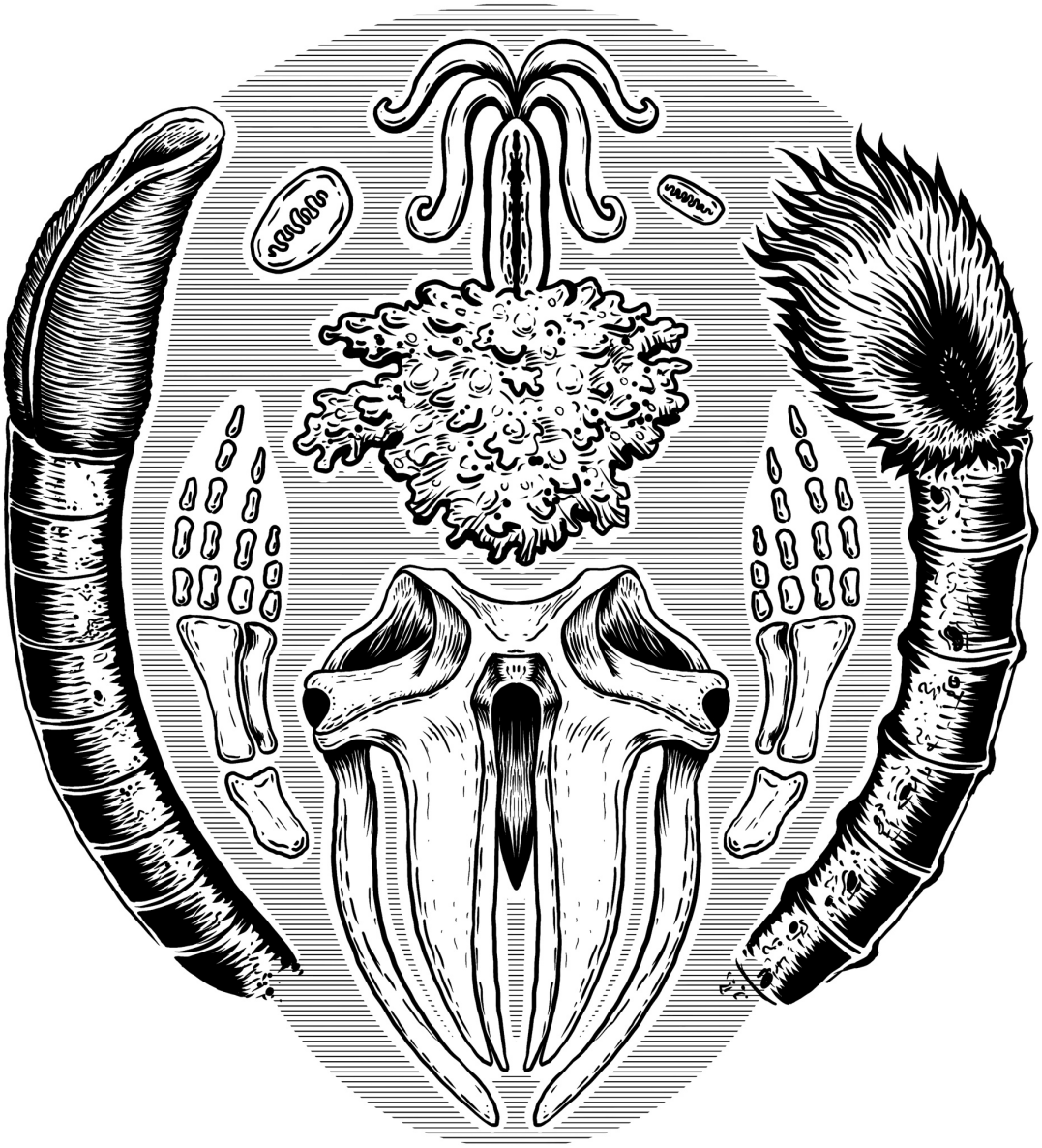
PhD THESIS

The genomic basis of bacterial symbiosis in *Osedax* and Vestimentifera (Siboglinidae, Annelida)

Giacomo Moggioli



Supervisors: Dr. José M. Martín-Durán, Dr. Lee Henry, Dr. Yannick Wurm

Submitted in partial fulfilment of the requirements of the Degree of Doctor of Philosophy

Queen Mary University of London

March 2023

Chapter 2 – Material and Methods

In my PhD project I assembled and annotated the genomes of three siboglinid species (*O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*) to investigate the genomic basis of bacterial symbiosis in this annelid group. As such, my research was largely computational, except for the fieldwork and lab-work required to obtain the samples used for genomic and transcriptomic analyses, which collaborators conducted. Therefore, this chapter collects all the techniques I developed and implemented to generate the results that will be reported in **Chapter 3**, **Chapter 4** and **Chapter 5**. Our project was a joint effort between different research groups in which I executed the bulk of the analyses on the host genome. Others contributed to more specific analytical aspects and credits of the contributor are found in the text and in **Table 2**.

Contributor	Contribution
Prof Greg Rouse	Specimen collection
Dr Nadezhda N. Rimskaya-Korsakova	Specimen collection
Martin Tran	gDNA extraction
Dr Chema Martin-Duran	Main supervision, RNA extraction
Balig Panossian	Symbiont genome assembly and annotation, symbiont comparative analyses
Francisco M. Martín- Zamora	Bayesian inference of phylogeny
Dr. Yanan Sun	Reconstruction of immune system
Dr. Daniel Thiel	Reconstruction of GPCRs
Dr Lee Henry	Supervision
Giacomo Moggioli	All the other analyses

Table 2 – Contributions to this PhD project: On the left are specified the contributors to this PhD project and on the right is specified the work they done.

2.1 - *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila* specimen collection

A specimen of *Osedax frankpressi* was collected by Prof. Greg Rouse off the Californian coast at "Whalefall 2893", located in the Monterey submarine canyon at 2893m deep, using the Remoted Operated Veichle (ROV) "Tiburon" from the ship "R/V Western Flyer" in November 2004. In addition, individuals of *Oasisia alvinae* and *Riftia pachyptila* were acquired by Prof. Greg Rouse from the Matterhorn vent site, located in the Gulf of California at 3653m deep, using the ROV "SuBastian" from the ship "R/V Falkor" in November 2018. Mexican samples were collected under CONAPESCA permit PPFE/DGOPA-200/18.

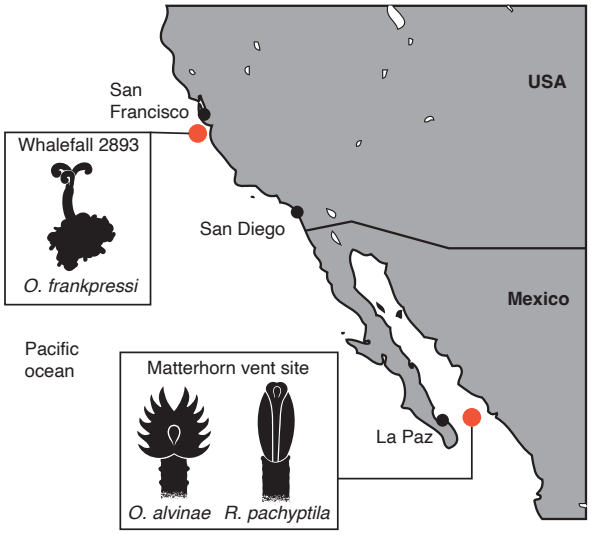


Figure 3 – Specimen collection sites: Schematic map of southern California and northwest Mexico indicates the main cities (black dots) and collection sites (red dots).

2.2 - *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila* gDNA and RNA extraction

An entire adult female *O. frankpressi*, a piece of trunk (including trophosome) of *Oasisia alvinae*, and a piece of vestimentum of *R. pachyptila* were used by Martin Tran to

extract ultra-high molecular weight genomic DNA (gDNA), following the Bionano Genomics IrysPrep agar-based, animal tissue protocol (Catalogue # 80002). In addition, samples from the same specimens of *Oasisia alvinae* and *R. pachyptila* or a separate individual of *O. frankpressi* were used to extract total RNA from different tissues. We obtained total-RNA from the crown, the opisthosome and throphosome with an additional set of replicates for *Oasisia alvinae* and from the crown and the trunk wall with no replicates for *R. pachyptila*, using a NEB totalRNA Monarch kit. *O. frankpressi*'s total-RNA from its body and its roots, with no replicates, was extracted by Prof. Greg Rouse's group [85] and sent to our laboratory.

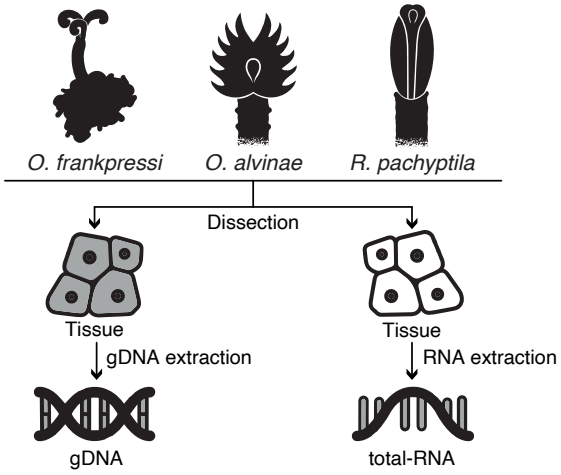


Figure 4 – gDNA and RNA extraction: Visual representation of our gDNA and RNA extraction workflow.

2.3 - *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila* gDNA and RNA sequencing

The gDNA was used for long read PacBio sequencing and short read Illumina sequencing at the Genome Centre of the University California Berkeley in a PacBio Sequel II and Illumina NovaSeq6000 platforms, respectively. In addition, total-RNA was used for standard strand-specific RNA Illumina library preparation, and the libraries were sequenced to a depth of 40-50 million paired reads of 150 bases length. During the RNA library preparation, eukaryotic mRNA was isolated from bacterial RNA and other forms of eukaryotic RNA

employing the PolyA step, which is based on oligo dT primers which bind the poly-adenylated tail found only on eukaryotic mRNA.

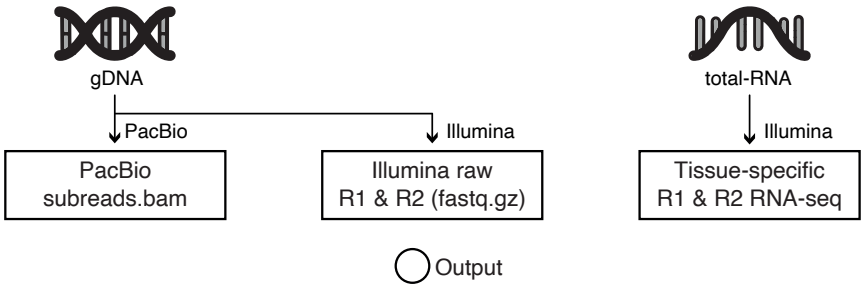


Figure 5 – gDNA and RNA sequencing: Visual representation of gDNA and RNA sequencing from which we obtained Illumina short-reads, PacBio long-reads and RNA-seq data.

2.4 - High Performance Computing cluster

Most of our analyses were run on the High Performance Computing (HPC) cluster of Queen Mary University of London (QMUL), commonly referred to as "Apocrita" [86]. Apocrita consists of around 350 nodes and 12,500 cores and is located in the Jisc-shared data centre in Slough, under management of the ITS Research team at QMUL. Apocrita allowed us to run our analyses in parallel in different cores, eventually reducing the time taken to compute. The genome assembly pipeline required the most resources compared with our other analyses, reaching peaks of 480G of RAM consumption. In most of the cases, we used Anaconda3 [87] to create virtual environments inside Apocrita in order to be able to freely install software and dependencies, and run the analyses needed for this project.

2.5 - Genome assembly of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*

The genome assembly pipeline has the purpose of constructing the haploid versions of the genomes of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila* starting from the raw data that we obtained from sequencing the ultra-high molecular weight gDNA with both PacBio

and Illumina technologies. This pipeline includes four main phases (**Fig. 6**): an initial assembly using PacBio data, a polishing step which combine both PacBio and Illumina data in order to improve the quality of the initial assembly, a decontamination step in which prokaryotic sequences are identified and removed from the assembly, and a final step in which the haploid version of the host is obtained.

After each of the 4 main steps of the pipeline, I performed quality controls using both BUSCO v.3.0.2 [88] and QUAST v.5.0.2 [89]. I used BUSCO to compare the three assemblies with the Metazoa Odb10 database, which contains 954 universally distributed single-copy genes which are relatively well preserved in all the lineages of the metazoan tree. Based on the assumption that a metazoan organism should display a conserved set of single copies BUSCO genes, the higher the BUSCO's completeness score is, the more accurate a genome assembly can be considered. On the other hand, QUAST was used to evaluate general statistics of the three assemblies such as genome size, the N50 contiguity value, GC content and number of contigs.

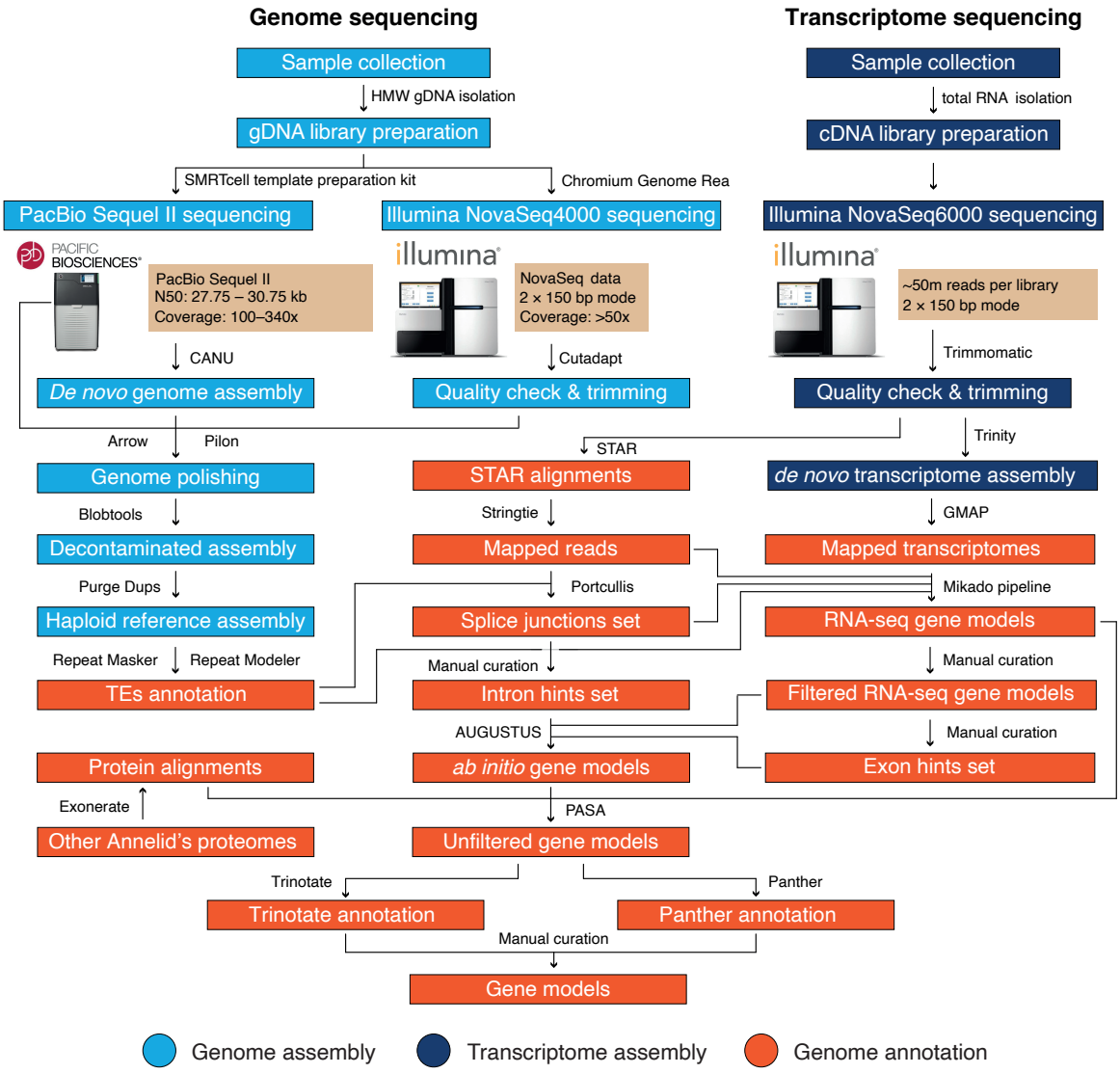


Figure 6 – Genome assembly and annotation pipeline overview: Visual representation of the main steps of the pipeline I used to obtain the annotated genomes of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.5.1 - Initial Assembly

I used CANU v.1.8 [90] for the initial assembly of the genomes of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*. CANU is an assembler designed for high-noise long reads such as those produced by PacBio and Oxford Nanopore technologies. CANU introduces an advanced strategy for handling repeats, which statistically reduces the chance a repetitive *k*-mer will be selected for overlapping reads. It runs in four different steps: 1 – Firstly, it detects

overlaps in the long reads using a variant of the greedy "best overlap graph" algorithm [91]. 2 – The correction step finds the indels and substitutions, the two main errors in PacBio data, and outputs corrected reads. I used the PacBio corrected reads generated in this way in other analyses as well, such as in *k*-mer analyses. 3 – Trimming is needed to identify unsupported bases, hairpin adapters, chimeric sequences, and other anomalies in the reads. 4 – Finally, CANU produces the initial assembly overlapping the trimmed-corrected reads.

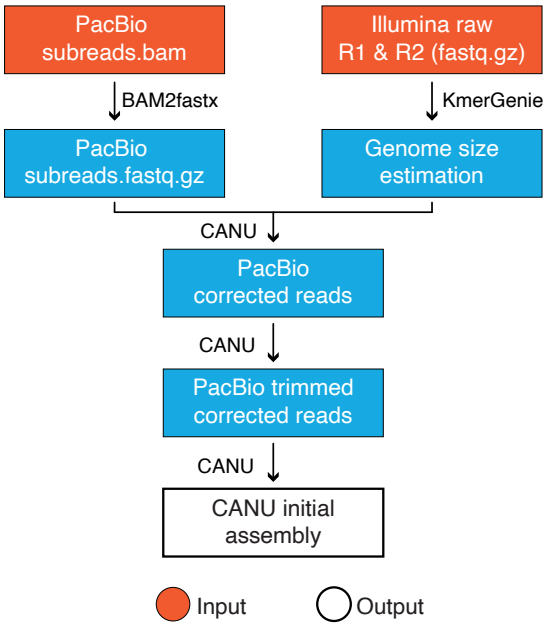


Figure 7 – Initial assembly: Visual representation of the initial CANU assembly. Boxes represent data files while arrows represent the software implemented in the pipeline.

In order to run CANU, I needed an initial estimation of the genome size of the three species, which I obtained using KmerGenie v.1.7016 [92] and the Illumina raw reads. Furthermore, I extracted raw PacBio sequences in fasta format ("subreads.fastq.gz") from the raw PacBio "subreads.bam" files using BAM2fastx tools v.1.3.0 [93] in order to use them as input for CANU.

I launched CANU using the following command:

```
CANU \
-p species_name \
-d /path/to/output \
genomeSize=Xg \
maxMemory=480g \
maxThreads=48 \
useGrid=false \
gridEngineResourceOption="-l h_vmem=MEMORY -pe smp THREADS" \
batOptions="-dg 3 -db 3 -dr 1 -ca 500 -cp 50" \
-pacbio-raw /path/to/input/PacBio_raw.fastq.gz
```

- `genomeSize=Xg`, where X is the estimation obtained with KmerGenie (500 Mb for *O. frankpressi*, 1 Gb for *Oasisia alvinae* and 750 Mb for *R. pachyptila*).
- `gridEngineResourceOption`: These options specify the grid settings needed to request memory and threads for each job, CANU will update "THREADS" and "MEMORY" accordingly to the requirements for each job.
- `batOptions`: the first three parameters set an error rate threshold for: `-dg` assembling contigs, `-db` detecting bubbles and `-dr` detecting repeats. The two other parameters regulate the fragmentation of contigs in case of an overlapping repeat. `-ca 500` tells CANU to split the contig in case a repeat is in an overlap area between two reads in which one overlapping sequence is less than 500bp shorter than the other overlap. `-cp` is similar to `-ca` but compares the percentages of the read lengths, `-cp 50` would break the contig unless one overlap is 50% bigger than the other.

2.5.2 - Polishing

The polishing step is needed to improve further the base-quality of the assemblies produced with an initial assembler, CANU in my pipeline, by correcting possible errors that

occurred during the sequencing of long reads. I conducted two rounds of polishing with Arrow (pbgcpp v.1.9.0) [94] using the PacBio reads, followed by two additional rounds of polishing with Pilon v.1.23 [95] using Illumina reads.

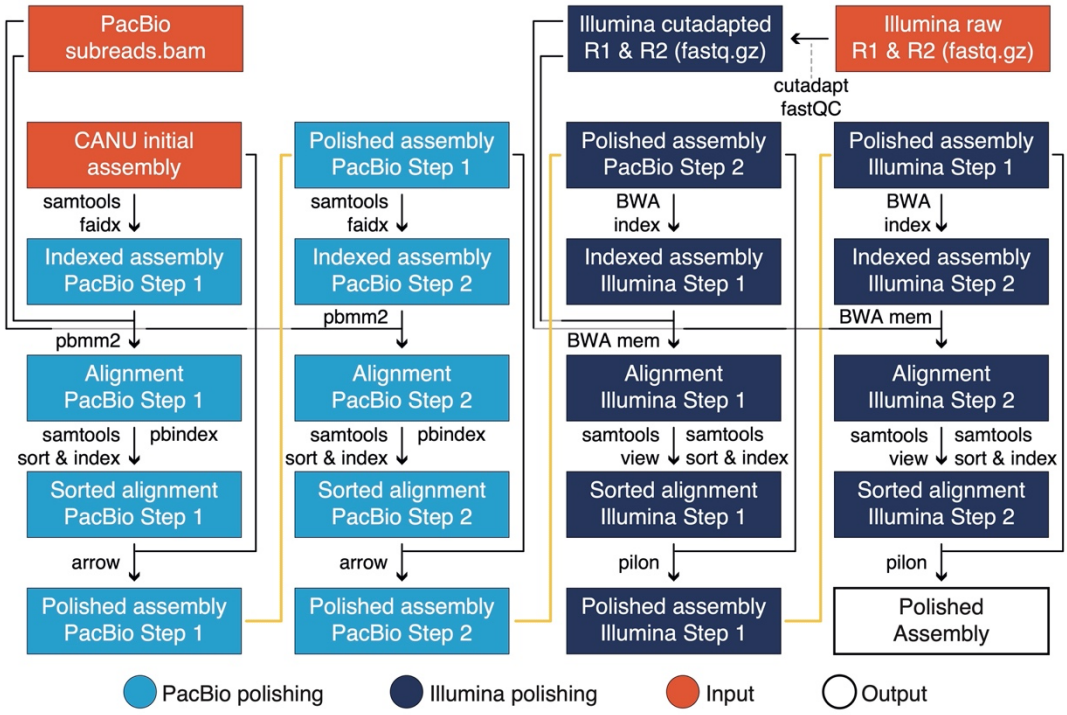


Figure 8 – Polishing: Visual representation of the steps involved in the polishing process. The yellow lines connect boxes that represent the same data file. Boxes represent data files, while arrows represent the software implemented in the pipeline.

During the polishing step I used Pbmm2 v.1.1.0 [96] to align the PacBio reads to the three initial assemblies, Arrow to polish the alignments with PacBio long reads, Cutadapt v.2.5 [97] to remove the adapters from Illumina raw reads, FastQC v.0.11.8 [98] to evaluate the quality of the Illumina reads, Bwa v.0.7.17 [99] to align the adapter-trimmed Illumina reads to the 3 PacBio-polished assemblies, Pilon to improve the quality of the genomes using the higher accuracy of short Illumina reads and, finally, I used SAMtools v.1.3.1 [100] in various steps of the polishing pipeline to index, sort and re-format SAM files into BAM files.

2.5.3 - Decontamination

An important step in *de novo* genome assemblies of a eukaryotic organism is the identification and filtering of contigs with a procaryotic origin. The main software I used to accomplish this task was BlobTools v.2.1 [101].

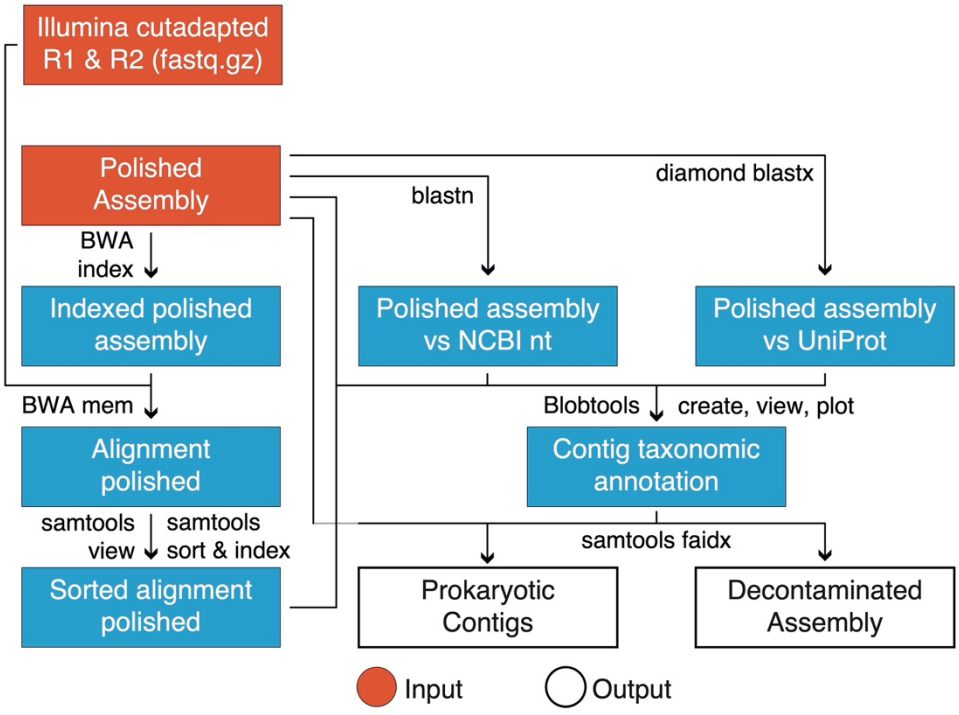


Figure 9 – Decontamination: Visual representation of the decontamination strategy. Boxes represent data files, while arrows represent the software implemented in the pipeline.

The polished versions of the genomes of *Oasisia alvinae, O. frankpressi* and *R. pachyptila* and the respective quality checked and adaptor-trimmed Illumina R1 and R2 fastq files were used to obtain sorted alignments with a combination of BWA and SAMtools. At the same time, I used BlastN v.2.9 [102] to compare the polished assemblies with the NCBI nucleotide database (nt_v.5) [103], and Diamond BlastX v.0.8.22 [104] to align the polished assemblies with the UniProt reference proteomes database [105]. I then used the outputs of both BlastN and BlastX, the polished assembly and the sorted alignment as inputs for BlobTools. This software is able to provide a taxonomic annotation for each contig in the polished assemblies taking into account the GC content of the sequences, the coverage of the

Illumina reads, and the sequence similarity matches obtained with Blast. Finally, SAMtools faidx was used to extract the eukaryotic contigs and obtain in this way the decontaminated assemblies. I decided to opt for a more conservative approach, so I also included contigs that were not matching any specific taxonomic group in the final decontaminated assemblies.

2.5.4 – Purging Haplotypes

Obtaining the haploid version of the three assemblies is the final step of the genome assembly pipeline. To do so, I used Purge_Dups v.1.0.1 [106], which evaluates sequence similarity and read coverage (from Pairwise mApping Format (paf) alignment files generated with Minimap2 v.2.17 [107]) to collapse distinct alleles, at regions of heterozygosity, into a single allele.

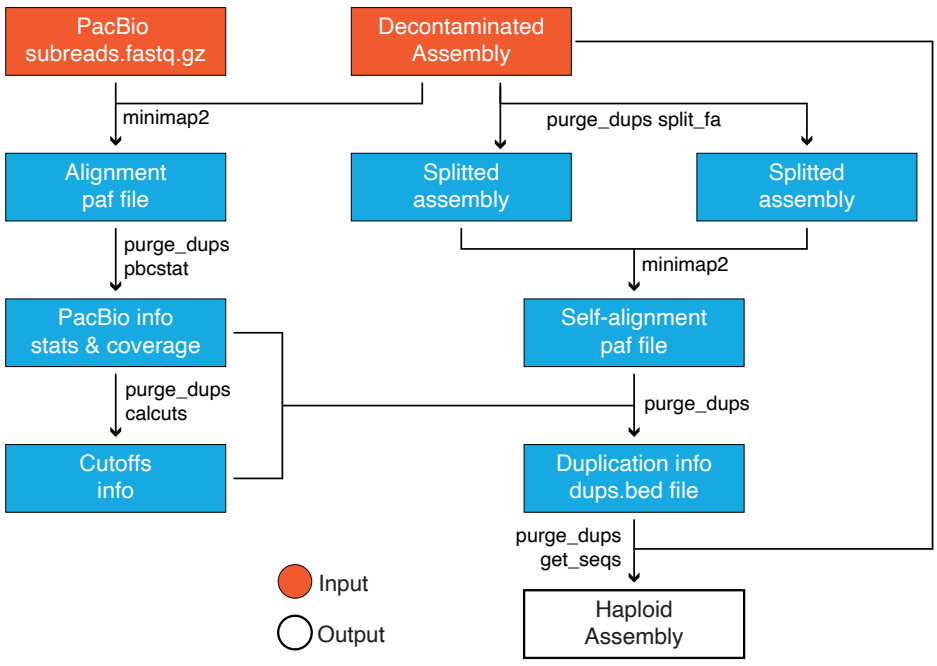


Figure 10 – Purging haplotypes: Visual representation of the de-haploidisation strategy. Boxes represent data files while arrows represent the software implemented in the pipeline.

The haploid assemblies are the final products of the genome assembly pipeline and the main inputs for the genome annotation pipeline.

2.6 - *k*-mer analyses

To estimate the accuracy of haplotype removal, potential contaminations in the Illumina data and, finally, to estimate the genome sizes, I run *k*-mer analyses based on 21-mers in the focal species. The graphical representation of the distribution of the *k*-mers extracted during analyses of this kind is called a *k*-mer spectra. *k*-mer spectra reveal useful information about the quality of data, such as sequencing biases or potential contamination, and also provide an evaluation of the genomic complexity, such as the genome size or the level of heterozygosity [108].

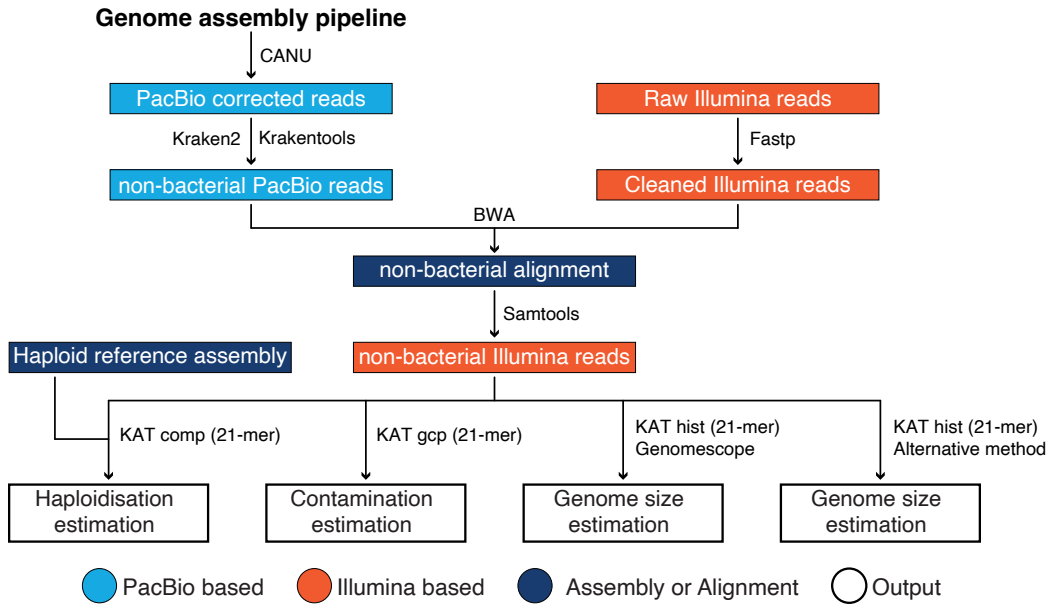


Figure 11 – *k*-mer analyses: Visual representation of the steps involved in the *k*-mer analyses which provided estimations of the genome sizes, haploidisation levels of the final assemblies and contamination levels of the raw Illumina short-reads. Boxes represent data files while arrows represent the software implemented in the pipeline.

I selected *k*-mers of 21 bases, or 21-mers, to have fragments large enough to assume they would tend to map uniquely to the genome but not excessively large to compromise the performance of the software used for these analyses and obtains sufficient resolution in the k-mer distributions. I based the 21-mers analyses on Illumina data instead of PacBio data for two

main reasons: First, Illumina reads are less error-prone than PacBio ones (error rates: ~0.1% for Illumina and 11–15% for PacBio [109]), and second, the advantages of having a long PacBio sequence (~30 kb and as long as 100 kb [109]) are irrelevant when the sequences are divided in units of 21 bases. Furthermore, to generate more accurate information about the genomes of *O. frankpressi* and *Oasisia alvinae*, I selected only the reads with a non-bacterial taxonomic annotation for the *k*-mer analyses. This procedure was not needed for *R. pachyptila* because the sample did not contain bacterial genomic material, as supported by the analyses with Blobtools (**Fig. 27**, **28**, **3.2.1.3 - Decontamination**) and the *k*-mer contamination estimation for *R. pachyptila* Illumina reads (**Fig. 33**, **3.2.2.1 - Contamination estimation**).

As the first step in my *k*-mer analyses, I extracted non-bacterial reads from the PacBio corrected reads for *Oasisia alvinae* and *O. frankpressi* using a combination of Kraken2 v.2.1.0 [110], which allowed the identification of bacterial genomic material, and Krakentools v.0.1 [110], the effective extraction software. I removed the adapters from the Illumina raw reads with Fastp v.0.20.1 [111], then I obtained non-bacterial Illumina reads by mapping the Illumina cleaned reads against the respective non-bacterial PacBio reads using BWA v.0.7.17 [99] and SAMtools v.1.3.1 [100].

I used KAT v.2.4.2 [108] as the main tool to perform *k-mer*-based quality controls of the genomes of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*. The Kat function `comp` allowed me to compare 21-mers generated from the non-bacterial Illumina reads, or from the cleaned Illumina reads for *R. pachyptila*, and the respective haploid assembly to assess the accuracy of haplotype removal in the final assemblies. Kat `gcp` function takes in account the GC count and the *k*-mer coverage level of the 21-mers obtained from the Illumina data and provides an estimation of the presence of other organisms, with a different GC profile, in the

input dataset. Finally, I used KAT hist function to generate histograms of the 21-mers which are used as input for the genome size estimation. I compared the size of the final haploid assemblies with the estimations obtained using GenomeScope2 [112] and an alternative approach assuming a Poisson distribution of the *k*-mer frequencies [113]. GenomeScope2 is a widely used online tool which infers a range of genome characteristics, such as genome size, from a given *k*-mer spectra by applying combinatorial theory and assuming a negative binomial model of the distribution of the *k*-mer frequencies. Together, these two approaches offer two estimations which I compared to the genome size of the final haploid assemblies for the evaluation of my results.

2.7 – Sequencing and genome assembly of *Siboglinum fiordicum*

A fourth siboglinid species belonging to Frenulata (**Fig. 1**, **1.1.1 – From family to phylum and back to family**), *Siboglinum fiordicum*, was originally intended to be included in this project. However, the small size of the worms made it impossible to extract sufficient genomic DNA to perform PacBio sequencing, and thus we could only generate Illumina short reads. The resulting assembly, as I explain below, was highly fragmented and we decided to not include this annelid in my analyses.

2.7.1 - Specimen collection, gDNA extraction and gDNA sequencing

A specimen of *S. fiordicum* was collected in Bergen (Norway) by Dr Nadezhda N. Rimskaya-Korsakova. Following the Bionano Genomics IrysPrep agar-based, animal tissue protocol (Catalogue # 80002), ultra-high molecular weight genomic DNA (gDNA) was extracted from an entire adult specimen and used to prepare a genomic library for short-read sequencing in an Illumina NovaSeq platform.

2.7.2 - Genome assembly

I could not use the same genome assembly pipeline that for *Oasisia alvinae*, *O. frankpressi* and *R. pachyptila* because I did not have PacBio long reads for *S. fiordicum.* Therefore, after estimating the genome size with KmerGenie v.1.7016 [92], I tested two different assemblers based on short Illumina reads, Platanus v.1.2.4 [114] and ABySS v.2.2.4 [115], and compared the differences of the genome statistics with the assemblies obtained with PacBio data for the other species, using both BUSCO v.3.0.2 [88] and QUAST v.5.0.2 [89]. ABySS is a memory efficient and relatively quick software but its performance is lower than that of other commonly used assemblers [116]. Platanus provides better results at higher read coverage compared to the other assemblers, it includes an algorithm to carefully assemble haplotype sequences that improves the resolution of heterozygous diploid genomes. Platanus produced better results for *S. fiordicum,* but BUSCO completeness was not high enough to convince us to include this genome in our project. Furthermore, I also checked the contamination level of the best assembly I managed to achieve for *S. fiordicum* using BlobTools v.2.1 [101] in the same way as described above in the decontamination step of the genome assembly pipeline.

2.8 - Genome annotation of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*

The genome annotation pipeline has the purpose of combining different sources of evidence, such as RNA-seq, *de novo* transcriptomes, inter-specific protein alignments and *ab initio* gene predictions, to identify the location of genes and predicting their function in the haploid versions of the genomes. This pipeline consists of eight main steps (**Fig. 6**): 1 -

Identification and soft-masking of the Transposable Elements (TEs), 2 – Generation of gene evidence using transcriptomic data, 3 – Combination of all the gene evidence into a single set which would then be used for gene predictions, 4 – Generation of *ab initio* gene predictions, 5 – Combining all the gene predictions into a final set of annotations, 6 – Filtering of spurious gene models to obtain a final set of *bona fide* gene models, 7 – Quality controls of the *de novo* annotations, 8 – Functional annotation of the gene models.

For each species the two main inputs that I used in the genome annotation pipeline are the haploid genome and the Illumina-based, tissue-specific RNA-seq libraries. In addition, in our version of *R. pachyptila* genome annotation I incorporated the transcriptomic data previously generated for this species by Hinzke *et al.* 2019 [34] (BioProject accession number PRJNA534438). These data comprise 22 RNA-seq libraries, five of which are coming from the plume, five from the vestimentum, six from the body wall and six from the trophosome.

2.8.1 - Step 1 – Annotation of transposable elements

The first step of the annotation pipeline (**Fig. 12**) is identifying and soft masking transposable elements (TEs) and other repeats in the three haploid assemblies. RepeatModeler v.2.0.1 [117] and RepBase [118] were used to build a consensus file, which is a *de novo* repeats library, for each of our three species. *Bona fide* genes were filtered out of the repeat libraries using Diamond BlastP v.0.8.22 [104] and the curated set of *O. fusiformis* proteins as database. Subsequently, RepeatMasker v.4.1.0 [119] was used to annotate the genomes using the consensus repeat predictions obtained previously. In parallel, LTR-finder v.1.07 [120] was used to identify and annotate Long Terminal Repeats (LTR). Then, I used RepeatCraft [121] to generate a consensus set of repeats by merging the output of RepeatMasker with the LTR-finder predictions for each genome. Finally, the “maskfasta” function of BEDtools v.2.30 [122]

allowed me to highlight the presence of TEs in the genomes by using lower case letters for the repetitive regions, obtaining in this way a soft-masked version of the assemblies for each species.

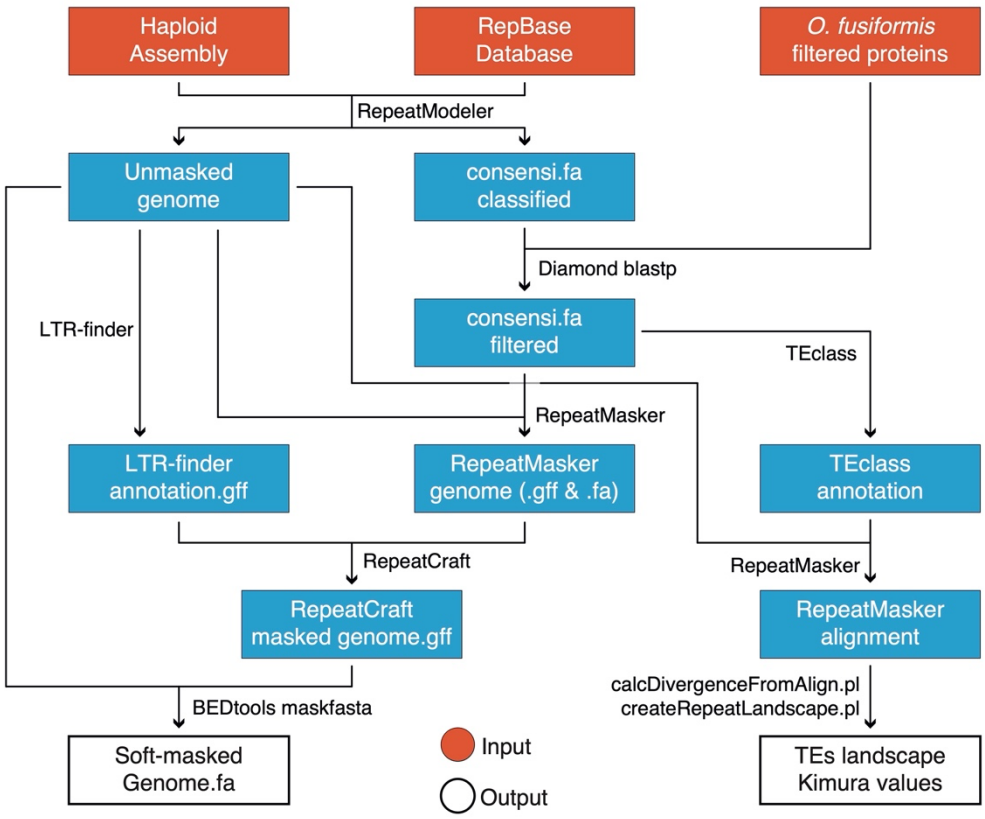


Figure 12 – Annotation of transposable elements: Visual representation of the steps involved in the annotation of transposable elements and in the generation of transposable elements landscapes with Kimura values. Boxes represent data files while arrows represent the software implemented in the pipeline.

Additionally, to explore the divergence of TEs in our genomes, I first used the online tool TEclass [123] to annotate the TEs previously identified by RepeatModeler. The machine learning oligomer-frequencies based approach offered by TEclass granted a higher percentage of annotated TEs compared to the RepeatModeler's sequence similarity approach. I used RepeatMasker v.4.1.0 [119] to obtain an alignment file from the unmasked version of the genomes and the TEclass annotations, which was used as input for the following step. Then I calculated the Kimura 2-parameters distances, which provided an estimation of genetic differences, of the TEs found in the three genomes using two scripts in the RepeatMasker suite,

calcDivergenceFromAlign.pl and a custom version of createRepeatLandscape.pl, which was fit to handle the TEclass annotations. Finally, I used ggplot2 v.3.3.0 [124] and Adobe Illustrator [125] to plot and polish the graphs.

2.8.2 - Step 2 – Generate gene evidence

In this step of the genome assembly pipeline, I mapped RNA-seq data against the genomes of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila* to identify translated regions in the focal genomes (**Fig. 13**).

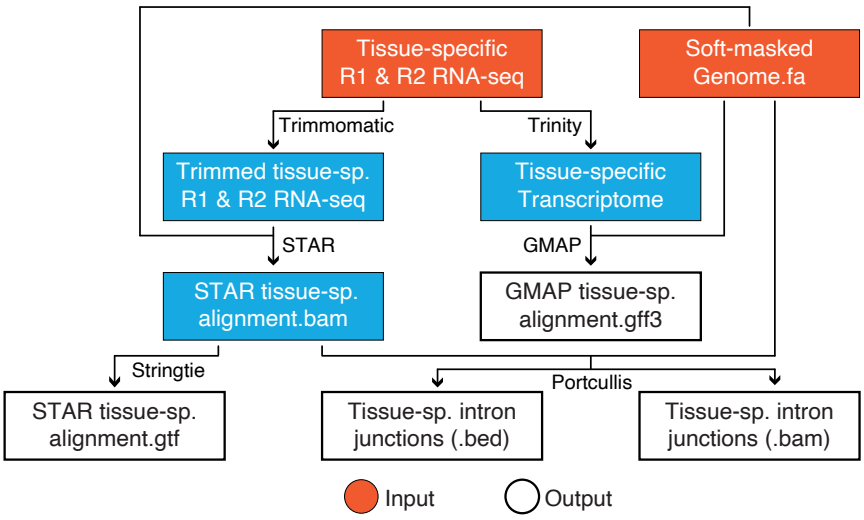


Figure 13 – Generation of gene evidence: Visual representation of the steps involved in the generation of gene evidences which would subsequently be used to obtain *ab initio* gene predictions and to generate a single gene set. Boxes represent data files while arrows represent the software implemented in the pipeline.

The short Illumina reads obtained from the transcriptome sequencing were used to generate *de novo* transcriptome assemblies with Trinity v.2.9.1 [126]. I generated in this way the transcriptomes of the crown, the opisthosoma and the trophosome for *Oasisia alvinae*, of the body and the root for *O. frankpressi*, and the transcriptomes of the crown and the trunk wall from the RNA-seq data obtained in this project for *R. pachyptila*. GMAP v.2017.09.30 [127] was used to map each of the transcriptome assemblies to their respective soft-masked

assemblies according to the species. In parallel, I trimmed the adaptors from the RNA-seq Illumina libraries with Trimmomatic v.0.35 [128] and aligned them to the soft-masked assemblies using STAR v.2.7.5a [129]. Next, I converted the BAM files generated with STAR into gene transfer format (GTF) files with StringTie v.2.1.2 [130] and I used Portcullis v.1.2.2 [131] to improve the prediction of intron junctions.

2.8.3 - Step 3 – Merge gene evidence

Before being able to use the gene evidence that I generated in the previous step as hints for the gene prediction, I needed to combine all evidence. Mikado v.2.Orc2 [132] (**Fig. 14**) allowed the selection of the best set of transcripts and curated splice junctions and the inclusion of them into a single transcriptome-based genomic annotation that can, thereafter, be used as hints for gene prediction. In the Mikado pipeline there is also a step using Junctools, included in the Portcullis v.1.2.2 [131] suite, to merge the tissue-specific intron junctions into a single file, a step using Diamond BlastX v.0.8.22 [104] and the reference proteomes database of UniProt [105] to generate protein homology for each of the transcripts provided to Mikado, and a step to provide predictions of the open reading frames using TransDecoder v.5.5.0 [133].

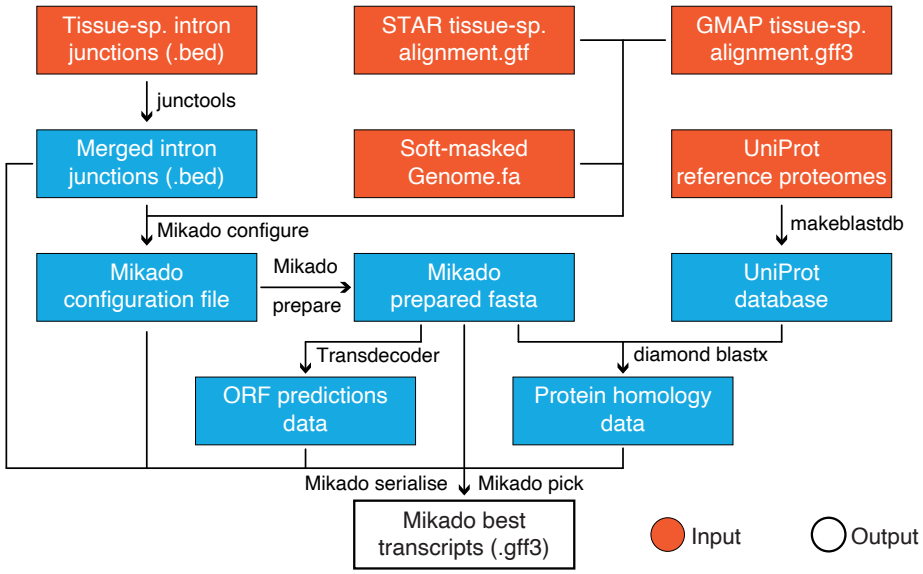


Figure 14 – Merging of the gene evidence: (**previous page**) Visual representation of the steps involved in combining the gene evidences previously obtained into a single file, which would subsequently be used as input for *ab initio* gene predictions and input in the generation of a single gene set. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.8.4 - Step 4 – Generate gene predictions

A crucial step in the genome assembly pipeline is the generation of *ab initio* gene predictions based on the evidence obtained in previous steps using the software Augustus v.3.3.3 [134] (**Fig. 15**). To run Augustus, I needed to generate further gene hints. Exonerate v.2.4.0 [135] was used to obtain exon hints by generating spliced alignments between curated proteomes of three related annelid species, *O. fusiformis, Capitella teleta* and *Lamellibrachia luymesi*, and the three soft-masked assemblies I previously obtained. The outputs of Exonerate were merged into a single file for our three siboglinid species employing a custom script developed by Dr Ferdinand Marlétaz. The tissue-specific alignment (.bam) files obtained previously with Portcullis were used to generate intron hints with the `bam2hints` script included in the Augustus suite. The last set of exon hints was obtained using a combination of Gffread v.0.12.1 [136] and the script `gtfToHintsMik.py` [137] from the best set of transcripts generated with Mikado. Then, I extracted full-length non-redundant transcripts from the output of Mikado

using the script `select_mik_train.py` [138] and I used them in combination with the soft-masked genome to generate an initial training set for Augustus for each of the three siboglinid species. Finally, the *ab initio* gene predictions were obtained with Augustus using as inputs the three sets of exon and intron hints, the initial training set and the soft-masked version of the genomes.

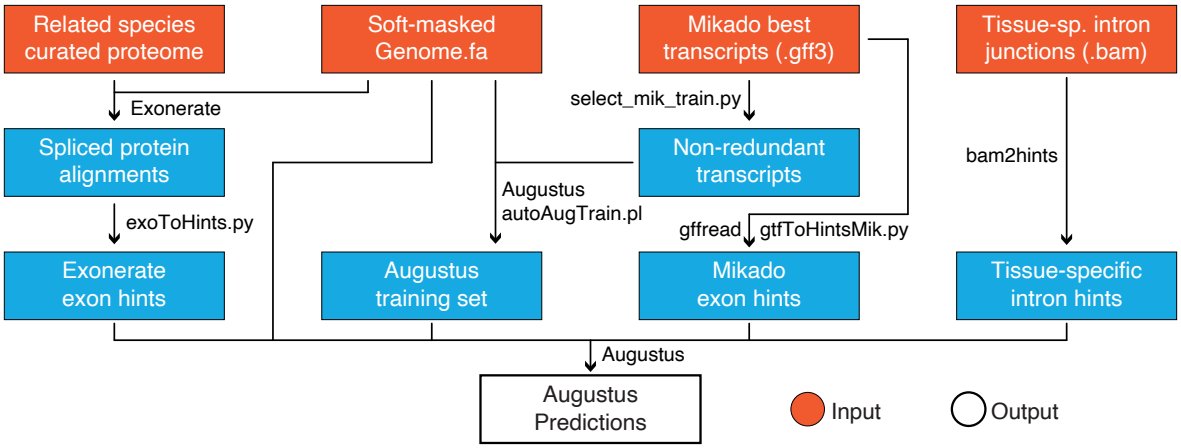


Figure 15 – Generation of gene predictions: Visual representation of the steps involved in generating *ab initio* gene predictions with Augustus. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.8.5 - Step 5 – Obtaining a single gene set

Next, I used PASA v.2.4.1 [139] to obtain a single gene set, for each species, by combining the predictions from Mikado and the *ab initio* gene annotations from Augustus (**Fig. 16**). To do so, I firstly removed the non-coding RNA (ncRNA) from the Mikado transcripts with a combination of AGAT v.0.5.0 [140] and Gffread v.0.12.1 [136]. Then, I used SeqClean [141] to identify evidence of polyadenylation, strip the poly-A, trim vector, and discard low quality sequences of the Mikado RNA transcripts files. PASA pipeline revolves around the build and update of a SQLite database [142] for each species, constructed using a genome fasta file and other sources of gene evidences. I used both the RNA Mikado transcripts and clean RNA Mikado transcripts to build an initial database with PASA. Then, I used Augustus output,

after a file conversion using Gffread, and the clean Mikado RNA transcripts to obtain the gff3 file that I used in a second step with PASA to generate the final single gene set in a gff3 format.

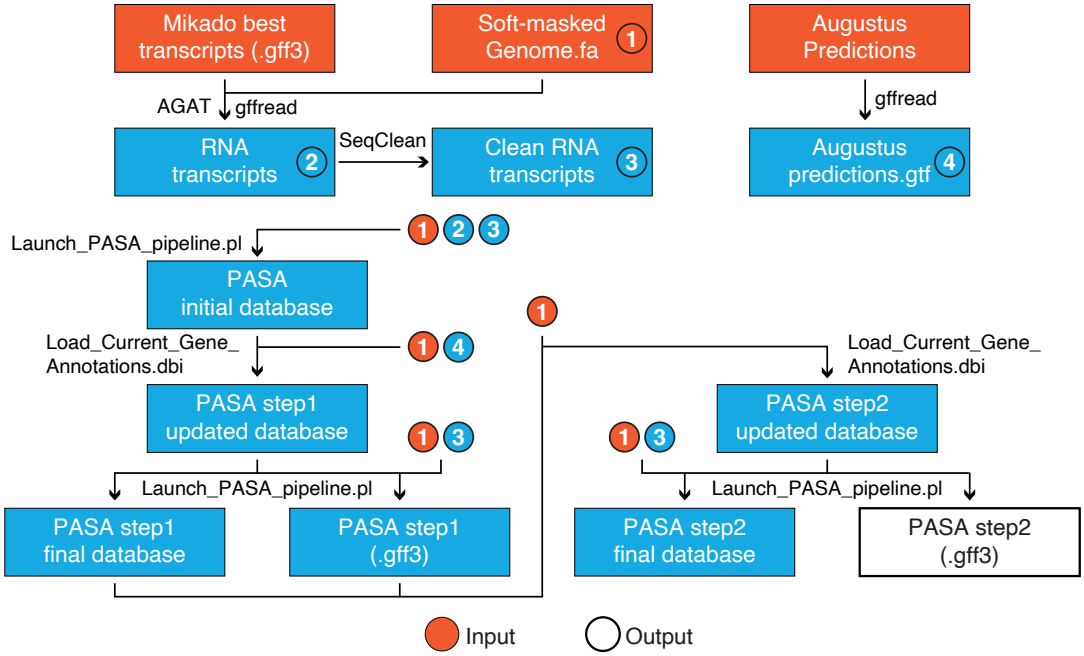


Figure 16 – Generation of a single gene set: Visual representation of the steps involved in combining *ab initio* gene predictions, Mikado best transcripts and soft-masked genomes into a single gene set file for each species. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.8.6 - Step6 – Filter gene models

The final set of annotations generated by PASA in the previous step needed to be filtered before assessing their quality (**Fig. 17**). I used a combination of AGAT v.0.5.0 [140] and Gffread v.0.12.1 [136] for most of the steps in this phase of the genome annotation pipeline. Firstly, I removed spurious gene models displaying in-frame stop codons. Then, the genes matching with a repeat sequence identified previously by RepeatMasker were discarded employing two Python scripts written by Dr Ferdinand Marlétaz. I also removed genes matching transposable elements, after checking the sequence similarity with the transposable elements contained in the RepeatMasker library "RepeatPeps" using Diamond BlastP v.0.8.22 [104]. Finally, I renamed the gene IDs and I obtained the final annotation gff3 file for *O.*

frankpressi, *Oasisia alvinae* and *R. pachyptila*. In addition, overlapping genes in the same position in the genome but in opposite DNA strands were merged together and, for each gene having one or more isoforms, I selected the isoform displaying the longest coding sequence using AGAT. This dataset is the non-redundant genome annotation and I used it for other analyses, such as the gene family evolutionary analyses.

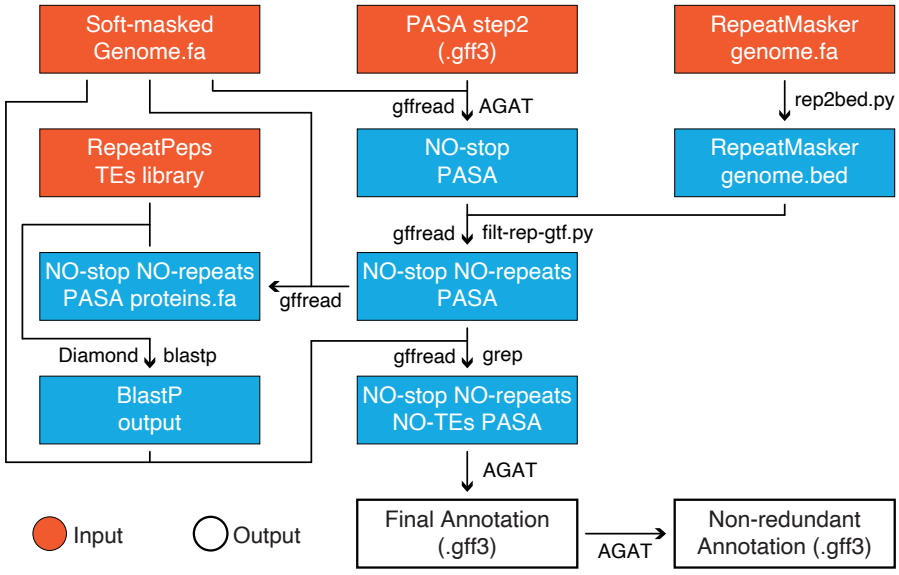


Figure 17 – Gene models filtering: Visual representation of the filtering strategy implemented to obtain a final annotation (gff3 file) and a non-redundant version of the final annotation for each species. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.8.7 - Step 7 – Validation

A quality assessment of the proteomes, extracted from the final annotations with a combination of AGAT and Gffread, was performed using BUSCO v.3.0.2 [88] with the Metazoa Odb10 database.

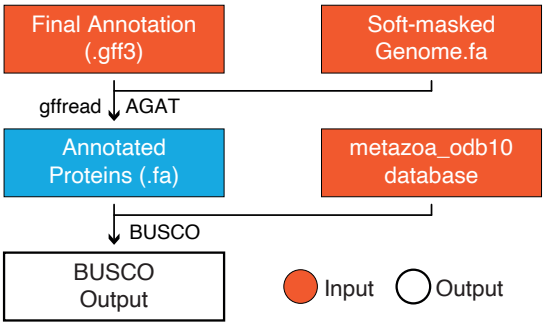


Figure 18 – Genome annotation validation: (**previous page**) Visual representation of validation strategy based on BUSCO of the final annotations. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.8.8 - Step 8 – Functional annotation

The final step of the pipeline was the functional annotation of the gene set, in which I compared the genes previously identified with databases of well annotated proteins to retrieve biological information (putative orthology, protein domains, gene ontology and KEGG annotation, etc) (**Fig. 19**). To achieve this, I used Trinotate v.3.2.1 [143] and Panther v.1.0.10 [144]. Trinotate compares the annotations generated by BLAST v.2.12.0+ [102], HMMER v.3.3.2 [145] and SignalP v.4.1 [146] to provide functional evidences and Panther is adopting the specificity of profile-Hidden Markov Model searches to compare the query proteins with its well supported database and provide further annotations to the final gene set. Finally, I obtained KEGG Orthology (KO) [147] numbers of the final gene models using the online tool KAAS [148]. The functional annotations of the genes identified in the three siboglinid genomes are crucial to characterise genomic events such as gene family expansions or reductions, therefore understanding the biological context of such events. The annotations I produced during this pipeline were thoroughly used in all our following analyses, they represent the basis of this PhD project from which I elaborated my considerations on the biology of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*.

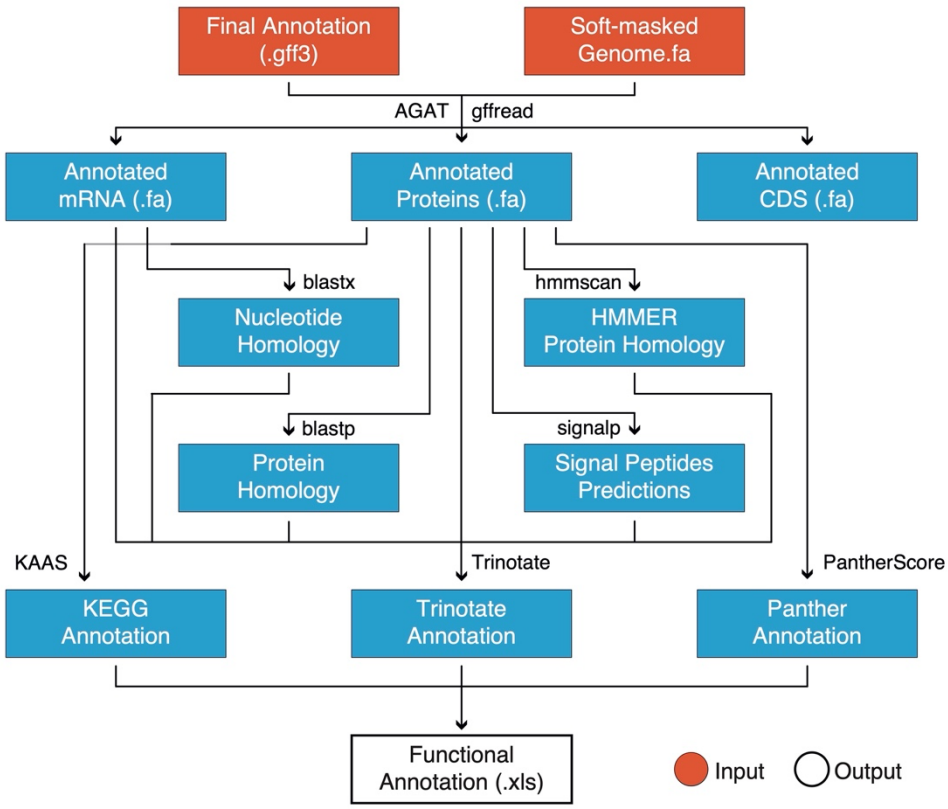


Figure 19 – Functional annotation: Visual representation of the strategy implemented to retrieve biological information for the gene models identified in the previous steps of the annotation pipeline. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.8.9 - Additional search for missing BUSCO in *Osedax*

The BUSCO completeness values of *O. frankpressi*, which I assessed during the genome assembly pipeline and genome annotation pipeline, are the lowest compared with the other two siboglinids (**Table 3**, **3.2 – Results and Discussion**). In total, 62 BUSCO genes are missing from *O. frankpressi* proteome. To verify if these genes are present in *Osedax frankpressi* genome but so divergent that fall through BUSCO's thresholds, I manually searched for these 62 genes in three different ways (**Fig. 20**). Firstly, I used a combination of AGAT v.0.5.0 [140], Gffread v.0.12.1 [136], BUSCO v.3.0.2 [88] and Seqtk v.1.3 [149] to obtain the protein sequences of the missing BUSCO genes from the BUSCO metazoa_odb10 database. Then I used BLAST v.2.12.0+ [102] (e-value: $1e^{-10}$) to compare the missing BUSCO

sequences with the proteome of *O. frankpressi*. In parallel, I functionally annotated the 62 missing BUSCO genes using Panther v.1.0.10 [144]. This allowed me to compare the Panther annotations with those obtained for *O. frankpressi* genome and verify the presence of any missing BUSCO genes. Additionally, I compared the Panther annotations of the missing BUSCO genes also with the Panther annotations of *Oasisia alvinae* and I extracted the matching *Oasisia alvinae* sequences. Finally, I used the *Oasisia alvinae* BUSCO-matching sequences as a query with BLAST (e-value: $1e^{-10}$) to search for homology in the proteome of *O. frankpressi*.

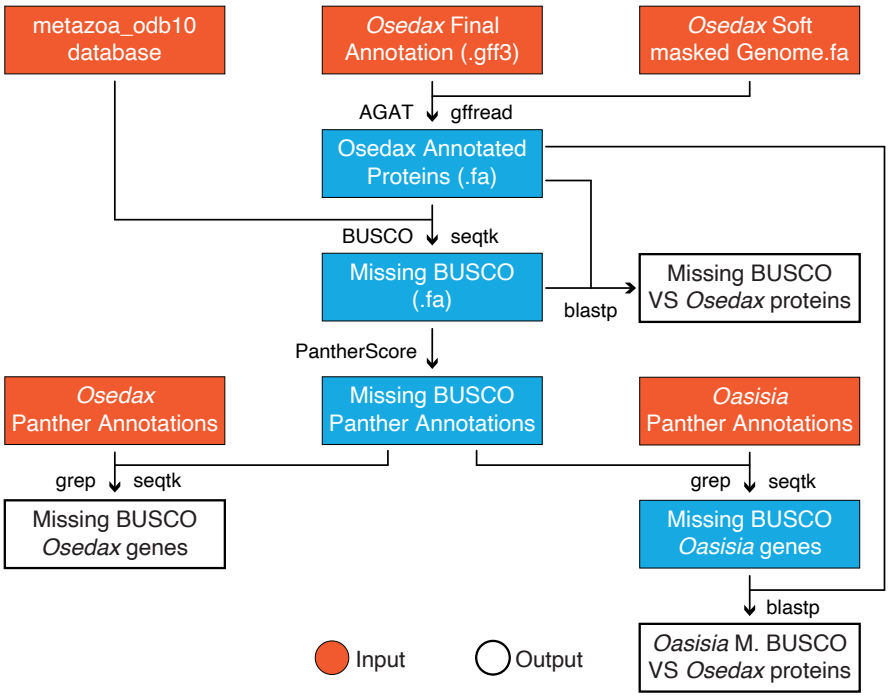


Figure 20 – Additional search for BUSCO genes in *O. frankpressi*: Visual representation of strategy implemented to retrieve the missing BUSCO genes in *O. frankpressi* genome. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.9 - *Riftia pachyptila* assembly and annotation comparison with previous versions

R. pachyptila is by far the best studied Siboglinidae. Oliveira *et al.* 2022 [69] produced a genome assembly and annotation of this species and Hinzke *et al.* 2019 [34] generated a

transcriptome. To evaluate the quality of our assembly for *R. pachyptila*, I compared our results with those two previous works [69] [34]. I analysed overall genomic stats using a combination of BUSCO v.3.0.2 [88], QUAST v.5.0.2 [89] and AGAT v.0.5.0 [140]. I used minimap2 v.2.17 [107] to align the *R. pachyptila* assembly reported in this project with the assembly generated by Oliveira *et al.* 2022 and the R package pafr [150] to generate a dot-plot representation of the sequence similarity between the two versions. Moreover, I performed a Reciprocal Best BLAST Hit (RBBH) method, using BLAST v.2.12.0+ [102], to identify and quantify the one-to-one BLAST matches between the gene models of *R. pachyptila* that I generated in this project and the gene models generated by Oliveira *et al.* 2022. Furthermore, I employed the same RBBH method to compare our *R. pachyptila* gene models with the non-redundant transcriptome obtained using a combination of Trinity v.2.9.1 [126] and cd-hit v.4.8.1 [151] from the raw RNA-seq sequences of *R. pachyptila* (BioProject accession number PRJNA534438) released by Hinzke *et al.* 2019. Finally, I assessed the number of different PFAM domains [152] identified in our version of *R. pachyptila* proteome, the proteome generated by Oliveira *et al.* 2022 and the transcriptome released by Hinzke *et al.* 2019 using PFAMscan v.1.6 [153].

2.10 - Assembly and annotation of the symbiont genomes

Balig Panossian assembled and annotated the genomes of the symbionts of *O. frankpressi* and *Oasisia alvinae.* Kraken2 v.2.1.0 [110] and Krakentools v.0.1 [110] were used to identify and isolate bacterial PacBio reads from the corrected PacBio reads generated with CANU during **2.5.1 – Initial assembly**. Metaflye v.2.9 [154] was then used to produce the initial bacterial assemblies and perform an initial polishing using the options `--pacbio-corr --meta --keep-haplotypes --iterations 10`. An additional polishing tool, NextPolish v.1.4.0 [155], was used to obtain the final assemblies of the symbionts of *O. frankpressi* and *Oasisia alvinae*.

The statistics and quality of the bacterial assemblies were evaluated using Bandage v.0.9.0 [156], which allows the user to visualise assembly graphs, MaxBin2 v.2.2.7 [157], to inspect the correct clustering of the contigs into bins according to their bacterial species, CheckM v.1.0.8 [158] and MetaQuast v.5.2.0 [159], to quantify the statistics of the binned bacterial contigs. In addition, the annotations for the genes in the bacterial assemblies were provided by Prokka v.1.14.5 [160] using the option `--compliant` to make sure the annotations were compliant with the Genbank/ENA/DDJB standards. Particular attention was dedicated to the identification of proteins involved in secretion systems, MacSyFinder v.2 [161] was used to identify matches with a curated collection of HMM profiles. Moreover, BlastKOALA v.2.2 [162] was used to assign KO numbers to the coding sequences, then KEGG Mapper v.5 [163] granted the visualisation of those KO numbers in maps which were used to assess the metabolic capabilities of each symbiont. Functional annotations, for the genes previously identified in the bacterial assemblies, were obtained by comparing the coding sequences with the NCBI COG database [164]. Furthermore, GSEA v.4.2.3 [165] and OrthoVenn2 v.2 [166] were used to run enrichment analyses based on functional categories and Gene Ontology terms. Accurate taxonomic classification of the bacterial genomes was provided by GTDB-Tk v.1.6.0 [167], including a curated collection of free living deep sea bacteria in the database used by the software. Finally, Circos v.0.69-9 [168] provided the visualisation of the bacterial genome assemblies.

2.11 - Gene family evolution analyses

Gene family analyses played an important role in this PhD project. They allowed me to find and characterise major changes in gene composition in the different lineages of Siboglinidae, providing fundamental data which I used as a base for my considerations on the evolution of siboglinids.

To obtain sufficient phylogenetic resolution during gene family analyses, I chose 25 species with complete genome assemblies and annotations from across the metazoan tree of life (**Fig. 21**), to include at least one species from the major animal phyla, from Cnidaria to Chordata. I included more annelid species than for any other group, seven of the 28 genomes, in order to achieve a good resolution of the annelid tree and evolutionary events leading to Siboglinidae.

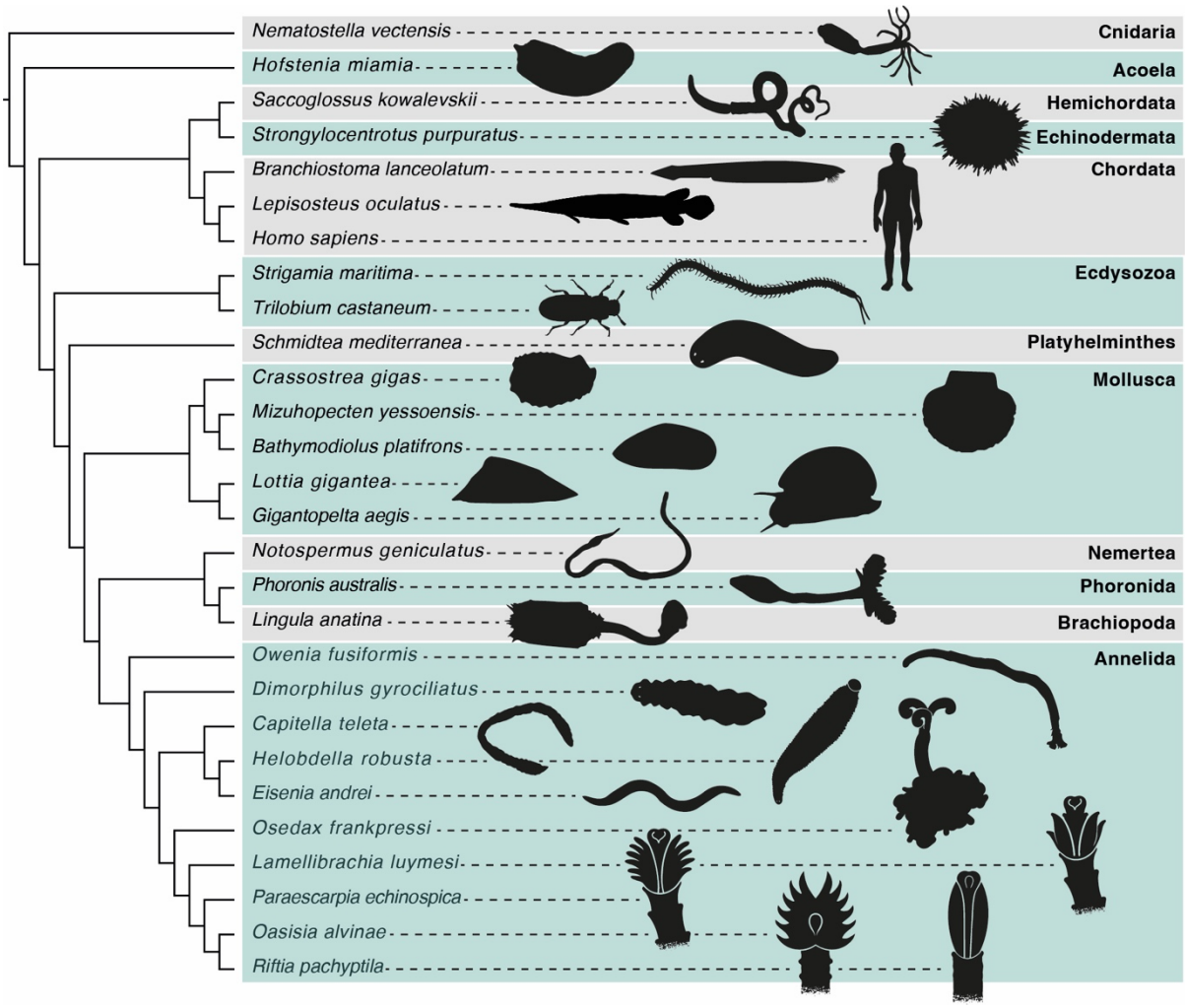


Figure 21 – Cladogram: Cladogram of the 28 metazoan species I included in the comparative analyses to identify gene family evolutionary events.

After obtaining the genomes from public available resources (**Supp. Table 8**), I generated non-redundant proteomes preserving only the longest isoform using AGAT v.0.5.0 [140] and then I removed spurious proteins using Gffread v.0.12.1 [136]. I inferred orthologous

groups using OrthoFinder v.2.5.2 [169], and I set it to run on the fast and accurate sequence similarity engine Diamond Blast v.2.0.9 [104] with the `--ultra-sensitive` option. OrthoFinder needed a Newick format tree of the phylogenetic relationships between the 28 species which I deduced from published literature [170] [171] [172]. Previously published code [173] allowed me to calculate gene family gains, expansions and losses at each node of the phylogenetic tree, of the 28 species included in these analyses, by using a hypergeometric test against the median gene number per species for a given family. The OrthoFinder output "Orthogroups.GeneCount.tsv" and the R function `prcomp` [174] were used to run the principal component analyses on those gene families which are present in at least three of the 28 species I included in this analysis. Furthermore, I used the R package TopGO v.2.42.0 [175] to perform enrichment analyses based on the GO terms of the genes belonging to expanded, gained, and lost gene families in all the siboglinid species included in the study and at each node of the siboglinid tree.

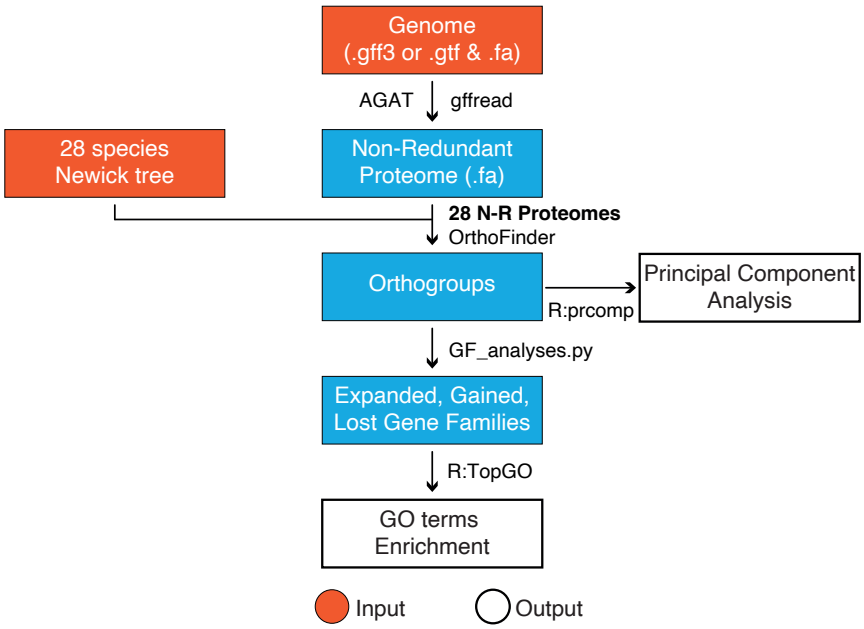


Figure 22 – Gene family analyses: Visual representation of the steps involved in the identification of gene families in the 28 metazoan species. The strategies implemented to run principal component analysis and GO terms enrichment analysis are also included in the figure. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.12 - Reconstruction of gene pathways

I used Panther IDs to assess the presence, absence and gene copy numbers by of each enzyme involved in 20 different amino acids biosynthetic pathways, biosynthesis of eight B vitamins, nitrogen metabolism, glycine degradation and seven different DNA repair pathways. I used Panther IDs because they are based on HMM profiles, which have a higher level of sensitivity on protein domain structures, thus granting more accuracy on the identification of homologs compared to sequence similarity by BLAST [145].

Information about each step in a pathway were collected from the databases MetaCyc [176] and KEGG [147]. The Panther IDs for the enzymes involved in those pathways were obtained from the Panther database [144] using the search function with the gene information obtained previously. This approach allowed me to compare the Panther IDs in my annotation to those of enzymes in the pathways of interest and obtain the copy numbers of the respective gene in the three genomes generated in this study. In addition, I functionally annotated, using Trinotate v.3.2.1 [143] and Panther v.1.0.10 [144] as described in the genome annotation pipeline step 8, the public-available genomes of two additional siboglinids, *L.luymesi* and *P. echinospica*. I also used the functional annotations already available for *O. fusiformis* and *C. teleta* [177]. Finally, ggplot2 v.3.3.0 [124], pheatmap v.1.0.12 [178] and Adobe Illustrator [125] were used to obtain the results as heatmaps.

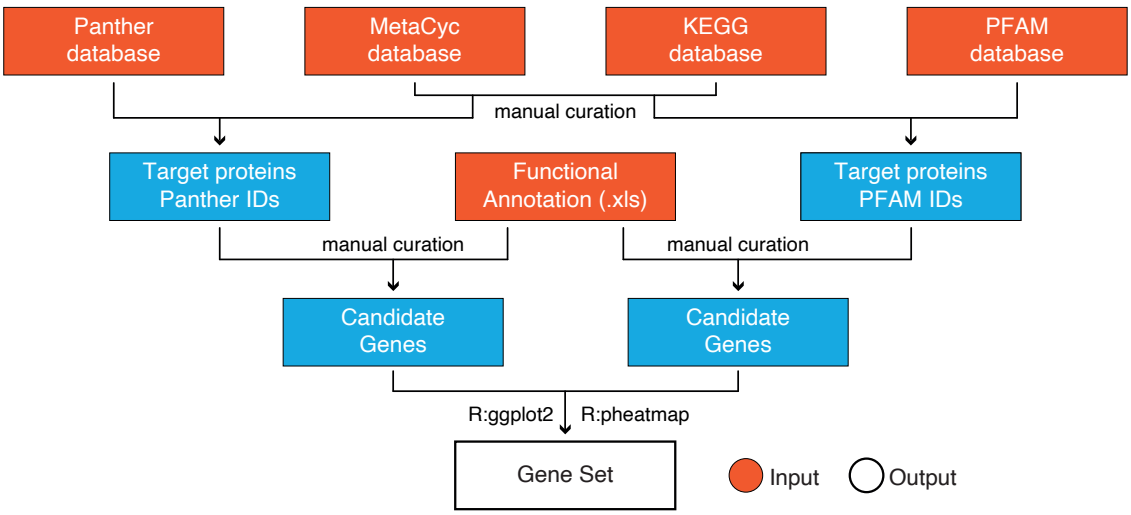


Figure 23 – Reconstruction of gene pathways: Visual representation of the strategy implemented to reconstruct the presence of functional gene pathways in the species included in these analyses. Boxes represent data files while arrows represent the software implemented in the pipeline.

Moreover, the repertoire of transcription factors, signalling ligands and receptors in the seven above-mentioned species was explored in a similar way using both the Panther IDs and the PFAM annotations, both generated with the functional annotation. I retrieved the PFAM annotations of each gene of interests using both published literature [173] and the PFAM database [152]. Panther IDs were also used to search for putative Matrix-Metalloproteinases (MMPs), Hox and ParaHox genes, Wnt, Fzl, BMP and BMP receptors in order to select candidate genes to use as input for gene orthology assignments.

2.13 – Reconstruction of KEGG pathways

I used KEGG mapper to reconstruct and visualise all metabolic pathways in the siboglinids *Oasisia alvinae*, *O. frankpressi*, *R. pachyptila*, *L. luymesi*, *P. echinospica*, and the asymbiotic annelids *O. fusiformis* and *C. teleta*. As already described in **2.8.8 - Step 8 – Functional annotation** for *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*, I also assigned KEGG Orthology (KO) [147] numbers to the gene models of *L.luymesi*, *P. echinospica*, *O.*

fusiformis and *C. teleta* using the online tool KAAS [148], which is based on the single best BLAST hit method. In addition, I used KofamKOALA (kofamscan v.1.3.0) [179] to assign further KO numbers to the gene models of the seven annelid species. KofamKOALA uses HMMER to compare query proteins against a database of KEGG orthologs. The combined lists of KO numbers generated with the two methods for each species were used as input for the online tool KEGG mapper [163] which enabled the reconstruction and visualisation of all the KEGG supported pathways identified in the seven species.

2.14 - Expression analyses

Additionally, I compared the expression of amino acid biosynthetic enzymes, enzymes involved in the glycine degradation and MMPs between the body and the roots of *O. frankpressi.* RNA-seq Illumina fastq files were trimmed using Trimmomatic v.0.36 [128] and gene models were extracted from the final annotation file of the genome of *O. frankpressi* using Gffread v.0.12.1 [136]. Then, I used Kallisto v.0.46.2 [180] to quantify the abundances of transcripts in the trimmed RNA-seq data and obtain the TPM (Transcripts per Kilobase Million) values for all the genes that I identified in the genome of *O. frankpressi*. I normalised the TPM values for all genes with the R function `data.matrix(scale(TPM_allGenes))` and I selected the values just for genes of interest. Finally, I obtained a visual representation using ggplot2 v.3.3.0 [124], pheatmap v.1.0.12 [178] and Adobe Illustrator [125].

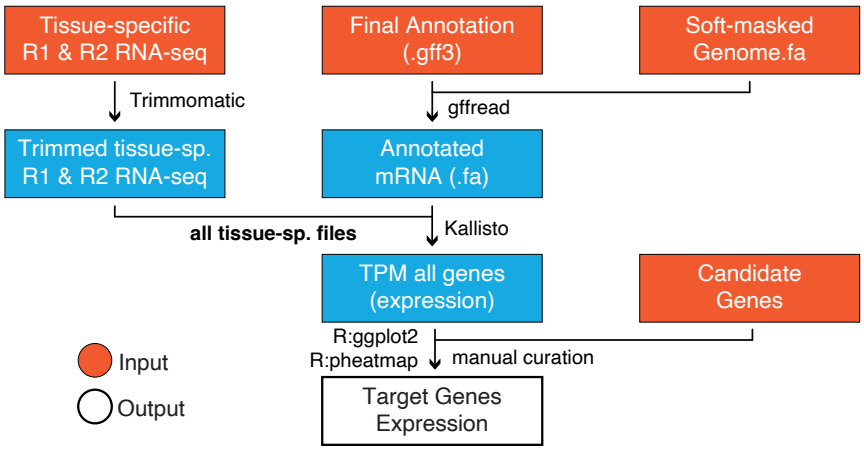


Figure 24 – Expression analyses: Visual representation of steps involved in the quantification of the gene expression levels in the tissues of *O. frankpressi*, *Oasisia alvinae* and *R. pachyptila*. Boxes represent data files while arrows represent the software implemented in the pipeline.

2.15 - Orthology assignments

The initial selection of putative MMPs, Hox and ParaHox genes, Wnt, Fzl, BMP and BMP receptors was based on the Panther IDs, as described previously in **2.12 - Reconstruction of gene pathways**. In addition, for Hox and ParaHox genes I selected also the candidate genes showing sequence similarity with Hox genes of *O. fusiformis* and ParaHox genes of both *O. fusiformis* and *C. teleta* using Diamond BlastP v.0.8.22 [104]. I used the online tool MAFFT [181], with the default options, to align candidate sequences to a curated set of proteins that I obtained either from previous studies (Hox and ParaHox genes, Wnt, Fzl, BMP and BMP receptors) [173], [182] or manually from Uniprot [105] (MMPs). Then, I used Jalview v.2.11.2.2 [183] to visualise and trim the alignment in order to focus my analyses only on the conserved domains and to remove divergent sequences from the dataset. The alignment file was submitted a second time to MAFFT, this time with the option `L-INS-I`, which performs better when only one domain is present in the alignment, generating a new alignment file that I visualised and optionally trimmed again using Jalview. After a final trim using trimAI

v.1.4.rev15 [184], the reconstruction of the phylogeny for each gene family was obtained with FastTree v.2.1.10 [185], using default options.

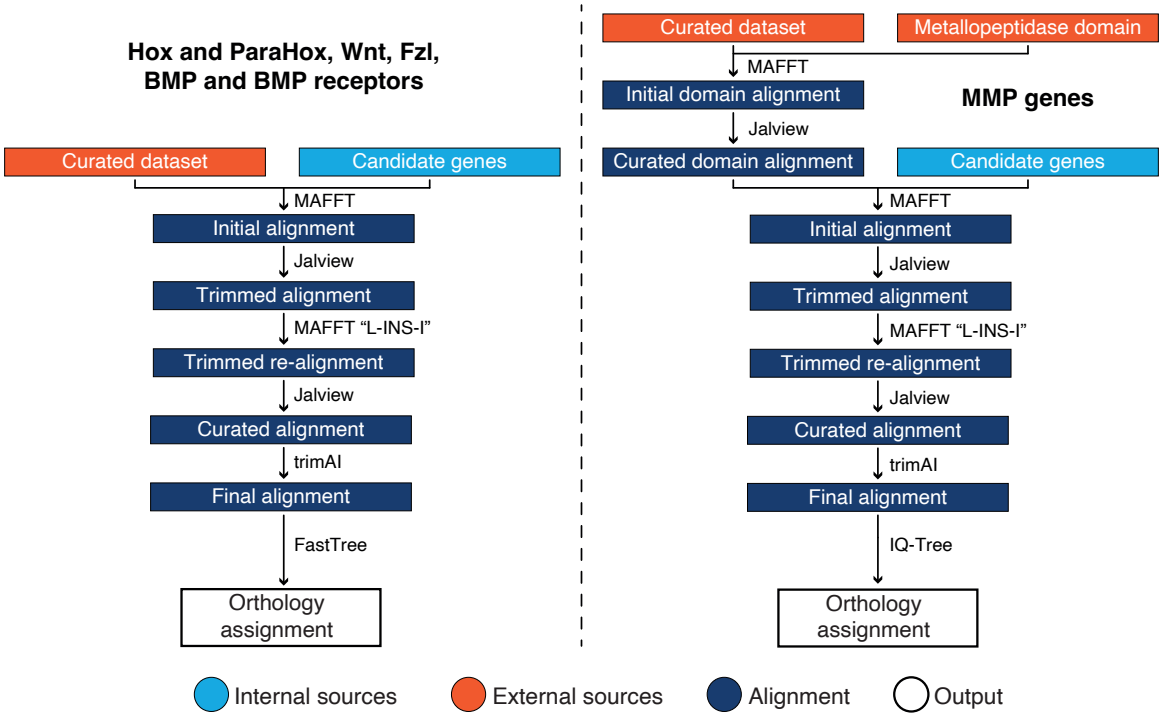


Figure 25 – Orthology assignments: Visual representation of the two strategies used to reconstruct phylogenetic relationships between selected candidate genes. Boxes represent data files while arrows represent the software implemented in the pipeline.

Due to the complexity of the MMPs phylogeny, I added a first step in which I aligned and trimmed just the metallopeptidase domain with MMPs sequences obtained from Uniprot. Then, I calculated the MMPs orthology using IQ-Tree v.2.2.0-beta [186] instead of FastTree. Both FastTree and IQ-Tree are based on the maximum likelihood method. FastTree is fast but cannot handle many sequences while keeping sufficient accuracy. On the other hand, IQ-Tree runs much slower, but it incorporates also stochastic techniques to produce the best possible tree for a set of sequences. I selected IQ-Tree with the options `-m MFP` to perform an extended model selection for the input data and run the tree using the best-fit model, and `-B 1000` to run 1,000 bootstrap replicates. Finally, I obtained the graphical representation of the phylogenetic relationships in the different sets of proteins using a combination of FigTree v.1.4.4 [187] and

Adobe Illustrator [125]. In addition, Francisco M. Martin-Zamora reconstructed the phylogeny of MMPs proteins using MrBayes [188], which uses a Markov chain Monte Carlo method to estimate the phylogenetic relationships among target sequences.

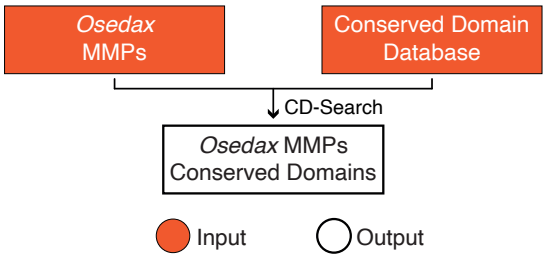


Figure 26 – Domain structures characterisation: Visual representation of the strategy to characterise the domain structures of *O. frankpressi*'s MMPs. Boxes represent data files while arrows represent the software implemented in the pipeline.

In addition, I characterised the domain structures of the 63 MMPs I identified in *O. frankpressi,* using the online function of CD-Search [189] with default options. This tool allowed me to compare the 63 MMPs sequences with the Conserved Domain Database (CDD) [190] and annotate the identified domains.

2.16 - Reconstruction of innate immune repertoires

Dr. Yanan Sun reconstructed the innate immune repertoire of selected annelids focusing on the content of pattern recognition receptors in *O. frankpressi*, Vestimetifera and two asymbiotic annelids, *Owenia fusiformis* and *C. teleta*. To do so, she compared the previously published pattern recognition receptors of Vestimentifera [71] with the gene families identified by Orthofinder in **2.11 – Gene family evolutionary analyses**. Then, fragmented proteins and proteins lacking the target domain were filtered out using the functional annotations generated during **2.8.8 - Step 8 – Functional annotation**. Finally, TPM values, obtained in **2.14 - Expression analyses**, in combination with TBtools v.1.042 [191] were used to obtain the pattern recognition receptors expression heatmaps.

2.17 - Reconstruction of the G protein-coupled receptor (GPCR) repertoire

Dr. Daniel Thiel characterised the G protein-coupled receptor (GPCR) repertoire of the species included in this project. Following an already published procedure [192], a dataset was built by downloading and processing the transcriptomes of focal species. Then, hmmer-3.1b2 [193], with an e-value cut-off of 1e-10, was used to generate HMM profiles from sequences of rhodopsin type GPCRs (PF00001), secretin type GPCRs (PF00002), glutamate type GPCRs (PF00003) and frizzled type GPCRs (PF01534), which were previously obtained from the Pfam webpage [194]. CLANS [195] is a tool which enable the functional annotation of query proteins by reconstructing a network of pairwise sequence similarities. The online version of CLANS [196] was used for the initial reconstruction of pairwise similarities based on BLAST, the edges with a score value lower than 1e-10 were removed before proceeding with the analyses. In addition, the offline version of CLANS was then used for the main cluster analysis, using a P-value of 1e-30 and selecting the edges with a score value up to 1e-15. Linkage clustering allowed the identification and the removal of sequences without pairwise connections. In addition, the vertebrate olfactory GPCR type-A receptor sequences were removed from the dataset because they are highly vertebrate specific, therefore they show no connections and strongly repulsed all other sequences. Finally, the annotation of the different gene clusters identified in these analyses was based on the presence of well characterised sequences *Drosophila melanogaster, Homo sapiens, Danio rerio* and *Platynereis dumerilii* within each cluster.

References

[34] T. Hinzke *et al.*, “Host-microbe interactions in the chemosynthetic Riftia pachyptila symbiosis,” *bioRxiv*, p. 651323, May 2019.

[69] A. L. De Oliveira, J. Mitchell, P. Girguis, and M. Bright, “Novel Insights on Obligate Symbiont Lifestyle and Adaptation to Chemosynthetic Environment as Revealed by the Giant Tubeworm Genome,” *Mol. Biol. Evol.*, vol. 39, no. 1, Jan. 2022.

[85] S. C. S. Andrade *et al.*, “Articulating ‘Archiannelids’: Phylogenomics and Annelid Relationships, with Emphasis on Meiofaunal Taxa,” *Mol. Biol. Evol.*, vol. 32, no. 11, pp. 2860–2875, Nov. 2015.

[86] T. King, S. Butcher, and L. Zalewski, “Apocrita - High Performance Computing Cluster for Queen Mary University of London,” Mar. 2017.

[87] “Anaconda | Anaconda Distribution.” [Online]. Available: https://www.anaconda.com/products/distribution. [Accessed: 12-Jul-2022].

[88] F. A. Simão, R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, and E. M. Zdobnov, “BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs,” *Bioinformatics*, vol. 31, no. 19, pp. 3210–3212, Oct. 2015.

[89] A. Gurevich, V. Saveliev, N. Vyahhi, and G. Tesler, “QUAST: quality assessment tool for genome assemblies,” *Bioinformatics*, vol. 29, no. 8, pp. 1072–1075, Apr. 2013.

[90] S. Koren, B. P. Walenz, K. Berlin, J. R. Miller, N. H. Bergman, and A. M. Phillippy, “Canu: scalable and accurate long-read assembly via adaptive *k* -mer weighting and repeat separation,” *Genome Res.*, vol. 27, no. 5, pp. 722–736, May 2017.

[91] J. R. Miller *et al.*, “Aggressive assembly of pyrosequencing reads with mates,” *Bioinformatics*, vol. 24, no. 24, p. 2818, Dec. 2008.

[92] R. Chikhi and P. Medvedev, “Informed and automated k-mer size selection for genome assembly.,” *Bioinformatics*, vol. 30, no. 1, pp. 31–7, Jan. 2014.

[93] "GitHub - PacificBiosciences/bam2fastx: Converting and demultiplexing of PacBio BAM files into gzipped fasta and fastq files." [Online]. Available: https://github.com/PacificBiosciences/bam2fastx. [Accessed: 28-Jun-2022].

[94] S. B. Kingan *et al.*, "A high-quality genome assembly from a single, field-collected spotted lanternfly (Lycorma delicatula) using the PacBio Sequel II system," *Gigascience*, vol. 8, no. 10, Oct. 2019.

[95] B. J. Walker *et al.*, "Pilon: An Integrated Tool for Comprehensive Microbial Variant Detection and Genome Assembly Improvement," *PLoS One*, vol. 9, no. 11, p. e112963, Nov. 2014.

[96] "PacificBiosciences/pbmm2: A minimap2 frontend for PacBio native data formats." [Online]. Available: https://github.com/PacificBiosciences/pbmm2. [Accessed: 06-Apr-2021].

[97] M. Martin, "Cutadapt removes adapter sequences from high-throughput sequencing reads," *EMBnet.journal*, vol. 17, no. 1, p. 10, May 2011.

[98] S. Andrews, "FastQC: a quality control tool for high throughput sequence data," *2010*. [Online]. Available: http://www.bioinformatics.babraham.ac.uk/projects/fastqc.

[99] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows-Wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, Jul. 2009.

[100] H. Li *et al.*, "The Sequence Alignment/Map format and SAMtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009.

[101] D. R. Laetsch and M. L. Blaxter, "BlobTools: Interrogation of genome assemblies," *F1000Research*, vol. 6, p. 1287, Jul. 2017.

[102] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, 1990.

[103] "Nucleotide - NCBI. Nucleotide [Internet]. Bethesda (MD): National Library of

Medicine (US), National Center for Biotechnology Information; [1988].” [Online]. Available: https://www.ncbi.nlm.nih.gov/nucleotide/. [Accessed: 29-Jun-2022].

[104] B. Buchfink, C. Xie, and D. H. Huson, “Fast and sensitive protein alignment using DIAMOND,” *Nature Methods*, vol. 12, no. 1. Nature Publishing Group, pp. 59–60, 01-Jan-2014.

[105] A. R *et al.*, “UniProt: the Universal Protein knowledgebase,” *Nucleic Acids Res.*, vol. 32, no. Database issue, pp. 13–14, 2004.

[106] D. Guan, S. A. McCarthy, J. Wood, K. Howe, Y. Wang, and R. Durbin, “Identifying and removing haplotypic duplication in primary genome assemblies,” *bioRxiv*, p. 729962, Aug. 2019.

[107] H. Li, “Minimap2: Pairwise alignment for nucleotide sequences,” *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, 2018.

[108] D. Mapleson, G. G. Accinelli, G. Kettleborough, J. Wright, and B. J. Clavijo, “KAT: A K-mer analysis toolkit to quality control NGS datasets and genome assemblies,” *Bioinformatics*, vol. 33, no. 4, pp. 574–576, Feb. 2017.

[109] N. De Maio *et al.*, “Comparison of long-read sequencing technologies in the hybrid assembly of complex bacterial genomes,” *Microb. Genomics*, vol. 5, no. 9, Sep. 2019.

[110] D. E. Wood, J. Lu, and B. Langmead, “Improved metagenomic analysis with Kraken 2,” *bioRxiv*, p. 762302, Sep. 2019.

[111] S. Chen, Y. Zhou, Y. Chen, and J. Gu, “fastp: an ultra-fast all-in-one FASTQ preprocessor,” *Bioinformatics*, vol. 34, no. 17, pp. i884–i890, Sep. 2018.

[112] T. R. Ranallo-Benavidez, K. S. Jaron, and M. C. Schatz, “GenomeScope 2.0 and Smudgeplot for reference-free profiling of polyploid genomes,” *Nat. Commun.*, vol. 11, no. 1, pp. 1–10, Dec. 2020.

[113] Computational Biology Core, “Genome Size Estimation Tutorial.” [Online]. Available:

https://bioinformatics.uconn.edu/genome-size-estimation-tutorial/#outline4. [Accessed: 07-Apr-2021].

[114] R. Kajitani *et al.*, "Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads," *Genome Res.*, vol. 24, no. 8, pp. 1384–1395, Apr. 2014.

[115] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and I. Birol, "ABySS: A parallel assembler for short read sequence data," *Genome Res.*, vol. 19, no. 6, pp. 1117–1123, Jun. 2009.

[116] J. Sohn and J.-W. Nam, "The present and future of *de novo* whole-genome assembly," *Brief. Bioinform.*, vol. 19, no. 1, p. bbw096, Oct. 2016.

[117] J. M. Flynn *et al.*, "RepeatModeler2 for automated genomic discovery of transposable element families," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 117, no. 17, pp. 9451–9457, Apr. 2020.

[118] W. Bao, K. K. Kojima, and O. Kohany, "Repbase Update, a database of repetitive elements in eukaryotic genomes," *Mob. DNA*, vol. 6, no. 1, pp. 1–6, Jun. 2015.

[119] R. Smit, AFA, Hubley, "RepeatModeler," *RepeatModeler Open-1.0. 2008-2015*. [Online]. Available: http://www.repeatmasker.org.

[120] Z. Xu and H. Wang, "LTR-FINDER: An efficient tool for the prediction of full-length LTR retrotransposons," *Nucleic Acids Res.*, vol. 35, no. SUPPL.2, p. W265, Jul. 2007.

[121] W. Y. Wong and O. Simakov, "RepeatCraft: a meta-pipeline for repetitive element de-fragmentation and annotation," *Bioinformatics*, vol. 35, no. 6, pp. 1051–1052, Mar. 2019.

[122] A. R. Quinlan and I. M. Hall, "BEDTools: a flexible suite of utilities for comparing genomic features," *Bioinformatics*, vol. 26, no. 6, pp. 841–842, Mar. 2010.

[123] G. Abrusán, N. Grundmann, L. DeMester, and W. Makalowski, "TEclass--a tool for

automated classification of unknown eukaryotic transposable elements.," *Bioinformatics*, vol. 25, no. 10, pp. 1329–30, May 2009.

[124] H. Wickham, *ggplot2, Elegant Graphics for Data Analysis*. Springer New York, 2009.

[125] "Industry-leading vector graphics software | Adobe Illustrator." [Online]. Available: https://www.adobe.com/uk/products/illustrator.html. [Accessed: 04-Jul-2022].

[126] B. J. Haas *et al.*, "De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis.," *Nat. Protoc.*, vol. 8, no. 8, pp. 1494–512, Aug. 2013.

[127] T. D. Wu and C. K. Watanabe, "GMAP: a genomic mapping and alignment program for mRNA and EST sequences," *Bioinformatics*, vol. 21, no. 9, pp. 1859–1875, May 2005.

[128] A. M. Bolger, M. Lohse, and B. Usadel, "Trimmomatic: A flexible trimmer for Illumina sequence data," *Bioinformatics*, vol. 30, no. 15, pp. 2114–2120, Aug. 2014.

[129] A. Dobin *et al.*, "STAR: Ultrafast universal RNA-seq aligner," *Bioinformatics*, vol. 29, no. 1, pp. 15–21, Jan. 2013.

[130] M. Pertea, G. M. Pertea, C. M. Antonescu, T. C. Chang, J. T. Mendell, and S. L. Salzberg, "StringTie enables improved reconstruction of a transcriptome from RNA-seq reads," *Nat. Biotechnol. 2015 333*, vol. 33, no. 3, pp. 290–295, Feb. 2015.

[131] D. Mapleson, L. Venturini, G. Kaithakottil, and D. Swarbreck, "Efficient and accurate detection of splice junctions from RNAseq with Portcullis," *bioRxiv*, p. 217620, Nov. 2017.

[132] L. Venturini, S. Caim, G. G. Kaithakottil, D. L. Mapleson, and D. Swarbreck, "Leveraging multiple transcriptome assembly methods for improved gene structure annotation," *Gigascience*, vol. 7, no. 8, Aug. 2018.

[133] B. J. Haas, "GitHub - TransDecoder/TransDecoder: TransDecoder source." [Online]. Available: https://github.com/TransDecoder/TransDecoder. [Accessed: 05-Jul-2022].

[134] M. Stanke, O. Keller, I. Gunduz, A. Hayes, S. Waack, and B. Morgenstern, "AUGUSTUS: ab initio prediction of alternative transcripts," *Nucleic Acids Res.*, vol. 34, no. Web Server, pp. W435–W439, Jul. 2006.

[135] G. S. C. Slater and E. Birney, "Automated generation of heuristics for biological sequence comparison," *BMC Bioinformatics*, vol. 6, pp. 31–31, Feb. 2005.

[136] G. Pertea and M. Pertea, "GFF Utilities: GffRead and GffCompare," *F1000Research*, vol. 9, p. 304, Apr. 2020.

[137] "gtfToHintsMik.py." [Online]. Available: https://www.dropbox.com/s/wqmwlj6c9dn440l/gtfToHintsMik.py?dl=0. [Accessed: 06-Jul-2022].

[138] "select_mik_train.py." [Online]. Available: https://www.dropbox.com/s/kgtty04uwwfig67/select_mik_train.py?dl=0,. [Accessed: 06-Jul-2022].

[139] B. J. Haas *et al.*, "Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies," *Nucleic Acids Res.*, vol. 31, no. 19, pp. 5654–5666, Oct. 2003.

[140] J. Dainat, "AGAT: Another Gff Analysis Toolkit to handle annotations in any GTF/GFF format," *Zenodo. Version v0.4.0*, Jun-. [Online]. Available: https://www.doi.org/10.5281/zenodo.3552717. [Accessed: 06-Apr-2021].

[141] "SeqClean." [Online]. Available: https://sourceforge.net/projects/seqclean/. [Accessed: 06-Apr-2021].

[142] "SQLite Home Page." [Online]. Available: https://www.sqlite.org/index.html. [Accessed: 07-Jul-2022].

[143] D. M. Bryant *et al.*, "A Tissue-Mapped Axolotl De Novo Transcriptome Enables Identification of Limb Regeneration Factors," *Cell Rep.*, vol. 18, no. 3, pp. 762–776,

Jan. 2017.

[144] P. D. Thomas *et al.*, “PANTHER: A library of protein families and subfamilies indexed by function,” *Genome Res.*, vol. 13, no. 9, pp. 2129–2141, Sep. 2003.

[145] R. D. Finn, J. Clements, and S. R. Eddy, “HMMER web server: Interactive sequence similarity searching,” *Nucleic Acids Res.*, vol. 39, no. SUPPL. 2, p. W29, Jul. 2011.

[146] J. J. Almagro Armenteros *et al.*, “SignalP 5.0 improves signal peptide predictions using deep neural networks,” *Nat. Biotechnol.*, vol. 37, no. 4, pp. 420–423, Apr. 2019.

[147] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, “KEGG as a reference resource for gene and protein annotation,” *Nucleic Acids Res.*, vol. 44, no. D1, pp. D457–D462, Jan. 2016.

[148] Y. Moriya, M. Itoh, S. Okuda, A. C. Yoshizawa, and M. Kanehisa, “KAAS: An automatic genome annotation and pathway reconstruction server,” *Nucleic Acids Res.*, vol. 35, no. SUPPL.2, p. W182, Jul. 2007.

[149] “GitHub - lh3/seqtk: Toolkit for processing sequences in FASTA/Q formats.” [Online]. Available: https://github.com/lh3/seqtk. [Accessed: 08-Jul-2022].

[150] D. Winter, “Read, manipulate and visualize Pairwise mApping Format data • pafr.” [Online]. Available: https://dwinter.github.io/pafr/index.html. [Accessed: 24-Jan-2023].

[151] W. Li and A. Godzik, “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences,” *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, Jul. 2006.

[152] R. D. Finn *et al.*, “Pfam: The protein families database,” *Nucleic Acids Research*, vol. 42, no. D1. Nucleic Acids Res, 01-Jan-2014.

[153] J. Mistry, A. Bateman, and R. D. Finn, “Predicting active site residue annotations in the Pfam database,” *BMC Bioinformatics*, vol. 8, no. 1, pp. 1–14, Aug. 2007.

[154] M. Kolmogorov *et al.*, “metaFlye: scalable long-read metagenome assembly using

repeat graphs," *Nat. Methods 2020 1711*, vol. 17, no. 11, pp. 1103–1110, Oct. 2020.

[155] J. Hu, J. Fan, Z. Sun, and S. Liu, "NextPolish: a fast and efficient genome polishing tool for long-read assembly," *Bioinformatics*, vol. 36, no. 7, pp. 2253–2255, Apr. 2020.

[156] R. R. Wick, M. B. Schultz, J. Zobel, and K. E. Holt, "Bandage: interactive visualization of de novo genome assemblies," *Bioinformatics*, vol. 31, no. 20, pp. 3350–3352, Oct. 2015.

[157] Y. W. Wu, B. A. Simmons, and S. W. Singer, "MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets," *Bioinformatics*, vol. 32, no. 4, pp. 605–607, Feb. 2016.

[158] D. H. Parks, M. Imelfort, C. T. Skennerton, P. Hugenholtz, and G. W. Tyson, "CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes," *Genome Res.*, vol. 25, no. 7, pp. 1043–1055, Jul. 2015.

[159] A. Mikheenko, V. Saveliev, and A. Gurevich, "MetaQUAST: evaluation of metagenome assemblies," *Bioinformatics*, vol. 32, no. 7, pp. 1088–1090, Apr. 2016.

[160] T. Seemann, "Prokka: rapid prokaryotic genome annotation," *Bioinformatics*, vol. 30, no. 14, pp. 2068–2069, Jul. 2014.

[161] S. S. Abby and E. P. C. Rocha, "Identification of Protein Secretion Systems in Bacterial Genomes Using MacSyFinder," *Methods Mol. Biol.*, vol. 1615, pp. 1–21, 2017.

[162] M. Kanehisa, Y. Sato, and K. Morishima, "BlastKOALA and GhostKOALA: KEGG Tools for Functional Characterization of Genome and Metagenome Sequences," *J. Mol. Biol.*, vol. 428, no. 4, pp. 726–731, Feb. 2016.

[163] M. Kanehisa and Y. Sato, "KEGG Mapper for inferring cellular functions from protein sequences," *Protein Sci.*, vol. 29, no. 1, pp. 28–35, Jan. 2020.

[164] R. L. Tatusov *et al.*, "The COG database: An updated vesion includes eukaryotes," *BMC Bioinformatics*, vol. 4, no. 1, pp. 1–14, Sep. 2003.

[165] A. Subramanian, H. Kuehn, J. Gould, P. Tamayo, and J. P. Mesirov, "GSEA-P: a desktop application for Gene Set Enrichment Analysis," *Bioinformatics*, vol. 23, no. 23, pp. 3251–3253, Dec. 2007.

[166] L. Xu *et al.*, "OrthoVenn2: a web server for whole-genome comparison and annotation of orthologous clusters across multiple species," *Nucleic Acids Res.*, vol. 47, no. W1, pp. W52–W58, Jul. 2019.

[167] P. A. Chaumeil, A. J. Mussig, P. Hugenholtz, and D. H. Parks, "GTDB-Tk: a toolkit to classify genomes with the Genome Taxonomy Database," *Bioinformatics*, vol. 36, no. 6, pp. 1925–1927, Mar. 2020.

[168] M. Krzywinski *et al.*, "Circos: an information aesthetic for comparative genomics," *Genome Res.*, vol. 19, no. 9, pp. 1639–1645, Sep. 2009.

[169] D. M. Emms and S. Kelly, "OrthoFinder: Phylogenetic orthology inference for comparative genomics," *Genome Biol.*, vol. 20, no. 1, p. 238, Nov. 2019.

[170] Y. Li, K. M. Kocot, C. Schander, S. R. Santos, D. J. Thornhill, and K. M. Halanych, "Mitogenomics reveals phylogeny and repeated motifs in control regions of the deep-sea family Siboglinidae (Annelida)," *Mol. Phylogenet. Evol.*, vol. 85, pp. 221–229, Apr. 2015.

[171] Y. Sun *et al.*, "The mitochondrial genome of the deep-sea tubeworm Paraescarpia echinospica (Siboglinidae, Annelida) and its phylogenetic implications," *Mitochondrial DNA. Part B, Resour.*, vol. 3, no. 1, pp. 131–132, Jan. 2018.

[172] Y. Lan *et al.*, "Hologenome analysis reveals dual symbiosis in the deep-sea hydrothermal vent snail Gigantopelta aegis," *Nat. Commun.*, vol. 12, no. 1, Dec. 2021.

[173] J. M. Martín-Durán *et al.*, "Conservative route to genome compaction in a miniature annelid," *Nat. Ecol. Evol. 2020 52*, vol. 5, no. 2, pp. 231–242, Nov. 2020.

[174] W. N. Venables and B. D. Ripley, "Modern Applied Statistics with S," 2002.

[175] A. Alexa and J. Rahnenfuhrer, “Bioconductor - topGO,” 2022. [Online]. Available: https://bioconductor.org/packages/release/bioc/html/topGO.html. [Accessed: 10-May-2022].

[176] R. Caspi *et al.*, “The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases,” *Nucleic Acids Res.*, vol. 42, no. D1, pp. D459–D471, Jan. 2014.

[177] F. M. Martín-Zamora *et al.*, “Annelid functional genomics reveal the origins of bilaterian life cycles,” *Nature*, vol. 615, no. 7950, Jan. 2023.

[178] “CRAN - Package pheatmap.” [Online]. Available: https://cran.r-project.org/web/packages/pheatmap/index.html. [Accessed: 13-Jul-2022].

[179] T. Aramaki *et al.*, “KofamKOALA: KEGG Ortholog assignment based on profile HMM and adaptive score threshold,” *Bioinformatics*, vol. 36, no. 7, pp. 2251–2252, Apr. 2020.

[180] N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter, “Near-optimal probabilistic RNA-seq quantification,” *Nat. Biotechnol.*, vol. 34, no. 5, pp. 525–527, May 2016.

[181] K. Katoh and D. M. Standley, “MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability,” *Mol. Biol. Evol.*, vol. 30, no. 4, p. 772, Apr. 2013.

[182] O. Seudre, A. M. Carrillo-Baltodano, Y. Liang, and J. M. Martín-Durán, “ERK1/2 is an ancestral organising signal in spiral cleavage,” *Nat. Commun. 2022 131*, vol. 13, no. 1, pp. 1–14, Apr. 2022.

[183] A. M. Waterhouse, J. B. Procter, D. M. A. Martin, M. Clamp, and G. J. Barton, “Jalview Version 2—a multiple sequence alignment editor and analysis workbench,” *Bioinformatics*, vol. 25, no. 9, pp. 1189–1191, May 2009.

[184] S. Capella-Gutiérrez, J. M. Silla-Martínez, and T. Gabaldón, “trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses,” *Bioinformatics*,

vol. 25, no. 15, p. 1972, Aug. 2009.

[185] M. N. Price, P. S. Dehal, and A. P. Arkin, “FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments,” *PLoS One*, vol. 5, no. 3, p. e9490, Mar. 2010.

[186] L. T. Nguyen, H. A. Schmidt, A. Von Haeseler, and B. Q. Minh, “IQ-TREE: A Fast and Effective Stochastic Algorithm for Estimating Maximum-Likelihood Phylogenies,” *Mol. Biol. Evol.*, vol. 32, no. 1, p. 268, Jan. 2015.

[187] E. Rambaut, A. (Institute of Evolutionary Biology, University of Edinburgh, “FigTree,” 2010. .

[188] J. P. Huelsenbeck and F. Ronquist, “MRBAYES: Bayesian inference of phylogenetic trees,” *Bioinformatics*, vol. 17, no. 8, pp. 754–755, Aug. 2001.

[189] A. Marchler-Bauer and S. H. Bryant, “CD-Search: protein domain annotations on the fly,” *Nucleic Acids Res.*, vol. 32, no. Web Server issue, p. W327, Jul. 2004.

[190] S. Lu *et al.*, “CDD/SPARCLE: the conserved domain database in 2020,” *Nucleic Acids Res.*, vol. 48, no. D1, pp. D265–D268, Jan. 2020.

[191] C. Chen *et al.*, “TBtools: An Integrative Toolkit Developed for Interactive Analyses of Big Biological Data,” *Mol. Plant*, vol. 13, no. 8, pp. 1194–1202, Aug. 2020.

[192] D. Thiel, L. A. Yañez-Guerra, M. Franz-Wachtel, A. Hejnol, and G. Jékely, “Nemertean, Brachiopod, and Phoronid Neuropeptidomics Reveals Ancestral Spiralian Signaling Systems,” *Mol. Biol. Evol.*, vol. 38, no. 11, pp. 4847–4866, Oct. 2021.

[193] S. R. Eddy, “Accelerated Profile HMM Searches,” *PLOS Comput. Biol.*, vol. 7, no. 10, p. e1002195, Oct. 2011.

[194] “Pfam: Home page.” [Online]. Available: https://pfam.xfam.org/. [Accessed: 13-Sep-2022].

[195] T. Frickey and A. Lupas, “CLANS: a Java application for visualizing protein families based on pairwise similarity,” *Bioinformatics*, vol. 20, no. 18, pp. 3702–3704, Dec.

2004.

[196] “CLANS | Bioinformatics Toolkit.” [Online]. Available: https://toolkit.tuebingen.mpg.de/tools/clans. [Accessed: 13-Sep-2022].