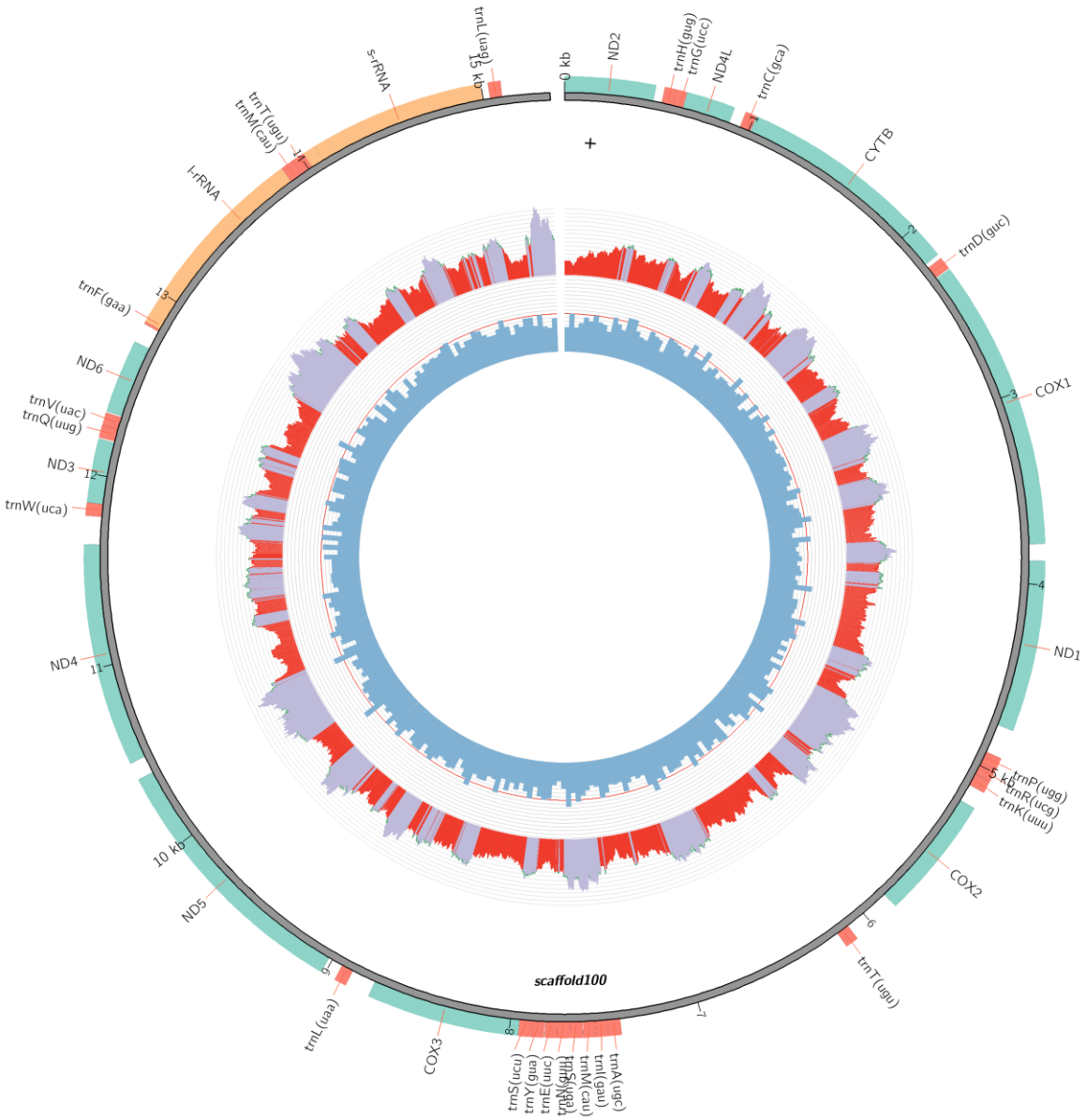


```
for number in 11 12 13 14 15 21 22 27 28 29 33 39 43 44

do

       ../MITObim.pl -sample testpool -ref Seeds_ID2_COI -readpool ../$number'_S'$number'_reads.fastq' --quick ./COI_ID2_F -end 500 --mismatch 20 --clean &> log_$number'_ID2_COI_F'

        mkdir $number'_ID2_COI_F'

        mv iteration* log_* initial-mapping-testpool-to-*-mt_assembly $number'_ID2_COI_F'


done
more_eash.sh (END)
```

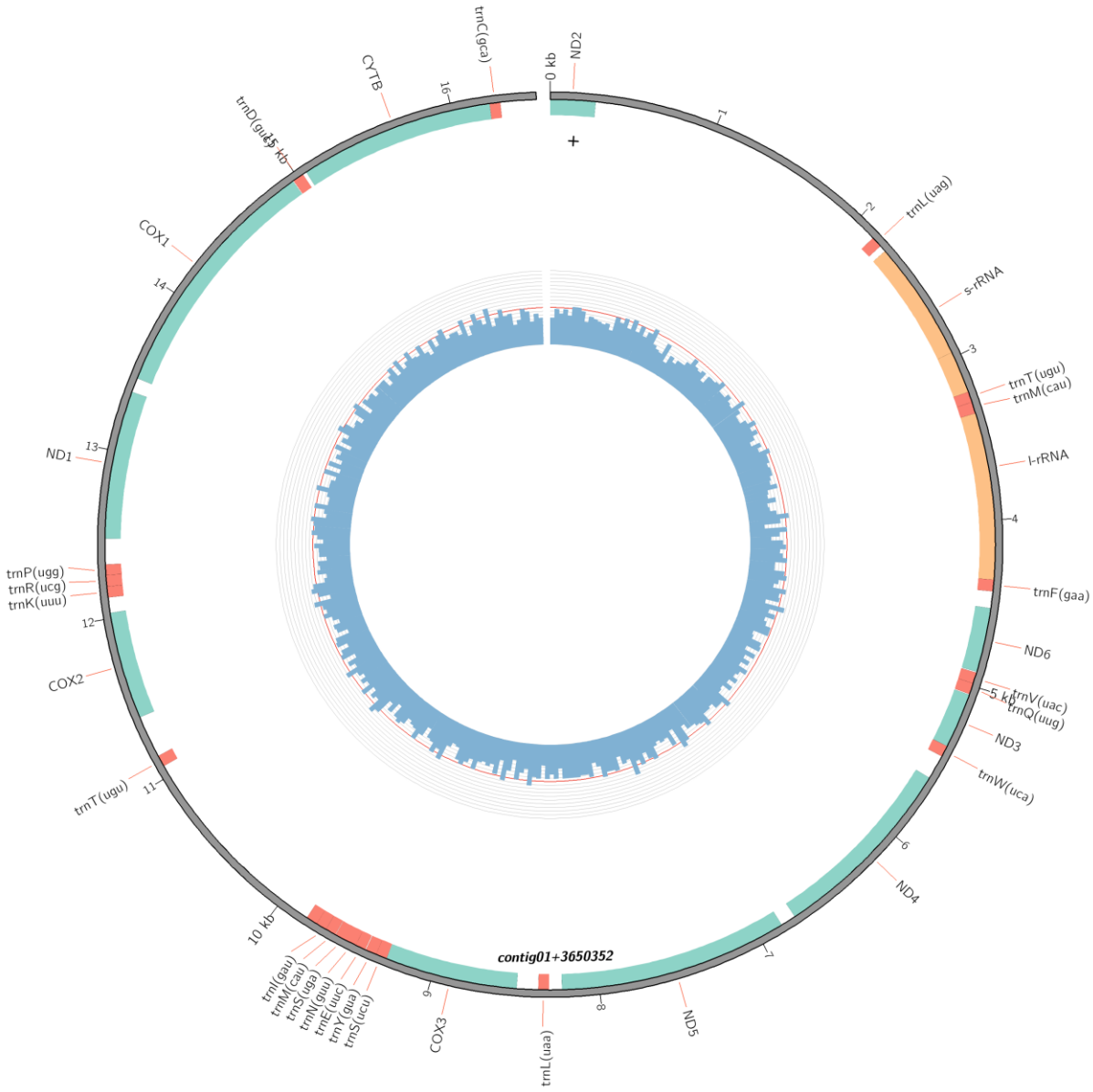
A.11 Shell script for running T3 mode for MITObim in bulk

```
cp $1 /spygenedata/polychaete_project/test_Mitoz

mv annotation_mitoz1.sh ../../../../

cd /spygenedata/polychaete_project/test_Mitoz

mv $1 topology=circular_$2


sed 's/_bb/ /g' topology=circular_$2 > output_mitobim

sed -e 's/X/n/g' -e 's/x/n/g' output_mitobim > topology=circular_$2_F




outprefix=$2_MitoBim
r1=topology=circular_$2_F

   sed 's/Seed_1/'"$outprefix"'_Annelida/g' mitoz_annotate_config.txt > output_mitobim1

   sed 's/1_S1_noreadthrough_R2.fastq/'"$r1"'/g' output_mitobim1 > config_annotate_$2_MitoBim

        source activate mitozEnv

                        python3 MitoZ_Annelida.py annotate --config config_annotate_$2_MitoBim

                                rm output_mitobim*
#                               rm config_annotate_$2_MitoBim

annotation_mitoz1.sh (END)
```

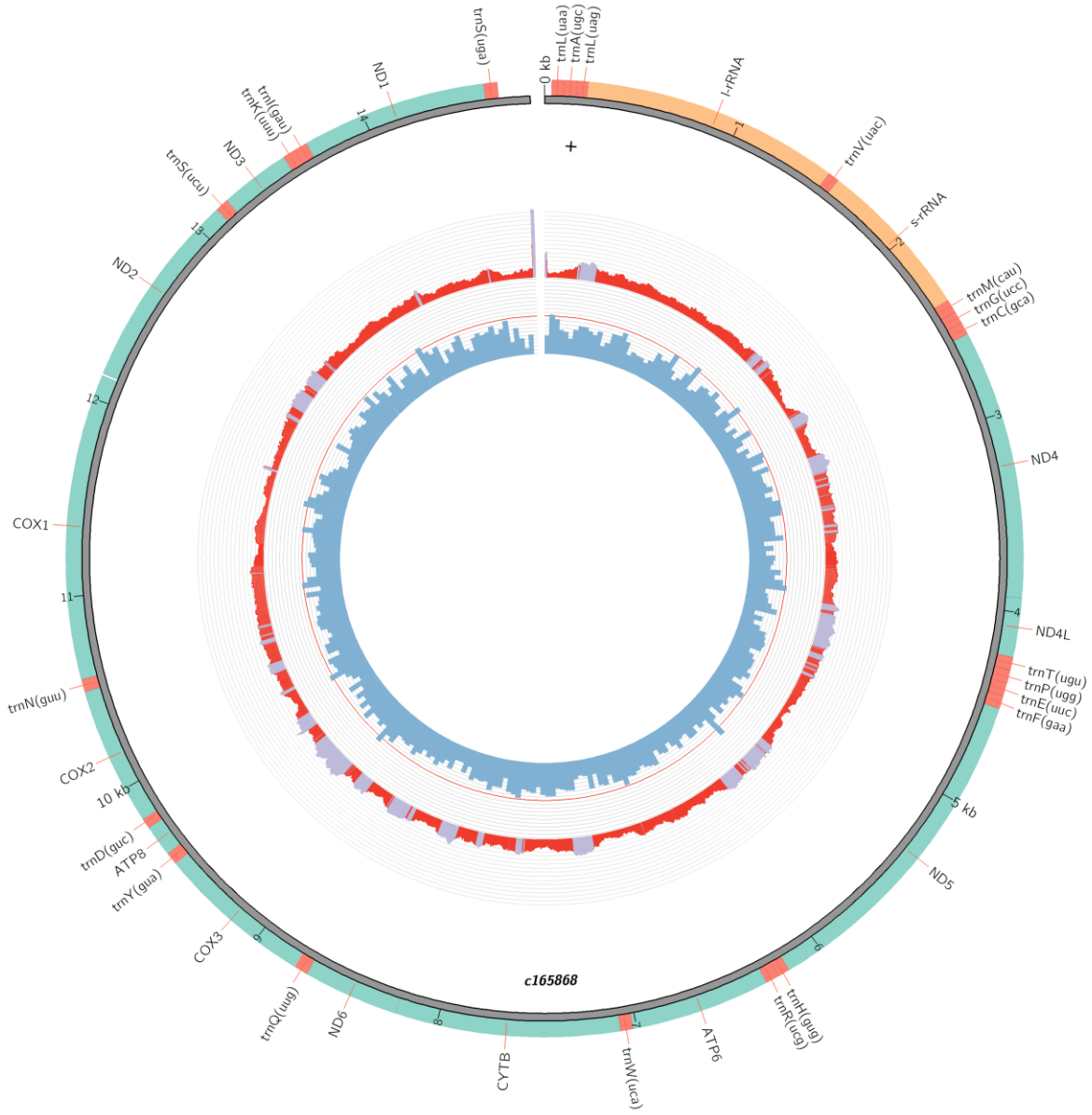
A.12 Shell script for using MitoZ to annotate the results that generated by MITObim



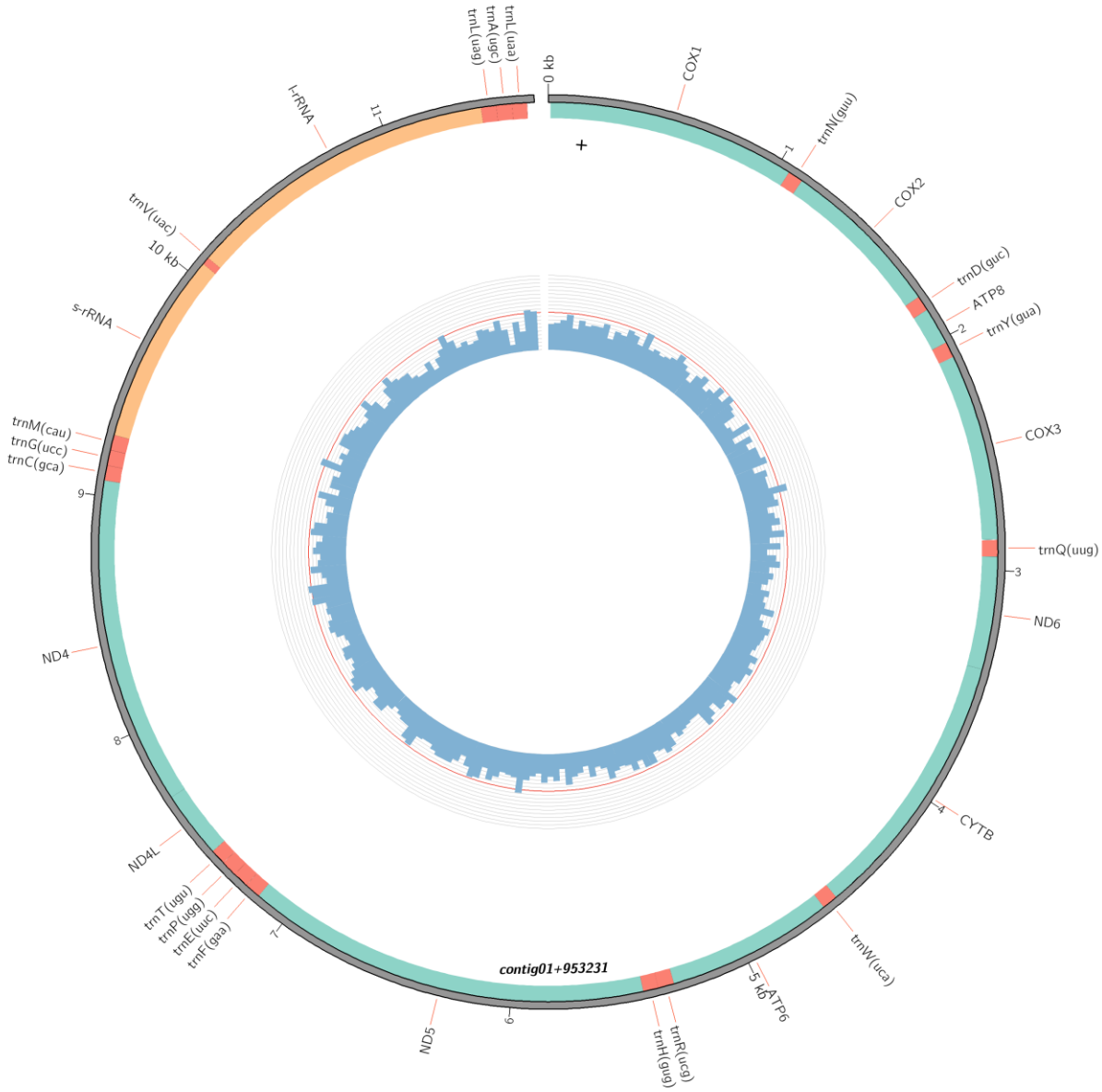
A.13 Nearly completed mitochondrial genome generated by MitoZ for ID 1. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution with green color as the outline, depth lower than the minimum value (default 20) showed in red, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



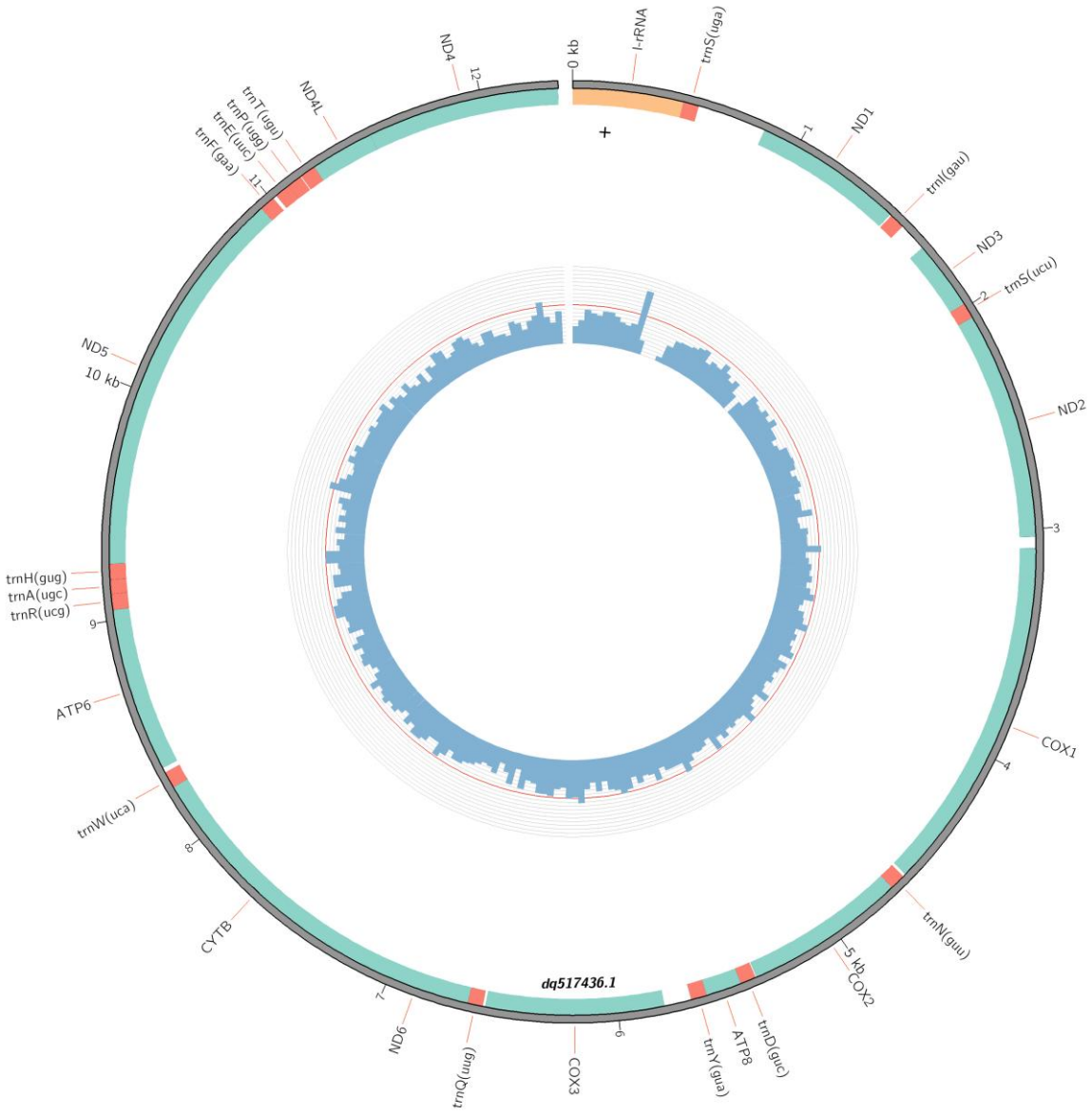
A.14 Nearly completed mitochondrial genome generated by NOVOPlasty for ID 1. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green..



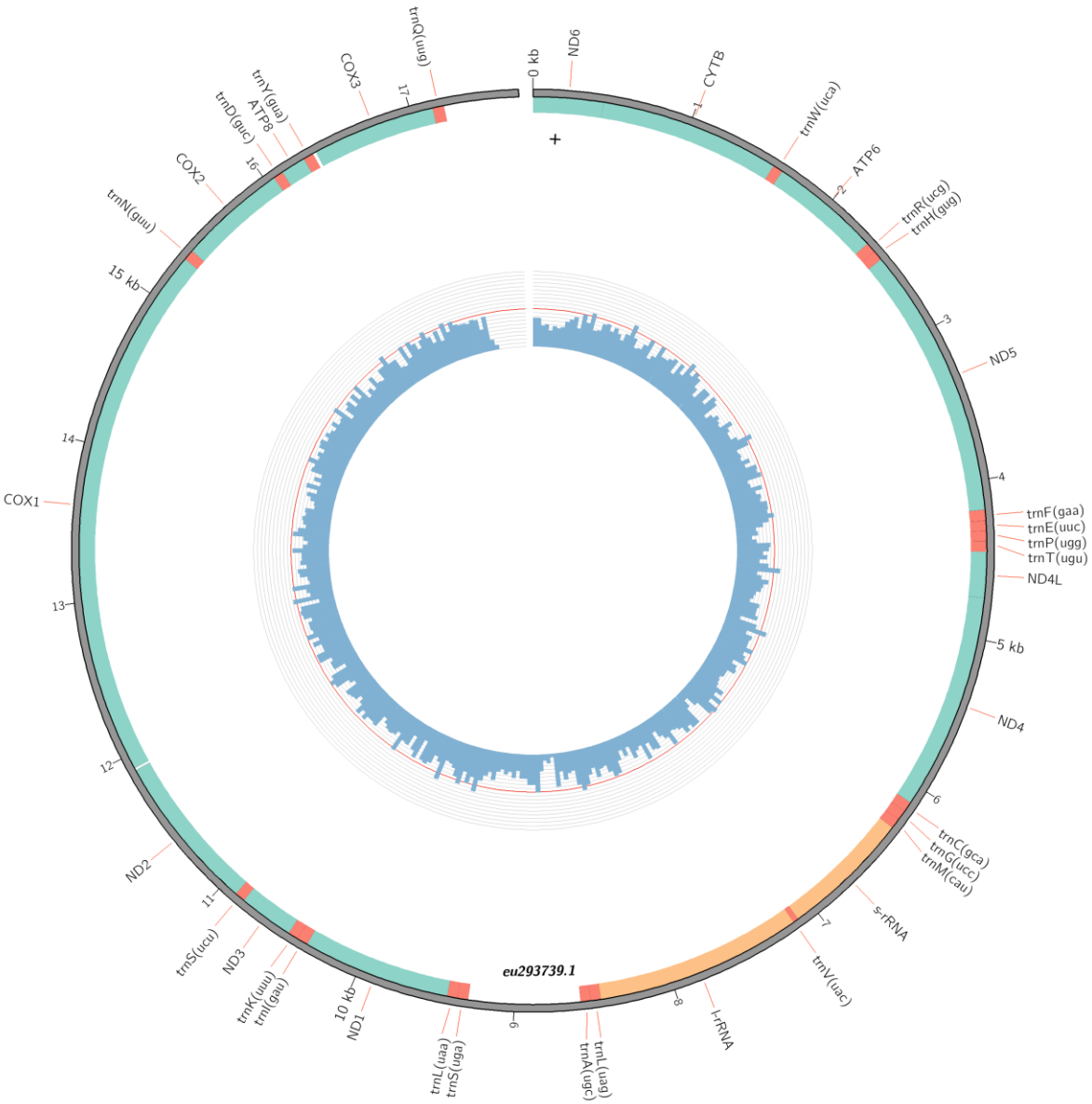
A.15 Nearly completed mitochondrial genome generated by MitoZ for ID10. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution with green color as the outline, depth lower than the minimum value (default 20) showed in red, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



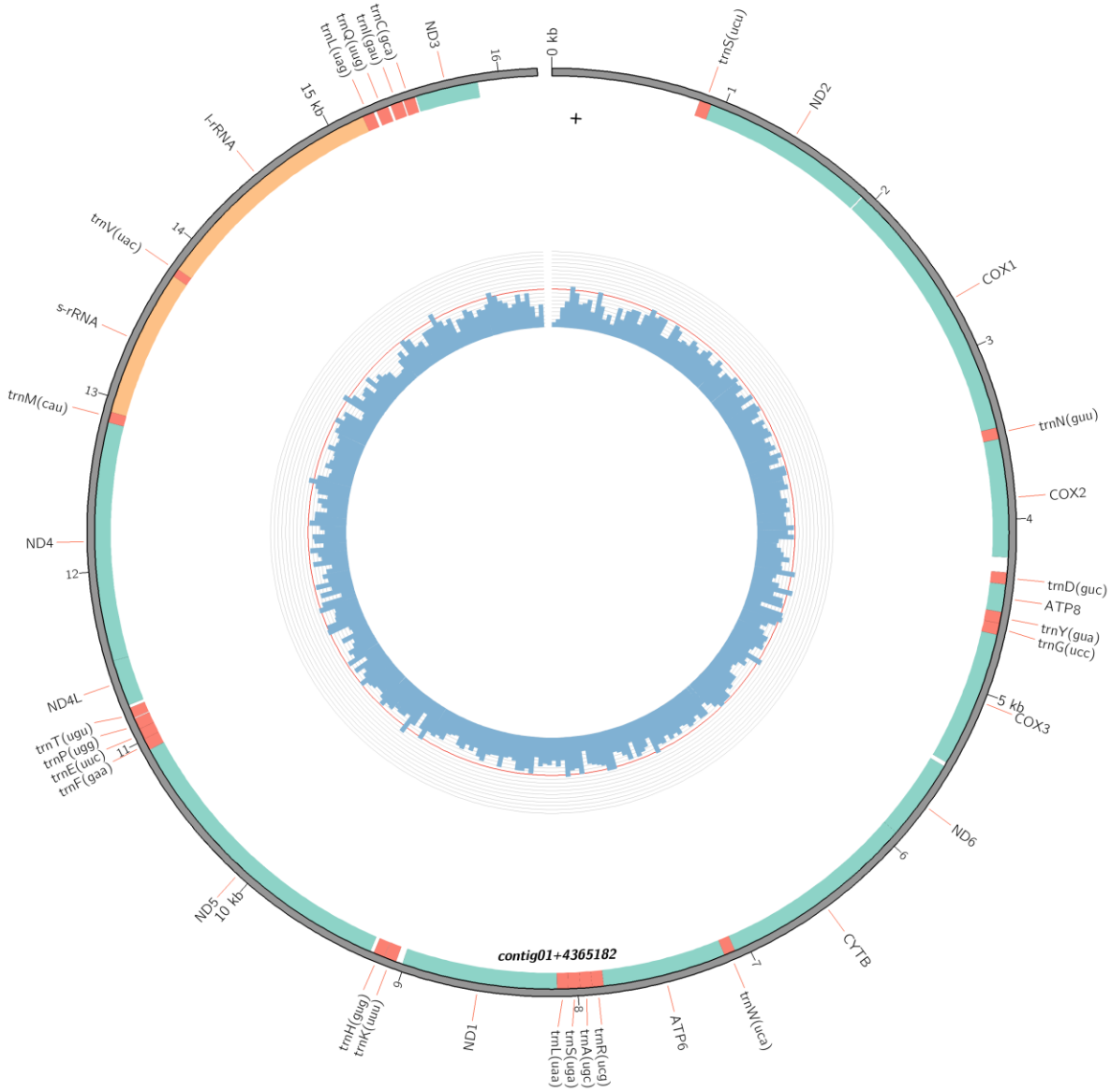
A.16 Nearly completed mitochondrial genome generated by NOVOPlasty for ID10. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



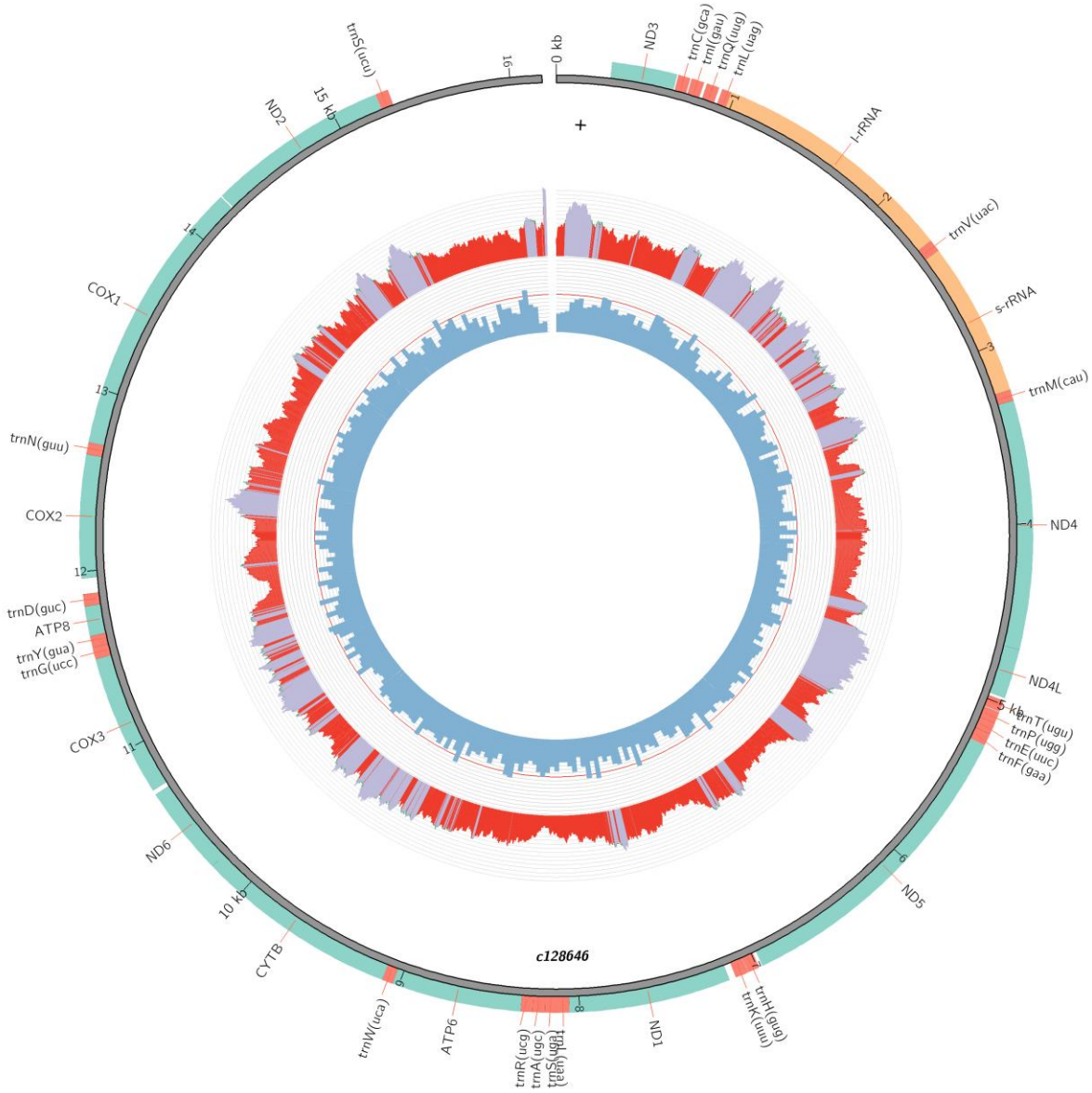
A.17 Nearly completed mitochondrial genome generated by MITObim for ID 17. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



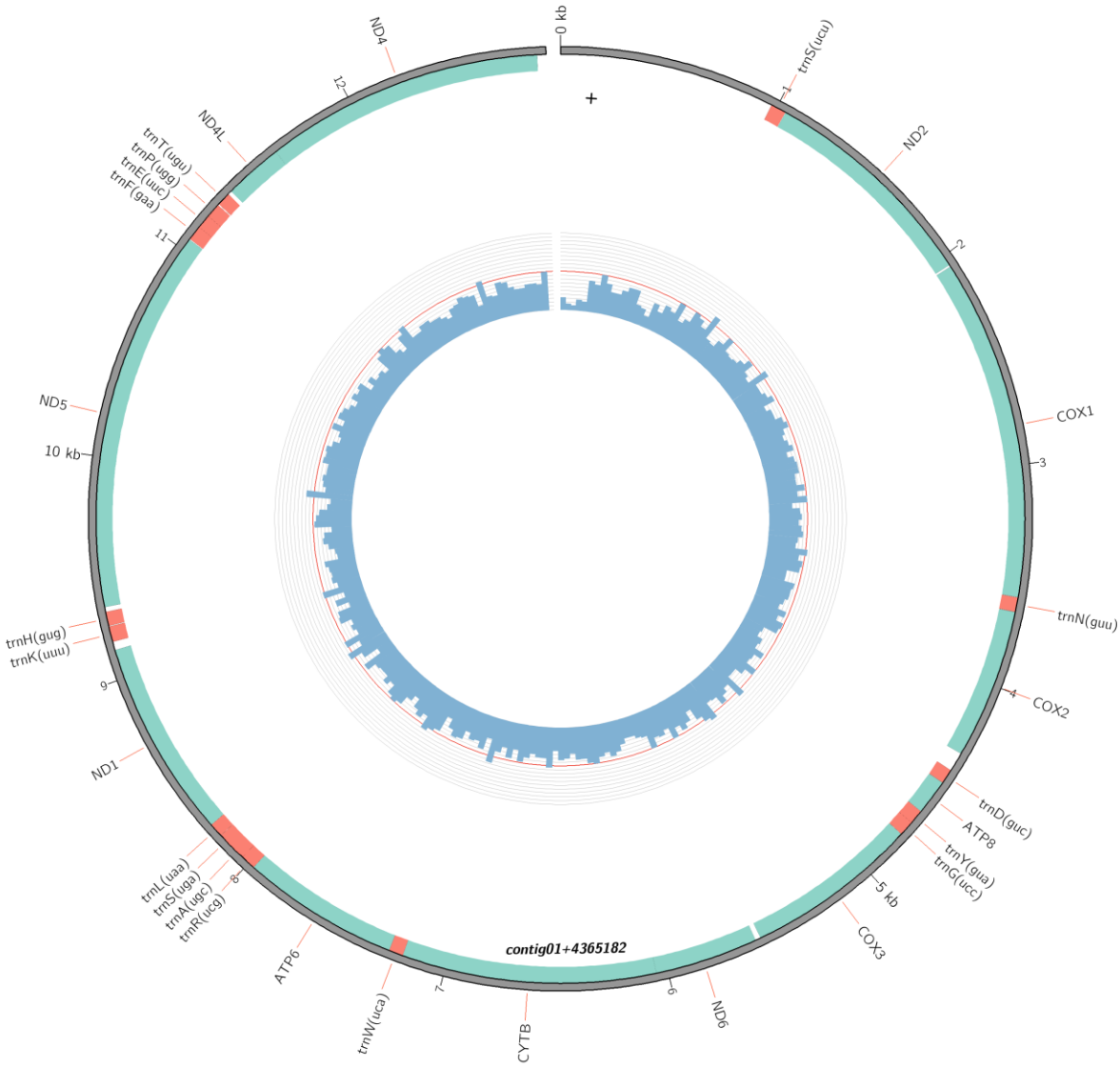
A.18 Nearly completed mitochondrial genome generated by MITObim for ID 20. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



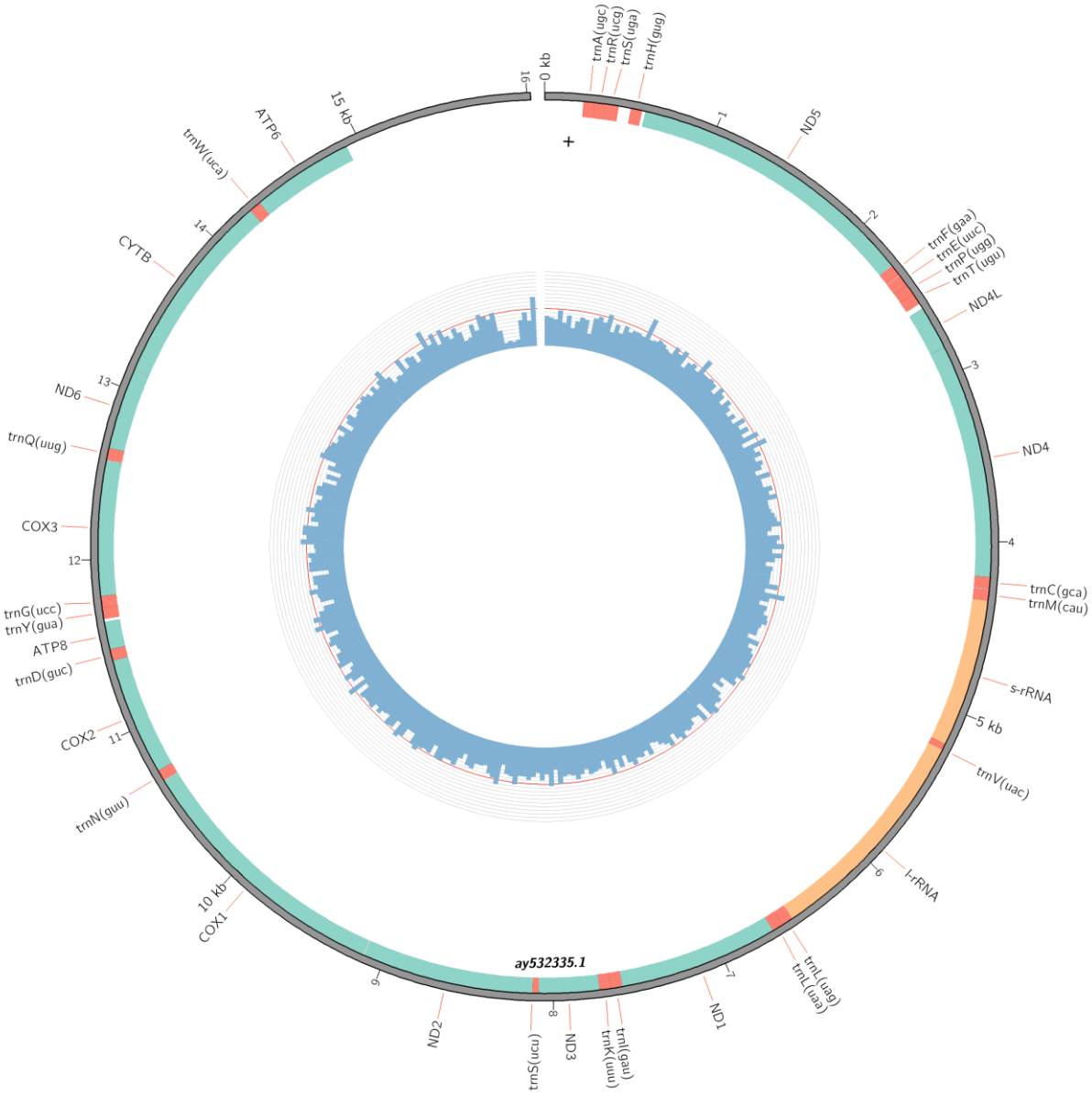
A.19 Nearly completed mitochondrial genome generated by NOVOPlasty for ID 24. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



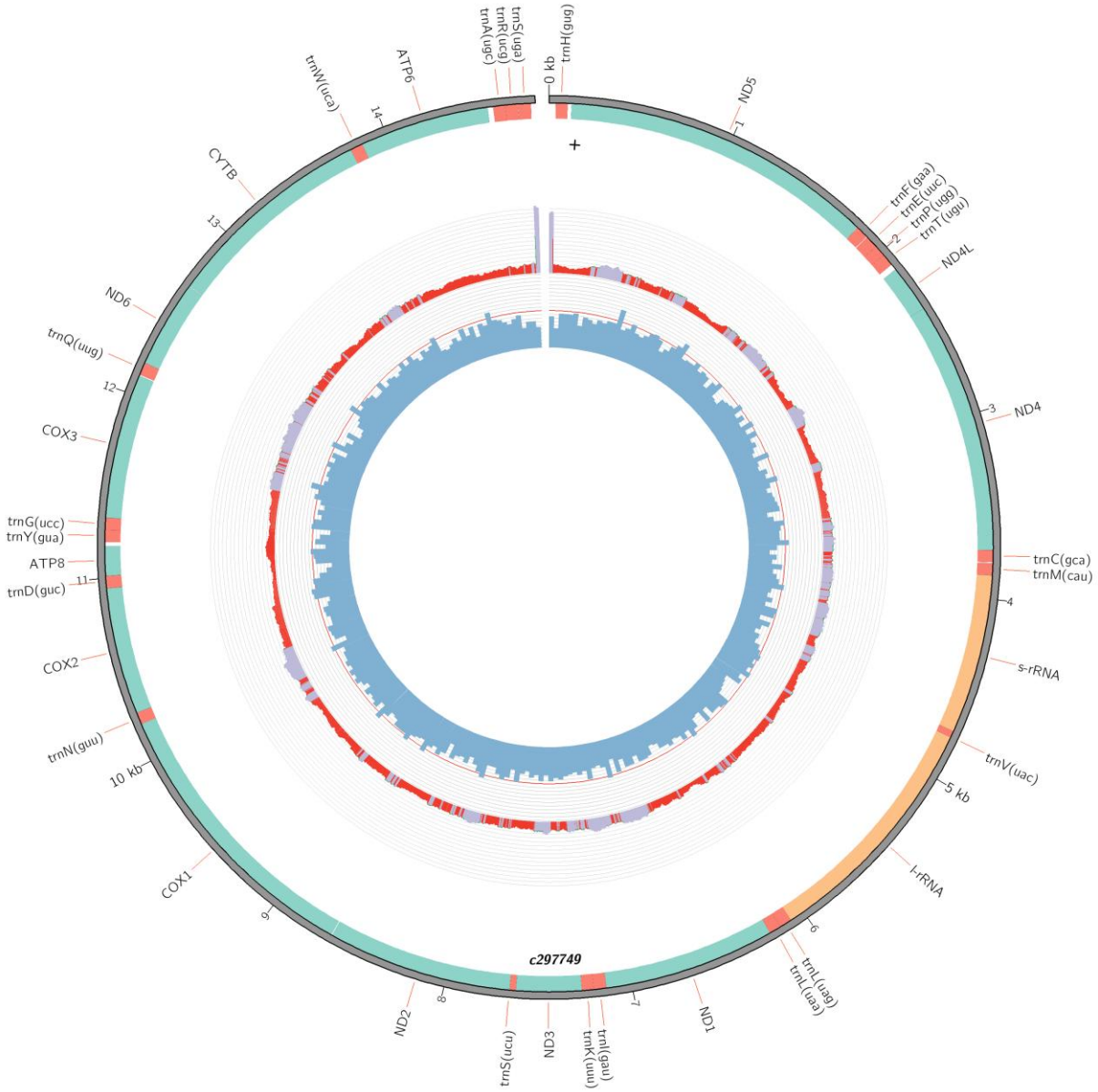
A.20 Nearly completed mitochondrial genome generated by MitoZ for ID 24. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution with green color as the outline, depth lower than the minimum value (default 20) showed in red, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



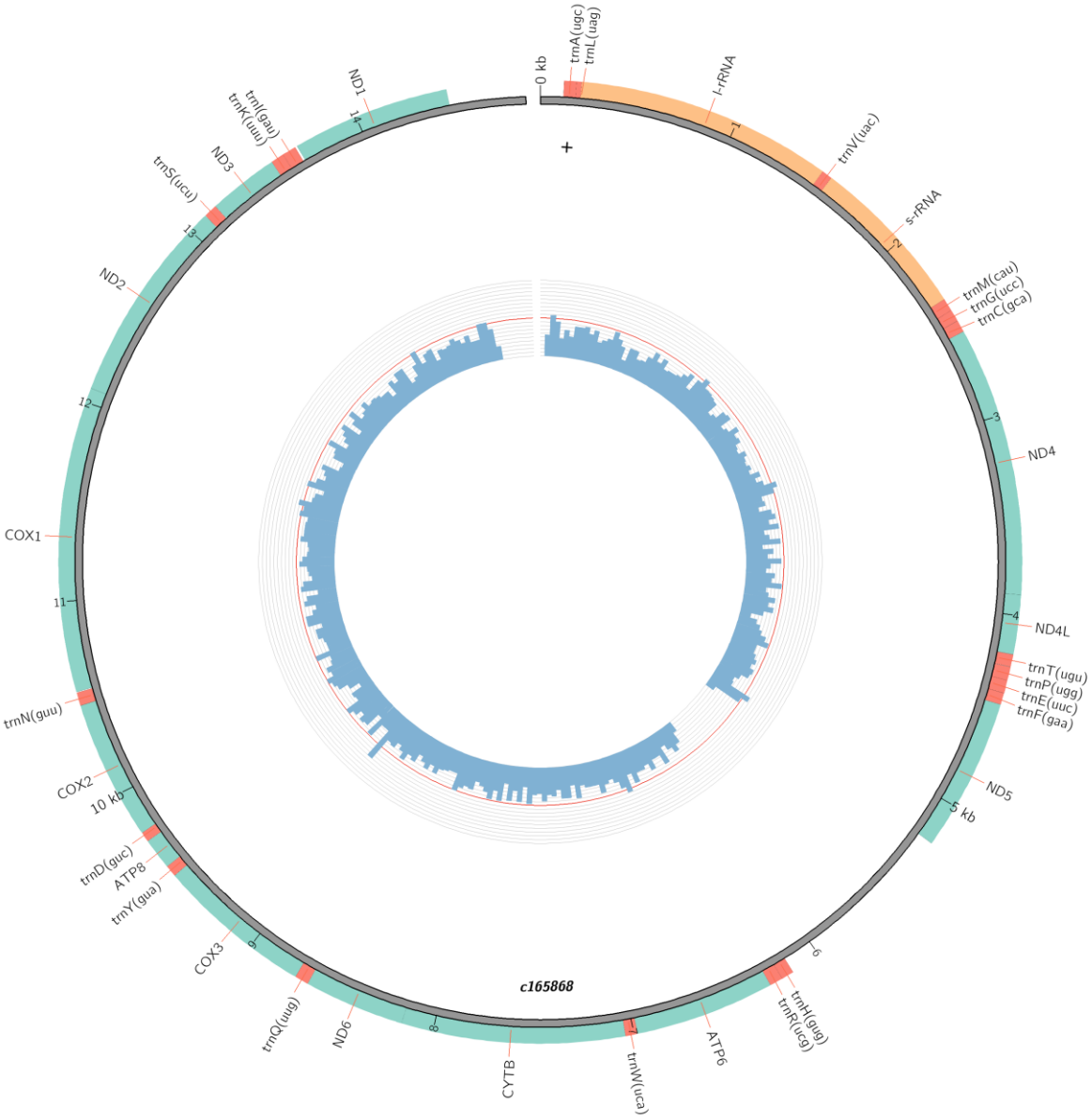
A.21 Nearly completed mitochondrial genome generated by MITObim for ID 24. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



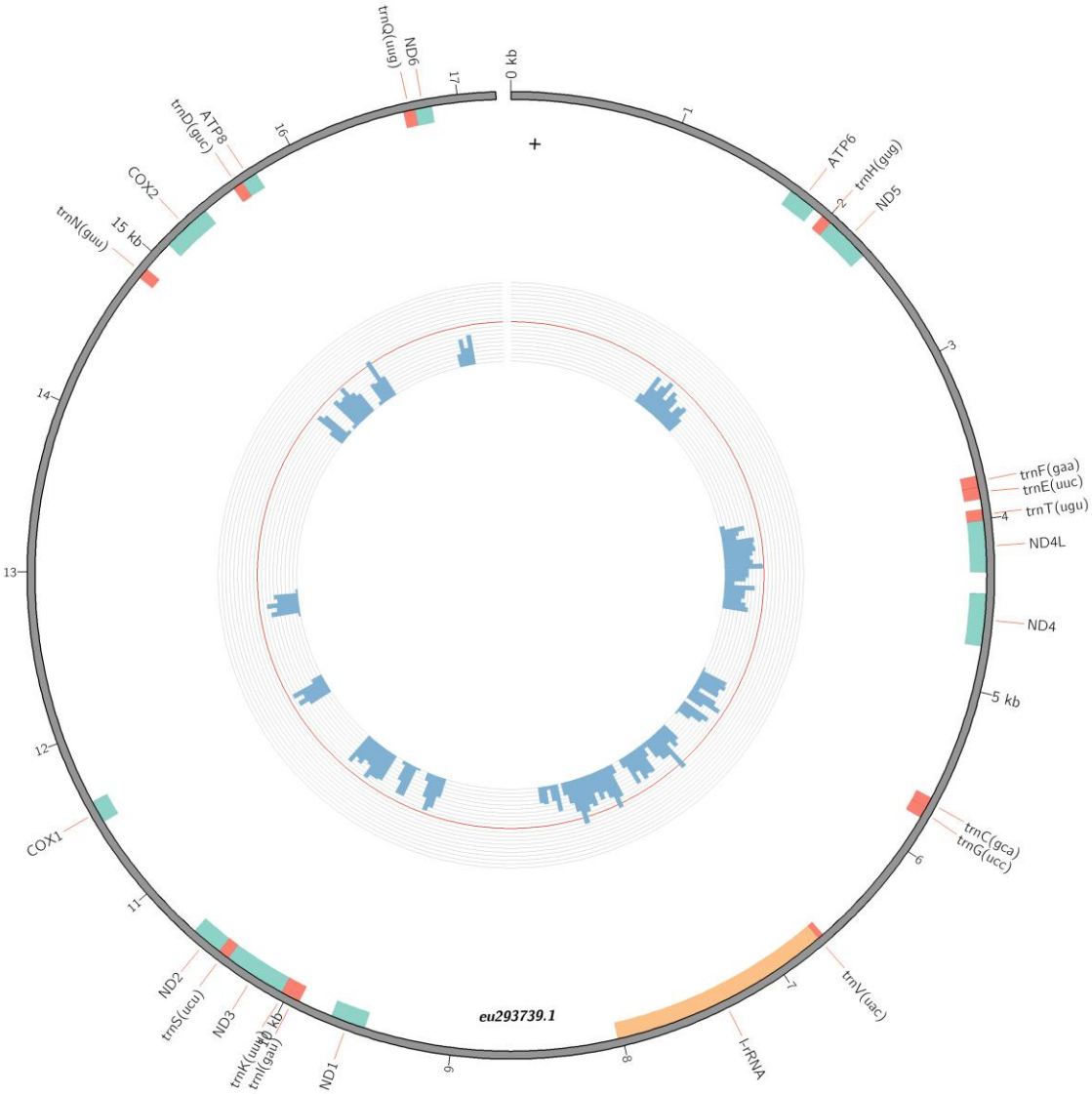
A.22 Nearly completed mitochondrial genome generated by MITObim for ID 26. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



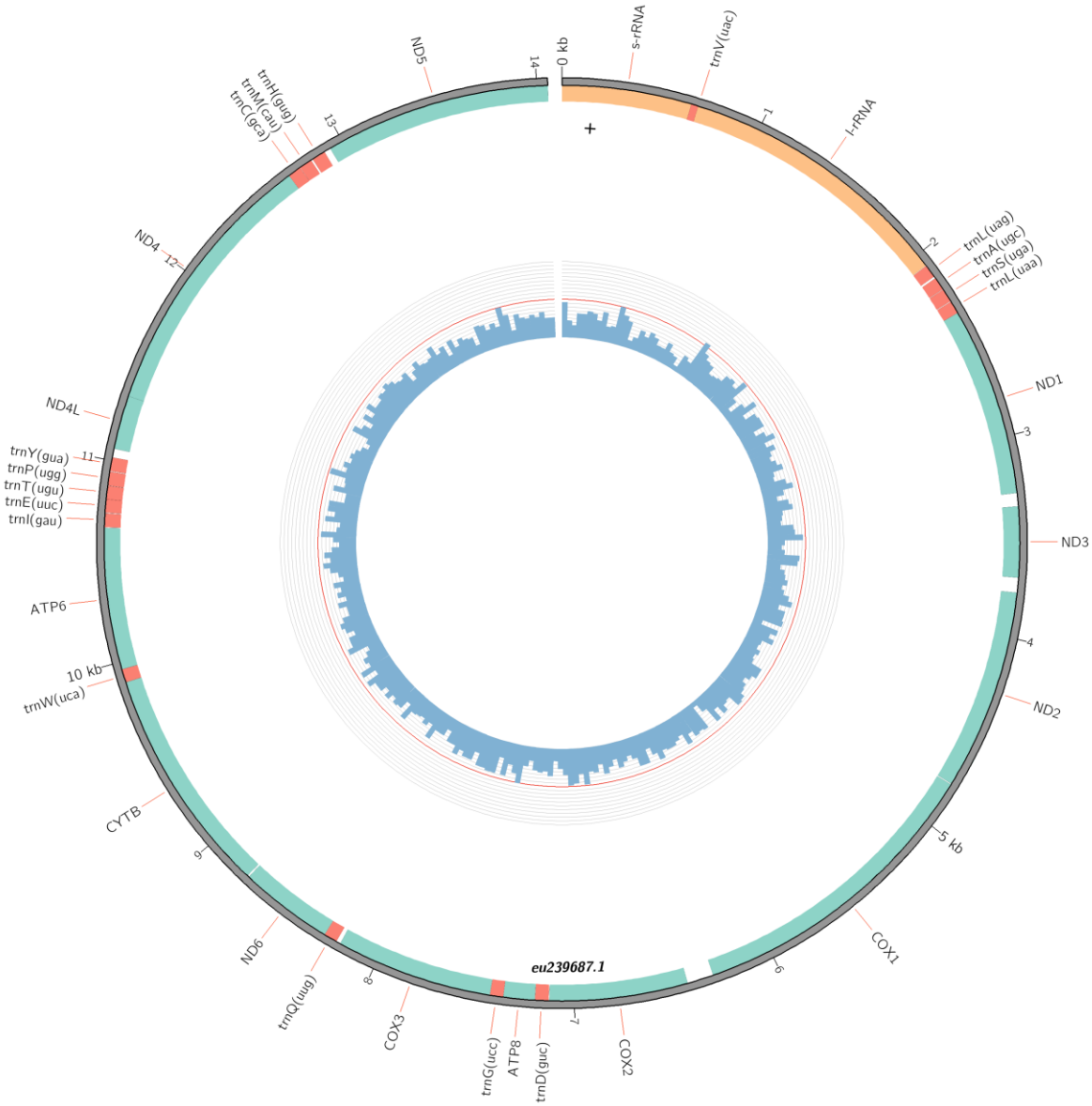
A.23 Nearly completed mitochondrial genome generated by MitoZ for ID 26. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution with green color as the outline, depth lower than the minimum value (default 20) showed in red, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



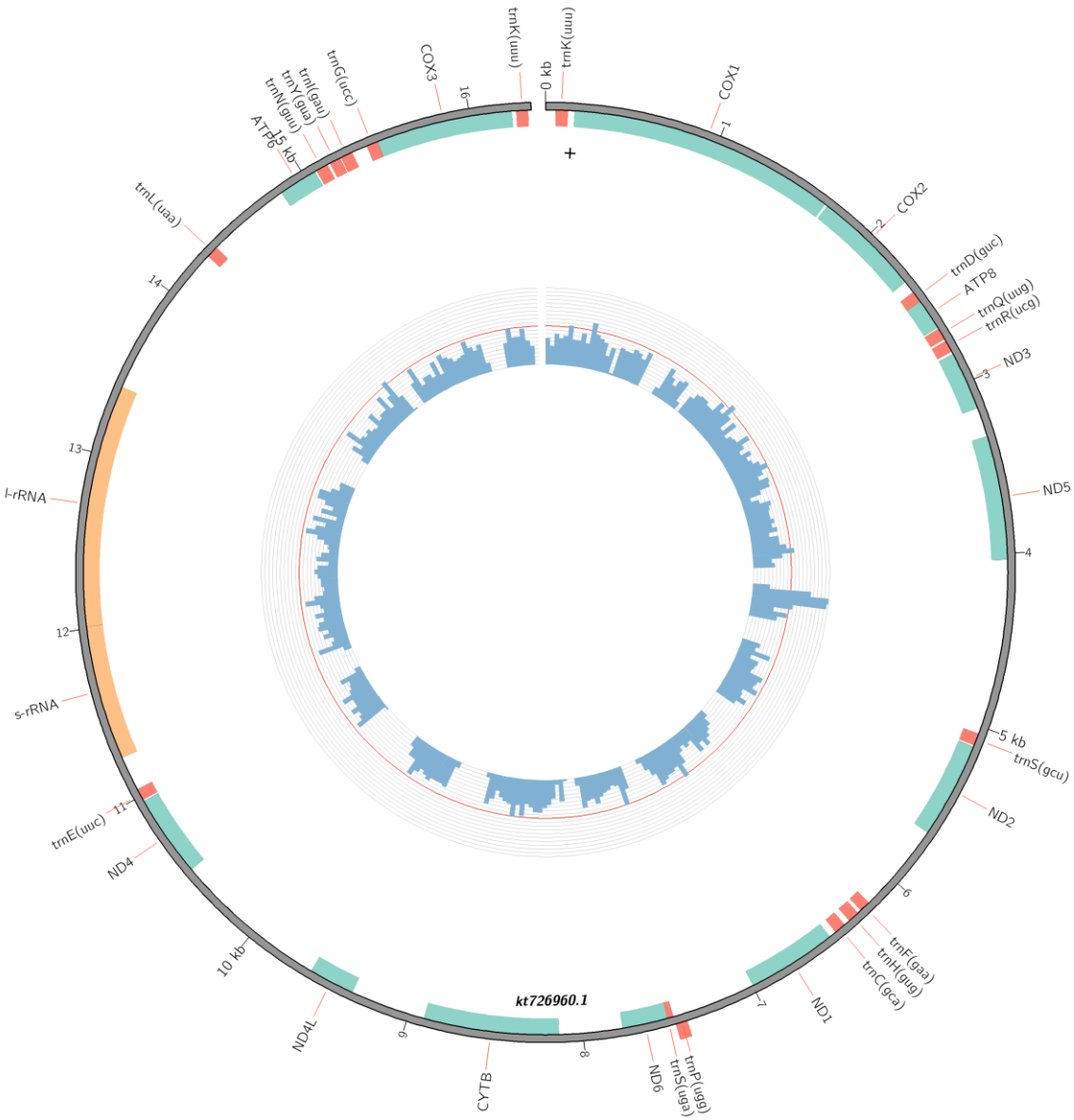
A.24 Nearly completed mitochondrial genome generated by MITObim for ID 34. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



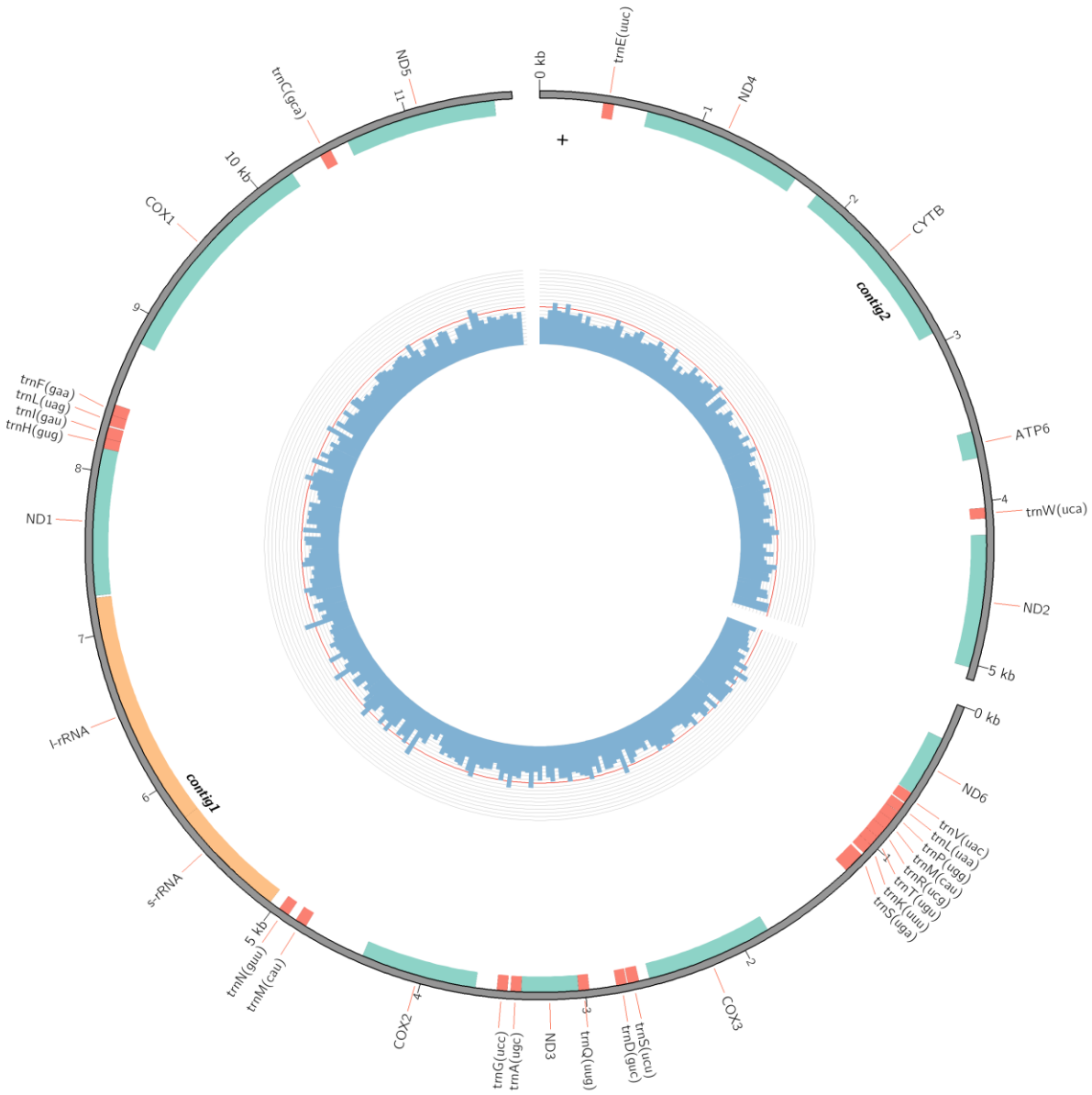
A.25 Mitochondrial genome generated by MITObim for ID 39. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



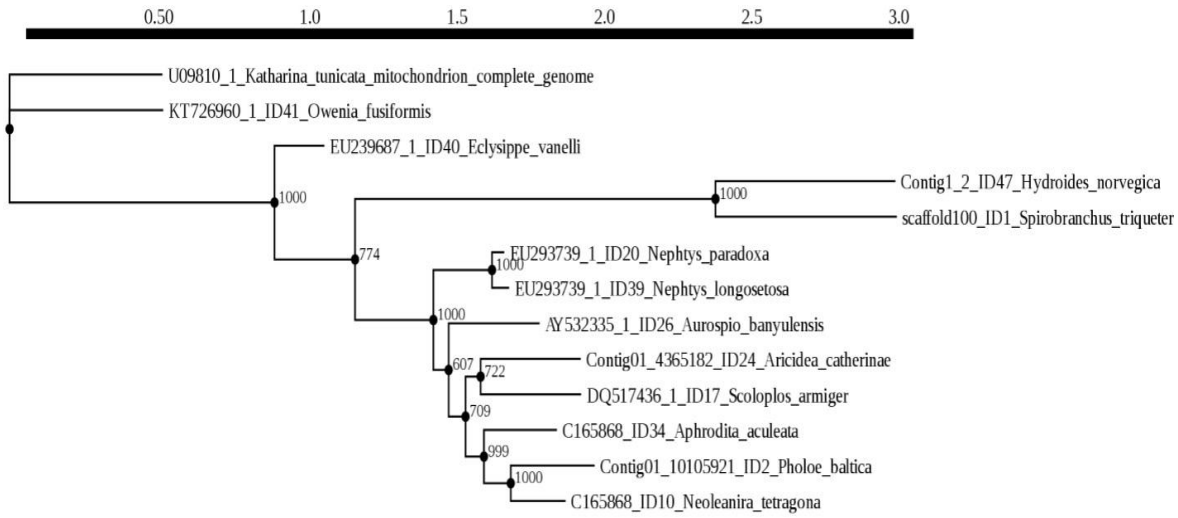
A.26 Nearly completed mitochondrial genome generated by MITObim for ID 40. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



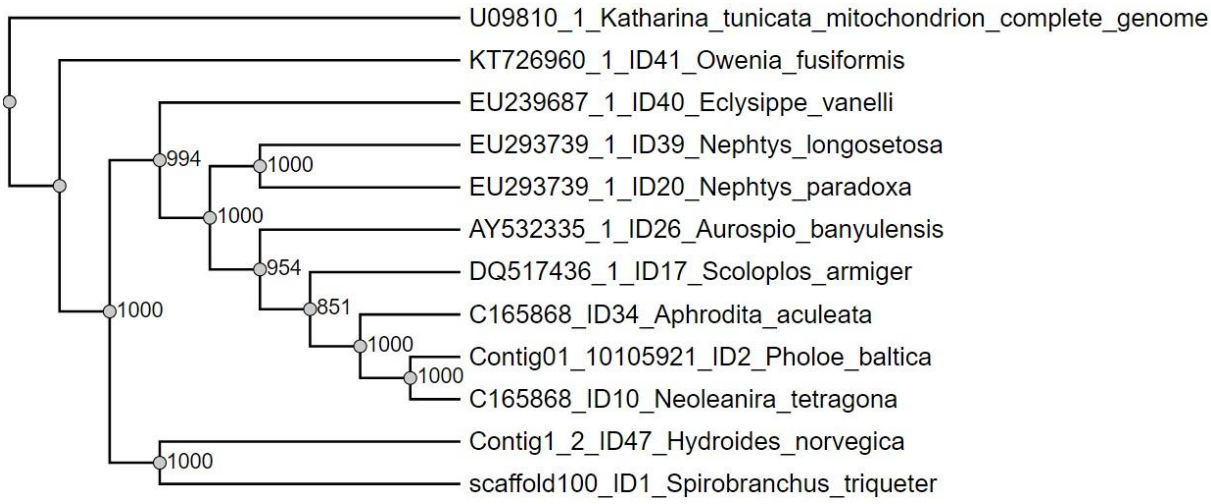
A.27 Nearly completed mitochondrial genome generated by MITObim for ID 41. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



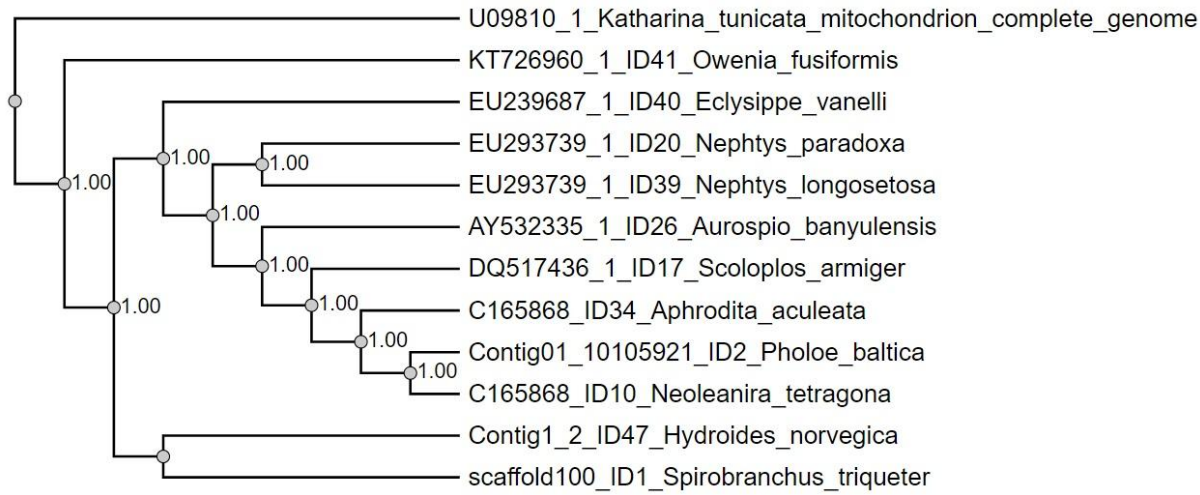
A.28 Nearly completed mitochondrial genome generated by NOVOPlasty for ID 47. This figure has 3 circles, the most inner one shows the GC content, the middle one reveals the depth distribution which is not shown, and the most outer one shows the gene features, rRNA showed in orange, tRNA showed in red and CDS showed in green.



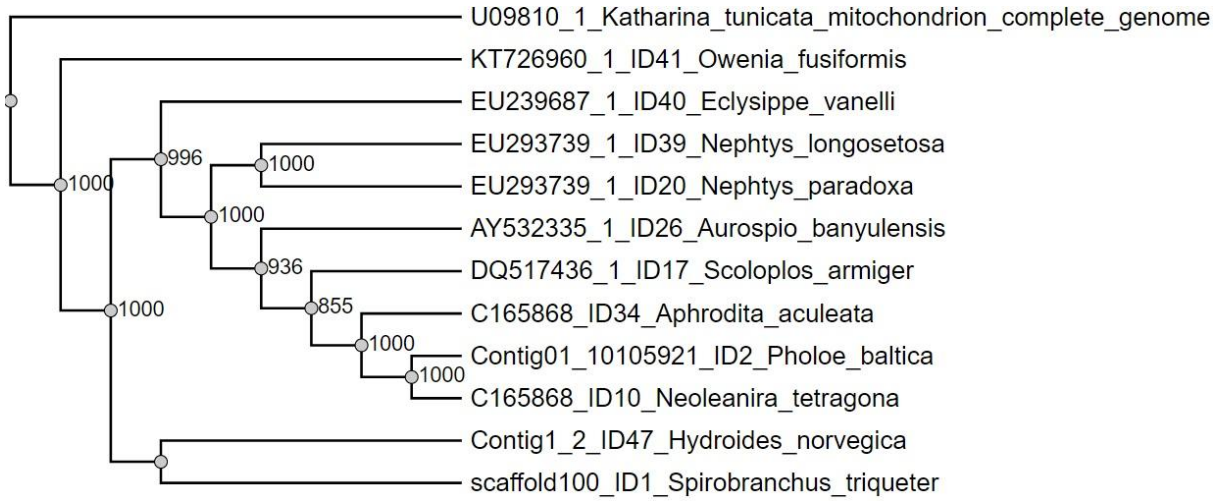
A.29 Phylogenetic tree inferred from 12 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML3.0 maximum-likelihood methods + BIC (Bayesian Information Criterion) Best model: GTR + G and Bootstrap 1000 replicates. The minimum Bootstrap value is 607.



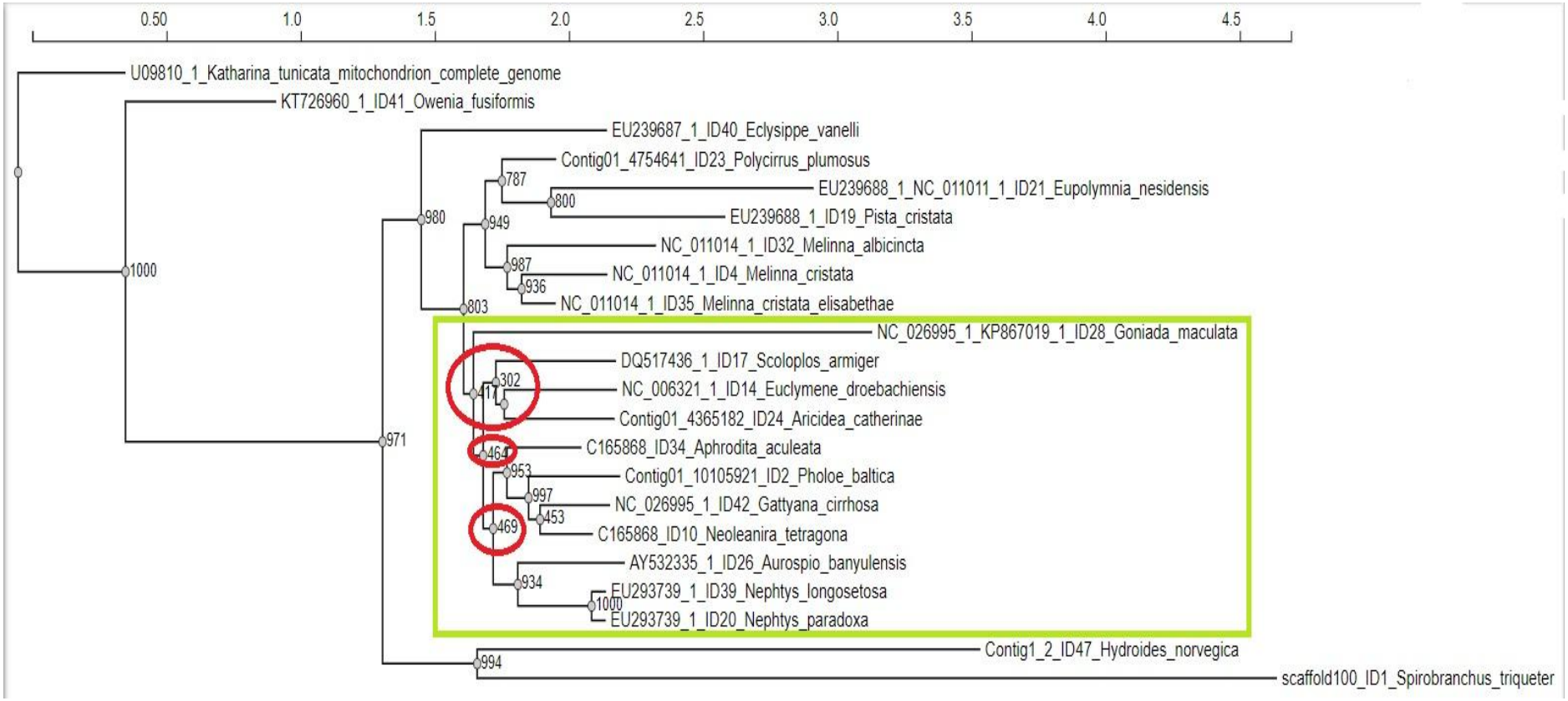
A.30 Phylogenetic tree inferred from 11 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML maximum-likelihood methods + AIC (Akaike Information Criterion) Best model: GTR + G and Bootstrap 1000 replicates. The minimum Bootstrap value is 851.



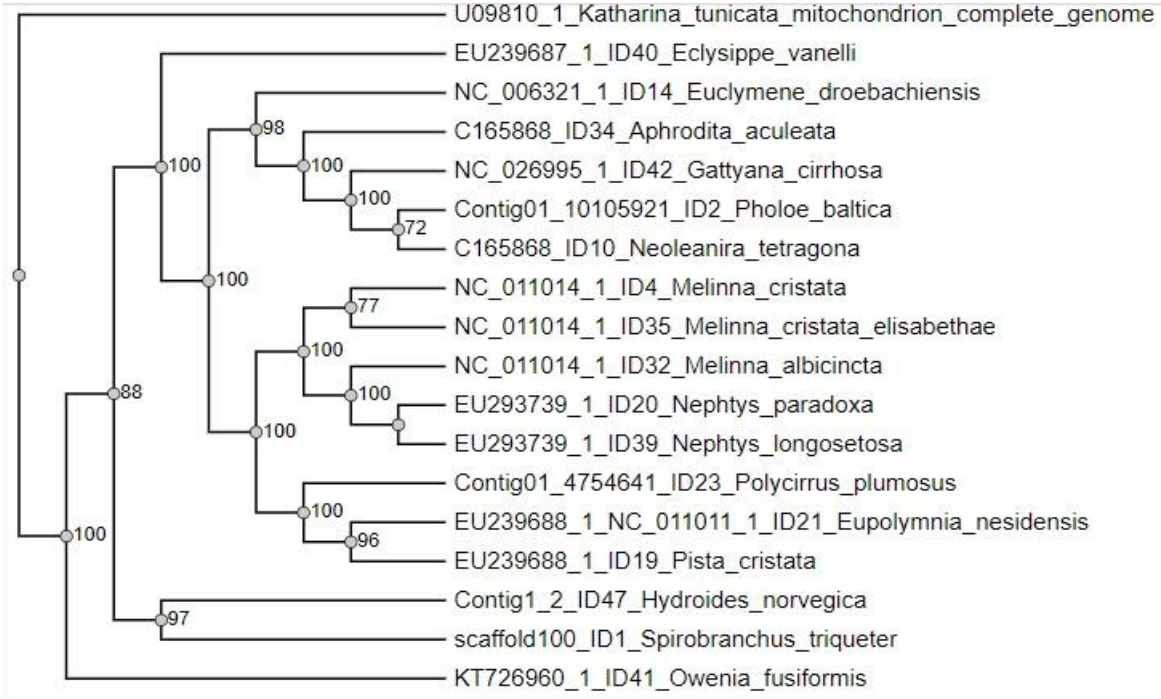
A.31 Phylogenetic tree inferred from 11 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML3.0 maximum-likelihood methods + AIC (Akaike Information Criterion) Best model: GTR + G and Fast likelihood-based method: aLRT SH-like. All the values are 1.0. (Forgot to choose Bootstrap 1000 for this one.)



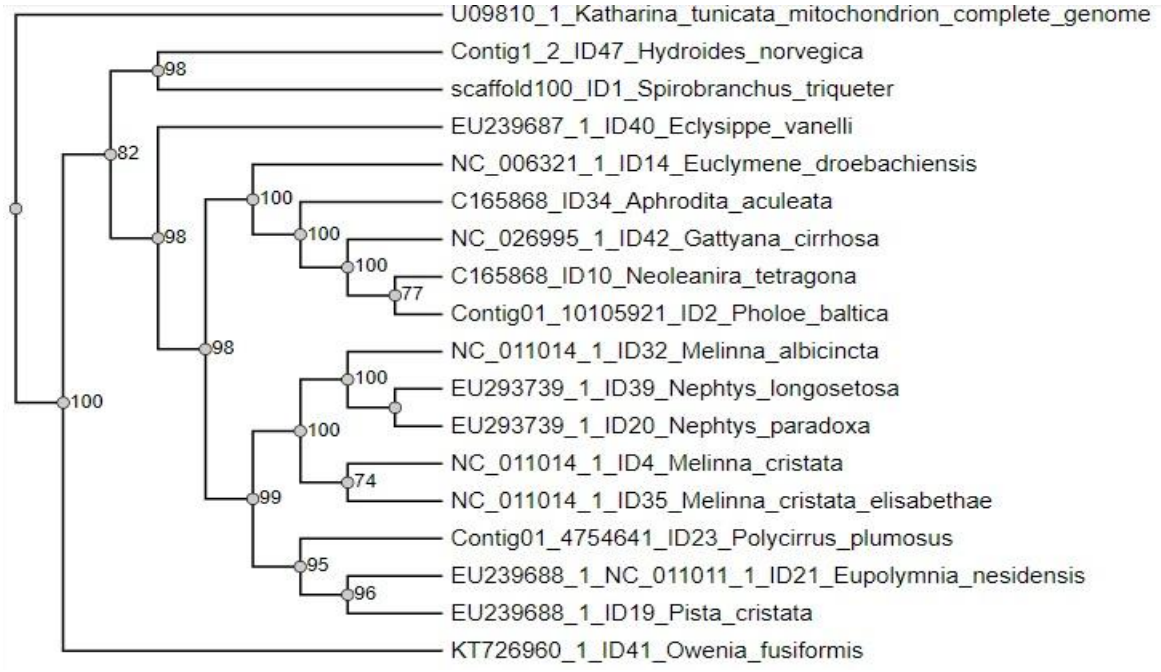
A.32 Phylogenetic tree inferred from 11 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML3.0 maximum-likelihood methods + BIC (Bayesian Information Criterion) Best model: GTR + G and Bootstrap 1000 replicates. The minimum Bootstrap value is 855.



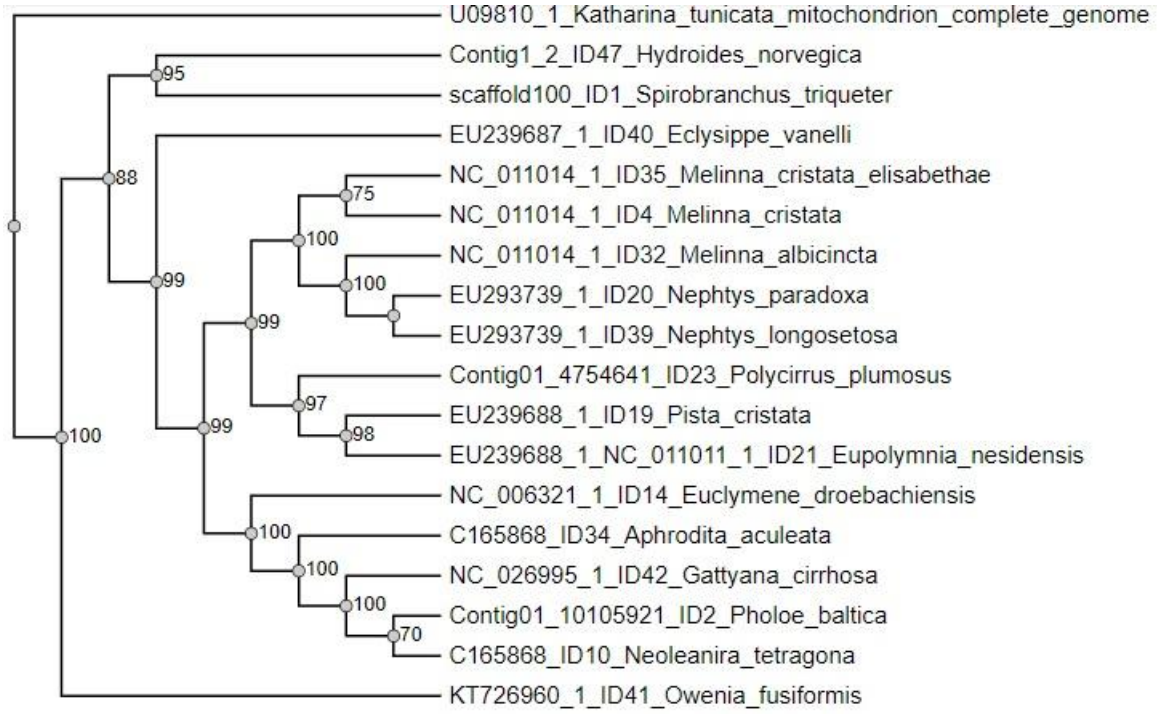
A.33 Phylogenetic tree inferred from 21 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML3.0 maximum-likelihood methods + BIC (Bayesian Information Criterion). Best model: GTR + G and Bootstrap 1000. The minimum Bootstrap value is 302. The unstable clade showed in green, while unstable nodes showed in red.



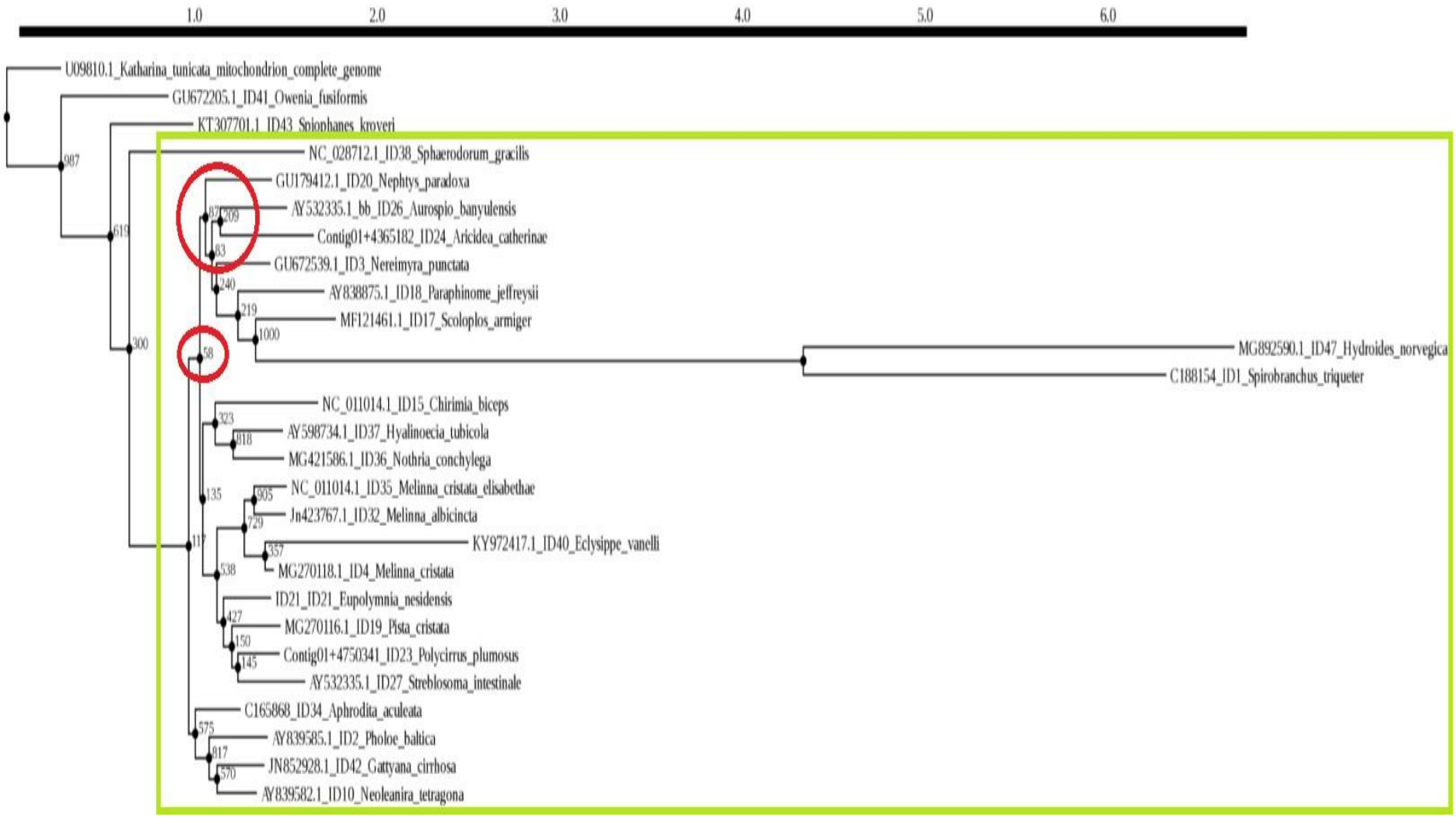
A.34 Phylogenetic tree inferred from 17 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML maximum-likelihood methods + AIC (Akaike Information Criterion) Best model: GTR + G and Bootstrap 100 replicates to reduce the computing time to 7 hour(s), 40 minute(s). The minimum Bootstrap value is 72.



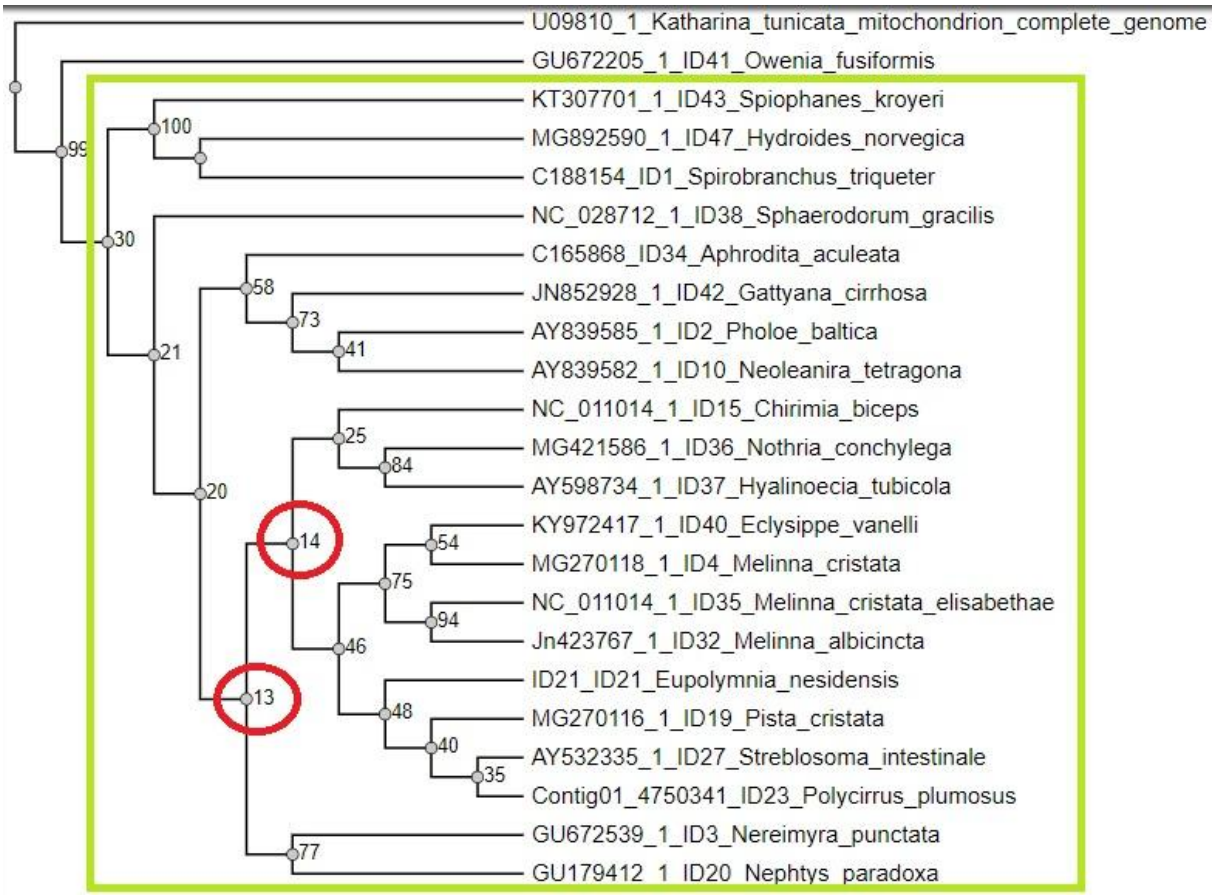
A.35 Phylogenetic tree inferred from 17 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML3.0 maximum-likelihood methods + AIC (Akaike Information Criterion) Best model: GTR + G and Bootstrap 100 replicates to reduce the computing time to 7 hour(s), 39 minute(s). The minimum Bootstrap value is 74.



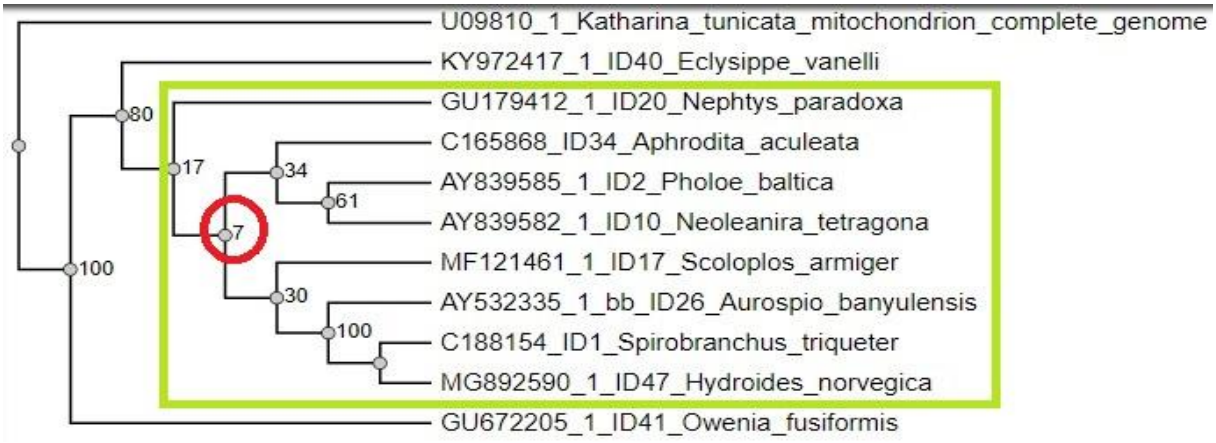
A.36 Phylogenetic tree inferred from 17 nearly completed mitochondrial genomes + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML maximum-likelihood methods + BIC (Bayesian Information Criterion) Best model: GTR + G and Bootstrap 100 replicates to reduce the computing time to 7 hour(s), 43 minute(s). The minimum Bootstrap value is 70.



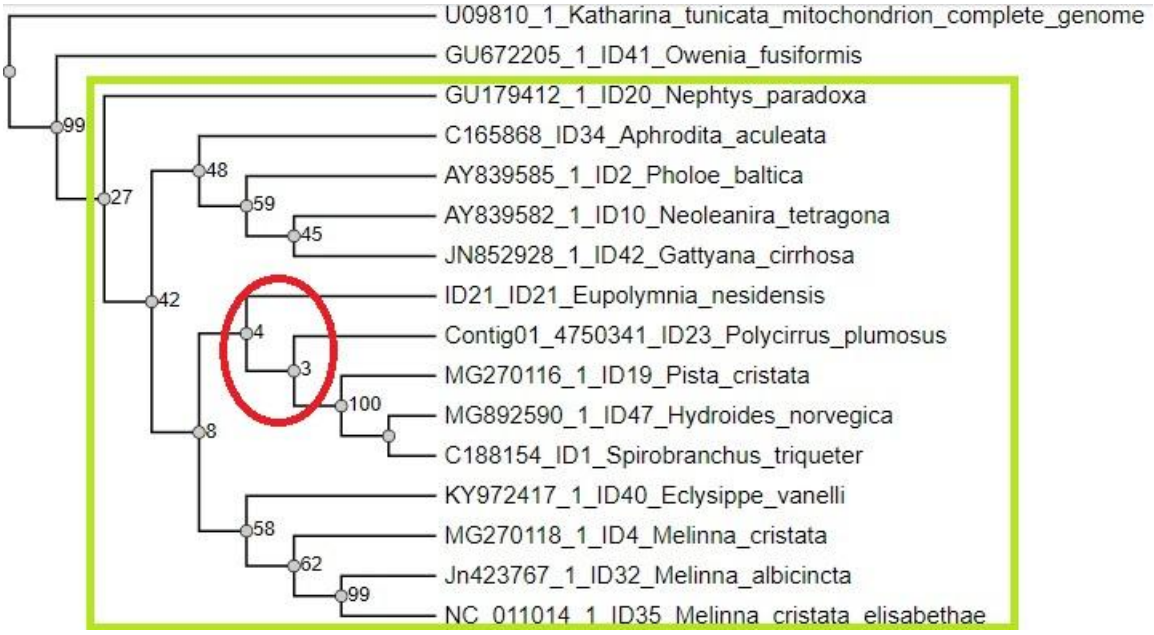
A.37 Phylogenetic tree inferred from 26 COI sequences + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML maximum-likelihood methods + BIC (Bayesian Information Criterion) Best model: GTR + G and Bootstrap 1000 replicates. The minimum Bootstrap value is 58. The unstable clade showed in green, while the most unstable nodes showed in red.



A.38 Phylogenetic tree inferred from 22 COI sequences + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML maximum-likelihood methods + BIC (Bayesian Information Criterion) Best model: GTR +G+I and Bootstrap 100 replicates. The minimum Bootstrap value is 13. The unstable clade showed in green, while the most unstable nodes showed in red.



A.39 Phylogenetic tree inferred from 10 COI sequences + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML maximum-likelihood methods + AIC (Akaike Information Criterion) Best model: GTR +G+I and Bootstrap 100 replicates. The minimum Bootstrap value is 7. The unstable clade showed in green, while the most unstable nodes showed in red.



A.40 Phylogenetic tree inferred from 15 COI sequences + outgroup: Katharina tunicata mitochondrion, complete genome (U09810.1), using PhyML3.0 maximum-likelihood methods + AIC (Akaike Information Criterion) Best model: GTR +G+I and Bootstrap 100 replicates. The minimum Bootstrap value is 3. The unstable clade showed in green, while the most unstable nodes showed in red.