UPPSALA
UNIVERSITET

UPTEC X 19045
Examensarbete 30 hp
December 2019

Bioinformatic approaches for detecting homologous genes in the genomes of non-model organisms

A case study of wing development genes in insect genomes

Lauri Mesilaakso

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Bioinformatic approaches for detecting homologous genes in the genomes of non-model organisms

Lauri Mesilaakso

Identifying homologous genes, that is genes from a common ancestor, is important in comparative genomic studies for understanding gene annotation and the predicted function of a gene. Several pieces of software, of which the most well-known is BLAST, have been developed for identifying homologues, but this can be challenging in non-model organisms where sometimes poor quality of genome assemblies and lack of annotation make it difficult to robustly identify homologues. The aim of this project was to build a bioinformatic framework for homology detection using genomes from non-model organisms. The approach developed used genome annotations, annotated polypeptide sequences and genome assembly sequences to detect homologous genes. The framework was applied to identify Drosophila melanogaster homologous wing development genes in the genomes of nine other insect species with the aim to understand the evolution of loss of wings. To identify changes related to wing loss, the homologous protein sequences obtained were aligned and phylogenetic trees were built from them. The aim of creating the multiple protein alignments and phylogenetic trees was to shed light on whether changes in gene sequences can be related to presence or absence of wings. From the set of 21 candidate wing development genes identified with literature and subsequent database searches, I tested eight and was successful in identifying homologues for all of them in eight of the 10 insect genomes. This was done using a combination of text searches in genome annotations, searches with Exonerate v. 2.4.0 alignment program in annotated polypeptide sequences and in genome assemblies. The eight genes chosen for testing the framework were based on initial finding of putative homologues in the eight insect genomes when using the first two steps of the framework. For the set of homologous wing development genes examined I was not able to identify any conclusive pattern of potential protein coding changes that correlated with loss of wings in these species. Improvement to the current pipeline could include using query sequences from closer relatives of the 8 test species than D. melanogaster and, of course, testing of the remaining wing development genes as well as further literature study of wing development genes. Together these could improve future studies on the evolution of wing loss in insects.

Handledare: Arild Husby, Erik Gudmunds och Douglas Scofield
Ämnesgranskare: Sandra Baldauf
Examinator: Jan Andersson
ISSN: 1401-2138, UPTEC X 19045

En metod för att hitta och studera gener i insekters arvsmaterial

Vi lever i genomeran. Nya tekniker för att undersöka hela biosfärens arvsmaterial har medfört en till synes obegränsad rymd av information att upptäcka. Många insektsarter har studerats med dessa tekniker, vissa mera noggrant än andra. En viktig del i undersökningen av detta material är att försöka förstå hur arvmaterial skapar funktionalitet i organismer där en viktig funktionalitetsskapare kallas gen. Gener är uppbyggda genom att tvinna ihop relativt enkla beståndsdelar nukleotider till långa kedjor, vilket bildar DNA. Det finns fyra olika sorters nukleotider och beroende på ordningen eller den så kallade ”sekvensen” av dessa i en gen kan genen få olika funktioner.

När arvsmaterialet av en organism börjar undersökas, är en väsentlig deluppgift i det hela att hitta gener. Ett sätt att genomföra detta är att jämföra gener kända från andra organismer till den nya organismens arvsmaterial. När det finns likheter som sannolikt inte skulle kunna uppstå av en slump i två organismer, oberoende av varandra, kan man anta ett evolutionärt gemensamt ursprung till likheten.

Målet i det här examensprojektet är tudelat. Det första målet är att presentera ett sätt att hitta likheter i gensekvenser när det studerade arvsmaterialet är av mycket varierande kvalité. I vissa fall kan likhetssökande till stor del baseras på det som redan har etablerats och publicerats i vetenskapsvärlden medan i andra fall är det enda man har en gensekvens och stor mängd av mer eller mindre osammanhängande arvsmaterial att söka igenom. Det andra målet är att försöka se om de hittade skillnaderna i liknande gensekvenserna kan vara kopplat till skillnader i vissa funktionaliteter i några särskilda insekter.

Det huvudsakliga sättet som projektet genomförts på börjar med att först titta igenom det som redan har publicerats. Ifall sökandet i det som publicerats inte hittar något fortsätter man till nästa steg. I det här steget söks först en mindre datamängd av arvsmaterial som är en direkt produkt av gener och om fortfarande inget hittas, söks sist i hela arvmaterialet av organismen.

Detta sätt att undersöka geners och dess sekvensernas likheter för att hitta gener i nya organismer och upptäcka kopplingar mellan funktionaliteter och sekvenser kan mycket lämpligt tillämpas för att förstå evolutionär utveckling av arvsmassan kopplat till olika organismers egenskaper.

Table of contents

Abbreviations

BLAST	Basic Local Alignment Search Tool
MPA	Multiple Protein Alignment

1 Introduction

Homology is defined as common ancestry and homologous genes are genes that share a common ancestral sequence. There are three different types of homology, orthology, paralogy and xenology, which are defined as homology deriving either from speciation, duplication or horizontal gene transfer event respectively (Koonin 2005, Li 2006). Sequences, amino acid or nucleotide, can be identified to be homologous by detecting statistically significant similarity between them – when two sequences share more than is expected by chance. The most parsimonious explanation of this excess similarity is then that the two sequences did not arise independently but share a common ancestor (Pearson 2013).

Sequences have different evolutionary distances which, together with the goal of the study, should be considered when choosing whether to use DNA or translated DNA sequences. DNA:DNA alignments seldom detect homology when the time of divergence is more than 200–400 million years whereas protein:protein (or translated DNA) alignments are much more sensitive. This is so because the functionality of proteins, determined by the protein sequence, is under selective pressure whereas for coding sequences, the translation of codons is degenerate for most of the amino acids and DNA alignments can thus be noisy. In addition, statistics for DNA:DNA alignments are often less accurate than the ones for protein:protein alignments (Pearson 2013). This is due to typically smaller sizes of protein databases in comparison to DNA databases.

A multitude of software has been developed for identifying similarities between sequences. Some of the widely used similarity searching programs are BLAST (Altschul *et al.* 1997) and exonerate (Slater & Birney 2005) both of which were extensively used in this project.

BLAST is a common name for a family of database search programs. They can use DNA or protein sequences both as queries and as databases to search in. BLAST implements variations of its core algorithm in its different search programs but what is common is that it utilises a heuristic method for fast discovery of local alignments with alignment scores that surpass a certain statistical significance threshold (Altschul 2014).

The BLAST algorithm for both DNA and protein query and subject sequences, follows a two-step procedure. First near-perfect matches between subject and words in the query sequence are sought. Words are continuous sequences of length k, k-mers, which are formed from the query. When near-perfect matches are found, the matches are then extended both upstream and downstream and are checked if they belong within longer, high-scoring segment pairs, HSPs. HSPs are gapped or ungapped aligned pairs of sequences which have a maximal aggregate score that cannot be improved and of which the score exceeds certain threshold value (Altschul *et al.* 1990). BLAST is an approximation of the Smith-Waterman local alignment algorithm which is a rigorous dynamic programming method for discovery of optimal local alignments. BLAST trades off the chance of not discovering weak sequence similarities to substantially increased speed in searching alignments (Altschul 2014).

The alignments calculated by BLAST are so called local alignments. They identify the most similar regions between two sequences and ignore the rest (Pearson 2005). For single domain proteins, the ends of alignment may coincide with the ends of the query protein. However, this is not the case e.g. for domains located in different sequence contexts in different proteins, BLAST can score high enough only on homologous domains (Pearson 2013). This is a challenge for BLAST when the goal is to capture as much of the homologous protein as possible. Exonerate, on the other hand, can utilise a variety of different alignment models, among others, alignment of proteins to genomes which allows introns as well as frameshifts in the alignment, and changes of exon phases when a codon is split by an intron (Slater & Birney 2005). This feature renders it superior to BLAST when alignment of whole proteins is essential.

As mentioned so far, there are algorithmic and computational challenges to finding homologs. Additionally, the input data, genome assemblies and annotations, can be of varying quality. Errors can happen at any level, assembly or annotation. Faulty assemblies can misguide sequence similarity searches. Such errors can be for instance inserted foreign DNA or just misplaced contigs (see Figure 1 A) or assemblies can be too fragmented and only partial alignments are successful (see Figure 1 B).

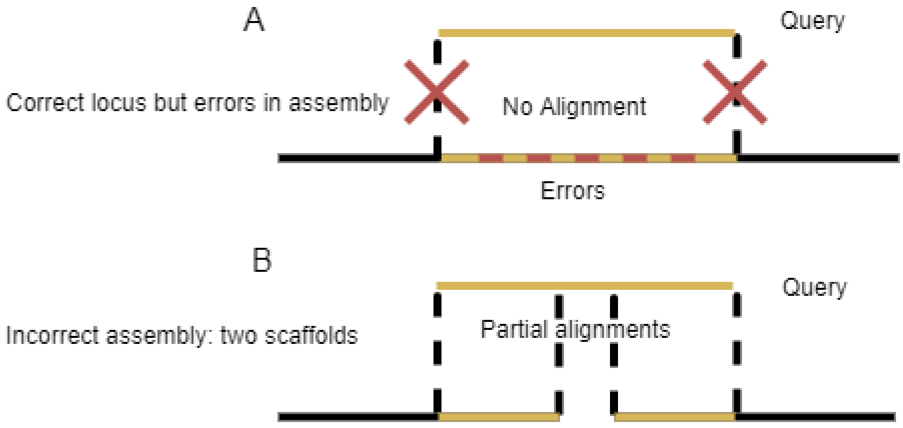

A
Query
Correct locus but errors in assembly
No Alignment
Errors
B
Query
Incorrect assembly: two scaffolds
Partial alignments

Figure 1. Two ways assembly errors can interfere with protein sequence retrieval. (A) a query can fail to align to subject sequence due to errors in the assembly. (B) query only aligns partially to the subject due to fragmented scaffolds.

On the other hand, annotations can also contain errors. Some of these errors can be caused by e.g. poor annotations that are generated by automated pipelines and lack functional information. Deficiencies in annotations can also hinder the finding of homologs through searching for names. Another consequence of poor annotations can also be that annotations have not found all translated coding sequences and searching in them cannot find the homologous proteins of interest.

The assignment of function is a central task in functional annotation of newly assembled genomes. In general, we are interested in identifying homologous proteins because homologous proteins tend to have similar functions, active sites or binding domains

(Thompson & Poch 2006). They share also significant three-dimensional structural similarity (Pearson 2005) which can further elucidate the function of the protein.

Once homologous proteins have been found in the species of interest, they all can be aligned to each other. This is called multiple protein alignment (MPA). MPAs provide an overall view of a family of proteins and are useful in identifying similar conserved patterns. MPAs show how a set of proteins may be related by identifying and arranging in columns similar, and therefore presumably homologous, residues, which tend to be structurally and functionally equivalent. The variation in aligned sequences, can be postulated to have arisen through substitution and insertion-deletions (‘indels’) events. E.g. differing lengths of homologous sequences can be explained by indel events (Thompson & Poch 2006). The underlining ambition in this undertaking is to gain a deepened understanding of the potential functional variation in the proteins involved in wing development in dipteran insects.

Phylogenetic trees are useful in visualising differences between sequences and while MPAs can give a detailed view of differences between each residue in each sequence, trees can indicate the accumulated differences between sequences which causes similar sequences to cluster together and therefore be associated with potential trait differences.

The goals of this project were twofold. The first goal is to develop a bioinformatic approach to search fragmented and often poorly annotated non-model organism genomes for homologs. The second goal is to apply this method to detect homologous wing development genes from *D. melanogaster* in other published insect genomes to provide potential insights into the loss of wings in some insect species.

The queries for the homologue search were found by searching in the literature for genes known to be involved in wing development. Earlier work on pea aphids *Acyrthosiphon pisum* (see gene names 1-11 in Table A1 in Appendix A) (Vellichirammal *et al.* 2017), the fruit fly *Drosophila melanogaster* (see genes 12-20 in Table A1 in Appendix A) (Abouheif 2002) and the brown planthopper *Nilaparvata lugens* (see gene name 21 in Table A1 in Appendix A) (Xu *et al.* 2015) have functionally characterised genes involved in wing development. In order to have well established query sequences for all genes, the genes discovered in non-drosophila species were searched by name in FlyBase (Thurmond *et al.* 2019), a database focused on *Drosophila* genes and genomes and gene sequences with the same names found in *Drosophila melanogaster* were selected for further use.

One of the main challenges that the pipeline developed for homology search was attempting to address is the highly varied quality of resources available. Several of the species selected are so called non-model organisms, meaning they are not species studied for a long time in the lab environment where both genomic resources and functional information is well developed. For example, both *M. extradentata*, *T. cristinae* and *C. hookeri* do not have functional annotation only structural. This is very different to the model organisms such as *Drosophila* (of which two species were included here *D. melanogaster* and *D. simulans*) which have been studied for well over hundred year and have entire databases devoted just for their annotation, see FlyBase (Thurmond *et al.* 2019). This general division between model vs non model organism at the available functional annotation also applies to the quality of the genome assemblies. For example, *D. melanogaster* assembly has 2 442 contigs with N50 of 21 485 538 bp whereas *T. cristinae* has 207 031 contigs with N50 of 8 919 bp. Contig N50 is defined as the length of contig, of which equal or longer sized contigs cover at least half of the genome sequence.

The goal was to collect a varied and broad enough set of species in order to have enough candidate homologous protein sequences from each category of wing morphology (winged vs wingless) for subsequent analyses of potential common patterns at the genetic level that are related to the observed phenotypic differences between the groups. Thus, four species with monomorphic apterous (i.e. wingless, see species 1-4 in Table 1), two species with monomorphic macropterous (i.e. winged, see species 5 and 6 in Table 1) were chosen. In addition, we also included four species that produce wings but that are polyphenic, meaning they can produce both short or long wings (see species 7-10 in Table 1).

One application area where comparative genomics approaches with homology search described above can bring interesting insights is the study of loss of wings in insects (Roff 1990). While most insects have wings, loss of wings have happened repeatedly in different lineages. One of the most likely reasons for this is that wing development is costly and trades off with other traits such as fecundity (Roff 1990). How this loss occur on the genetic level is however not clear. By searching for homologous genes in the genomes of different species of insects that are wingless compared to winged the aim was to see if there are certain anomalies in known wing development genes that can produce the loss of wings (e.g. a loss of function mutation in the same gene). As part of the group of insects that produce wings, we also included some species that are polyphenic meaning that one genotype has the ability to produce more than one phenotype when exposed to different environments (Kelly *et al.* 2012). The capability to adjust phenotype to environment is called phenotypic plasticity and is of adaptive importance to many insect species (Simpson *et al.* 2011). This is the reason why four polyphenic species were included in this study as well. Figure 2 gives examples of some of the species chosen for this study.

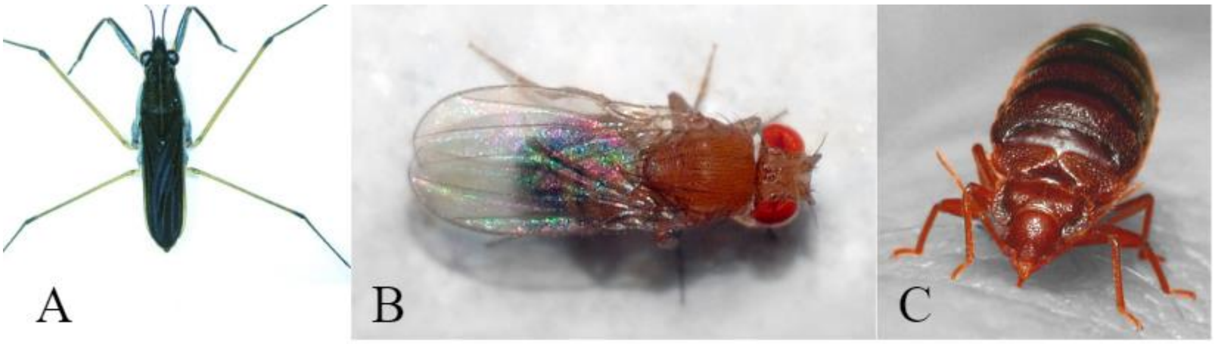


Figure 2. Illustration of some species chosen for this study. A Long-winged Gerris buenoi adult individual

(polyphenic), B Drosophila melanogaster (macropterous) (André Karwath aka Aka) and C Cimex lectularius (apterous) (AJC1 from UK).

2 Materials and methods

2.1 Genes and species of interest

For homology search of wing development genes in *Gerris buenoi*, a set of candidate genes were chosen from *Drosophila melanogaster* due to high quality of functional annotation of genes involved in wing development. Table A1 in Appendix A lists out the wing development genes of interest found in literature (Abouheif 2002, Xu *et al.* 2015, Vellichirammal *et al.* 2017) and of which the lexicographically first translated isoforms were used as query sequences for searching homologous sequences in other species. By lexicographically first translated isoforms is meant that if there were several annotated translated isoforms such as A, B and C. Translated isoform A was chosen because there is always at least one isoform (named as A) and picking the first one is simplest. Furthermore, the evolutionary distances are likely to be of lesser importance between isoforms in comparison to the putative homologues in the species of interest and therefore picking any one of the isoforms would likely be equally suitable.

Ten species with available genomes and differences in wing morphologies were chosen. They are listed n Table 1.

Table 1. The species used in this study and their wing morphology.

No	Species	Wing morphology
1	*C. hookeri* (smooth stick-insect)	Monomorphic apterous (wingless)
2	*C. lectularius* (bed bug)	Monomorphic apterous
3	*M. extradentata* (Vietnamese walking stick)	Monomorphic apterous
4	*T. cristinae* (Walking stick)	Monomorphic apterous
5	*D. melanogaster* (fruit fly)	Monomorphic macropterous (winged)
6	*D. simulans* (fruit fly)	Monomorphic macropterous
7	*A. pisum* (pea aphid)	Polyphenic (winged)
8	*F. exsecta* (narrow-headed ant)	Polyphenic

9	*G. buenoi* (water strider)	Polyphenic
10	*N. lugens* (brown planthopper)	Polyphenic

2.2 Genome, annotation and polypeptide data

For identifying homologous *Drosophila melanogaster* protein sequences in other insect genomes, a number of different data was used (Table B1 in Appendix B). The data described in this table was retrieved at different stages of the project and will be discussed and referenced in more detail in the section where those pieces of data were used in the developed pipeline.

2.3 Homology assessment

The search for homologous protein sequences to *D. melanogaster* genes in species listed in Table 1 used primarily two approaches, BLAST and exonerate. In the following the reason for opting for the latter approach is discussed followed by detailed description of the pipeline using this approach.

2.3.1 Homology search

It is not a coincidence that BLAST with its various database search programs has become such a household name in homology searching. For many bioinformatic purposes its tradeoff of sensitivity for weak sequence similarities for speed is justified. The challenge comes in that BLAST does not have any model for dealing with introns. The HSPs that BLAST returns can be part of an exon, contain many exons, simply be just noise, or a whole exon is not found by any HSP at all. There is no ready way that BLAST can by itself overcame these obstacles. BLAST, in its core, is a local aligner finding the best alignment of two sequences and ignoring the rest. If the goal is to find the full homologous sequence in the subject sequence, building gene models based on BLAST results is difficult. One solution for overcoming this weakness of not finding the whole proteins, can be attempted with tiling but the difficulties of e.g. determining where the aligned proteins start and end can sometimes become too challenging.

Therefore, exonerate became the main tool of choice for finding homologous protein sequences mainly due to its capability to incorporate splicing in the complete alignment model it uses. The exonerate approach, consisted of steps following each other with each of the steps giving some more evidence for homology of the putative polypeptide sequences (see below). If there were satisfactory matches found in an earlier step, the search was not continued to the next step. The whole pipeline detailing this approach is available on Github (Mesilaakso 2019). The three steps in this approach are described in more detail below.

2.3.2 Find name matches in annotations

The first step in finding homologous translated genes listed in Table A1 in Appendix A was by executing a text search on annotation gff-files (1, 3, 6, 7, 10, 11, 13, 15, 17 and 18 listed in Table B1 in Appendix B). This gave a general overview of which genes were annotated with the same name, suggesting homology. The text searches used regular expressions with first part as “mRNA\s\w+.+” and the second and last part being the gene name. Once the name match was found, the corresponding protein was retrieved from annotated polypeptide multifasta file using an id found on the same line. The program used for the searches in annotation files was zgrep, a command line text search program for compressed files.

2.3.3 Complement with exonerate searches against protein sequences

For *C. lectularius, D. simulans, C. hookeri, G. buenoi, M. extradentata* and *T. cristinae* there were too few text search hits for several genes in order to move forward with enough candidate sequences for further analyses. In addition, for *T. cristinae* I was not able to find any annotated polypeptide sequence file, so it was completely excluded from further downstream analyses. For the other five species except *T. cristinae,* exonerate v. 2.4.0 (Slater & Birney 2005) was run in order to find potential homologs. The exonerate alignments used the same *D. melanogaster* protein sequences as mentioned before as query sequences and annotated multifasta polypeptide files (2, 5, 9, 12 and 14 in Table B1 in Appendix B) as target sequences. The matches with highest raw scores, query coverages and alignment lengths were chosen and the full protein sequences where these best matches were found, were chosen as the putative homologous protein sequences. By picking as candidates the full annotated protein sequences which were found as best matches by exonerate thus utilised the protein models already established for the five species.

Searches with exonerate can be adjusted with various flags to suit the particular search task at hand. The flags used for the searches above defined the alignment model to be affine local. This alignment model allows alignments which can overlap each other both in the aligned query and target sequence and it is similar to the classic Smith-Waterman-Gotoh type of alignment. As Smith-Waterman algorithm though known to produce optimal local alignments between two sequences, has a quadratic time complexity, the Gotoh’s approximation to it reduces it from mn(m+n) to mn, where m and n are the lengths of the two sequences (Mott 2005). Other flags to be used were the choice of protein substitution matrix which was PAM250 and refine full flag which forces exonerate to exhaustively refine alignments of the pair of sequences by using dynamic programming over larger regions.

2.3.4 Complement with exonerate searches against genome sequences

After using exonerate against annotated polypeptide sequences of certain genes, *C. hookeri* and *M. extradentata* completely lacked matches which seemed to fit with the other protein sequences when they were aligned in multiple protein alignments. In those cases, another search with exonerate was executed using the same *D. melanogaster* protein sequences against the genome assemblies (4 and 16 Table B1 in Appendix B). This additional search yielded matches which aligned significantly better to the other protein sequences (i.e. protein annotations in these two genomes were insufficient with respect to identifying these homologues).

Exonerate search in this step used the same protein substitution matrix and refine full flag as previous search in annotated polypeptides but used another alignment model. The model used was “protein to genome” model which allows incorporation of gaps and frameshifts as well as modelling of introns and intron phases.

2.4 Multiple protein alignments

2.4.1 Create multiple protein alignments

Multiple protein alignments (MPAs) of putative homologous protein sequences were created using homologous protein sequences found in the three homology search steps described before. Out of the 21 wing development genes, eight were found to have putative homologs in all species listed in Table 1 except in *T. cristinae.* Due to reasons discussed later, the homology search detailed in 2.3 Putative homology search, was not fully applied to the other 13 genes of interest and therefore no putative homologs were included from them into MPAs.

In Table D1 in Appendix D are detailed the sources of where the eight genes were found. Notable is that for *D. melanogaster* and *D. simulans* certain protein sequences were retrieved directly from FlyBase (Thurmond *et al.* 2019) because the regular expression searches in the annotation gff-files did not return matches, which could have been expected to have been found based on that all the eight genes most likely should have homologs in other *Drosophila* flies than just *D. melanogaster*.

The heuristic used for choosing putative homologues from exonerate matches against annotated polypeptide sequences included surveying visualisations of the 10 best exonerate matches with highest raw scores and picking among those the 1-5 best hits with respect to raw scores and query coverages. If there were two or more matches with similar values in raw scores and query coverages clustered together, only one was picked out as these matches might be due to being different translated isoforms of the same gene.

Once the putative homologues were chosen, the full protein sequences where the matches were found, were gathered into multifasta files and MPAs of them were executed with MAFFT v 7.407 (Katoh & Standley 2013). Through this heuristic for certain genes more than one putative homologous protein sequences were retained in the MPAs. One possible explanation for this can be that these can be paralogs in the species.

2.4.2 Multiple protein alignment evaluation and refinement

After MPA was executed for putative homologous protein sequences for the nine species, the MPA was evaluated by eye to see if there were sequences which did not align well with other sequences. If there were such sequences they were manually removed from the alignment and the rest of the sequences were aligned with MAFFT again. This manual curation of removal and re-alignment was repeated until no divergent sequences were left in the alignment. If no putative homologous sequences were left for a species (as was the case for *C. hookeri* regarding Ultrabithorax and engrailed genes and for *M. extradentata* regarding Eip74EF and Ultrabithorax genes), the searching with exonerate against genome assemblies of these two species were executed (as described in section 2.3.4).

During this stage was also discovered that putatively homological *G. buenoi* Ecdysone receptor protein sequence was annotated in two pieces and they were therefore joined together for the final alignment as one.

2.5 Create phylogenetic trees of final alignments

Lastly, the final MPAs with likeliest homologous polypeptide sequences of each nine species were used to create approximately-maximum-likelihood phylogenetic trees using FastTree v. 2.1.10 (Price *et al.* 2010). The purpose of these trees was to visualise the total similarity or unsimilarity of sequences from each species in order to see if species with long and polyphenic wing morphology by themselves and species with short wing morphology by themselves would cluster together.

2.6 Practices applied for greater reproducibility

In parallel with the development of homology detection pipeline and obtaining of initial results of the application of the pipeline, certain practices for higher reproducibility were adopted as well. They are shortly discussed in the following.

2.6.1 Version control with Git and Github

In a computational biology project, keeping track of various of versions of files can be very helpful, in e.g. either reverting to previous versions of portions of files or restoring completely earlier versions of them. Git is a software developed for this purpose. Git integrates smoothly with many code repository services such as GitLab, Bitbucket or GitHub. Code repository services allow among others storing and sharing of files as well as ease of collaboration in a common computational project.

All through this current project both git and Github were extensively used. Initially, a private repository was created which contained all code and most of the input data as well as intermediary and final results except files with sizes more than 50 Mb, and lastly a public repository was also created. It contained the final developed pipeline with all necessary input files as well as all intermediary files and end results. Large genome files were though excluded from this repository due to their size (Mesilaakso 2019).

2.6.2 Computational notebooks

Throughout this project R Markdown computational notebooks implemented with R package knitr (Xie 2014) were used inside R studio IDE (RStudio Team 2019). R Markdown computational notebooks allow to combine code, rendered output (e.g. figures or tables) and written analysis in markdown syntax into single documents which can be rendered with ease to various formats such as HTML documents or PDFs. R Markdown files are plain text and thus can be readily version controlled. The main R Markdown document used in showcasing the developed pipeline can be found in the project's Github page (Mesilaakso 2019).

2.6.3 Docker process virtualisation

A common challenge in computational biology has long been the successful set-up of development environment with the bioinformatic tools required for computational experiments at hand. The same challenge is also met by those attempting to replicate the same experiments.

Docker is a virtualisation software which can alleviate this problem (Merkel 2014). A set of four Dockerfiles with certain additional configuration files were created which can be used to build images used for preparing and creating the final image. This final image is the blueprint from which can be created containers which contain the exact development environment used in the project. All the Dockerfiles including the configuration files are available in the project's Github repository (Mesilaakso 2019).

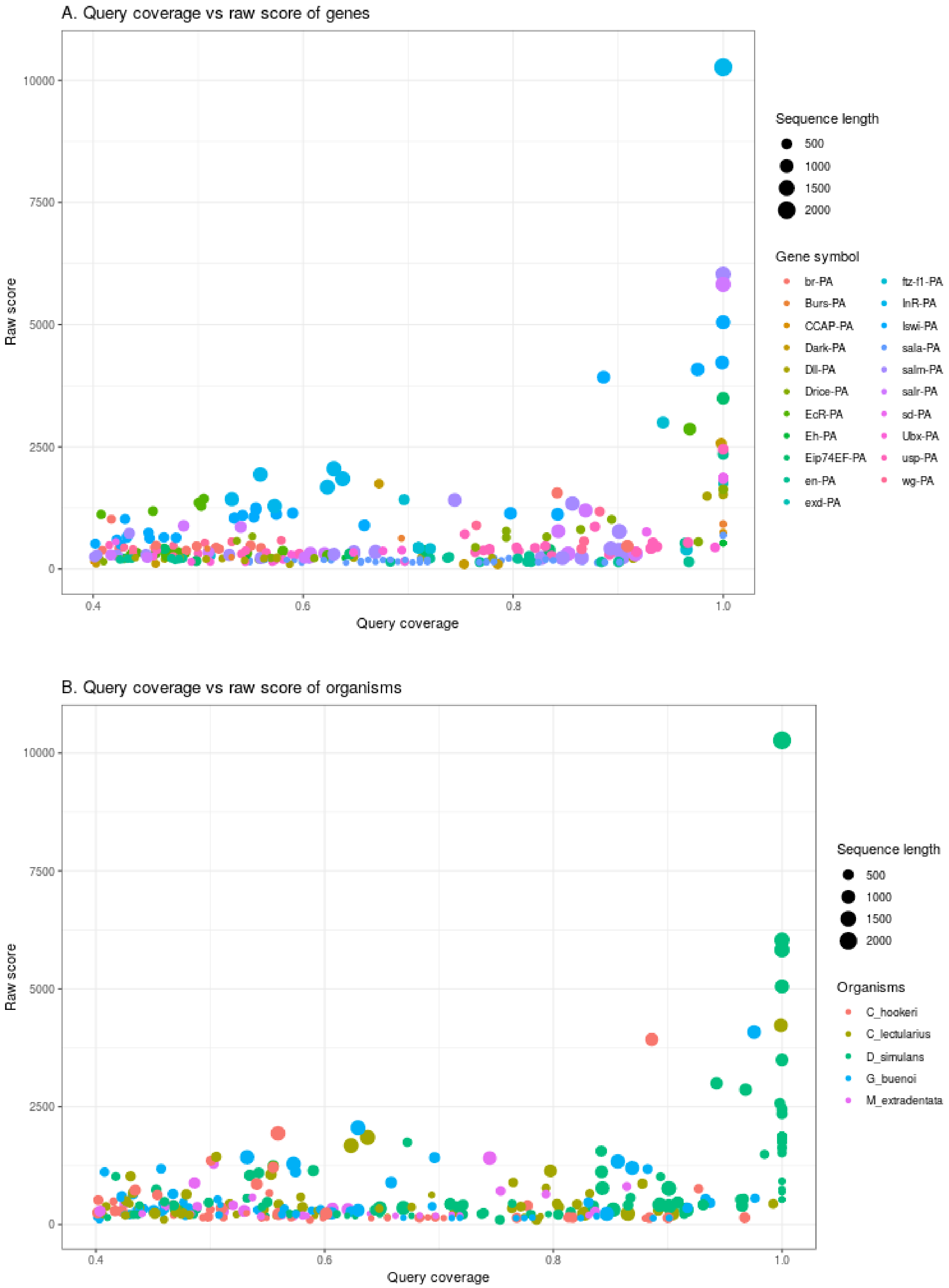
3 Results

3.1 Name matches in annotations

For name searching in the first step of putative homology detection, several matches were found in all but the following species: *C. hookeri*, *M. extradentata* and *T. cristinae*. They either lacked annotation altogether (as was the case for *T. cristinae*) or had annotations which didn't contain information about which genes were found in genomic positions. In Table C1 in Appendix C is summarised the search results of names of genes in gff-annotation files with zgrep.

3.2 Matches of exonerate searches in protein sequences

Exonerate searches against the five polypeptide multifasta files resulted in a great number of matches for all genes. Figure 3 illustrates the ten best matches (with respect to raw score) found for all the genes and in all five species' annotated polypeptide sequences. These ten best matches were also used as one of the sources among which the candidates for MPAs were chosen from. The number of best matches was restricted to ten completely arbitrarily. However, the restriction seemed to be justified because among the ten best matches, the chosen candidates seemed to "fit" with the other candidates in subsequent MPAs apart from the cases discussed in the next section.


A. Query coverage vs raw score of genes

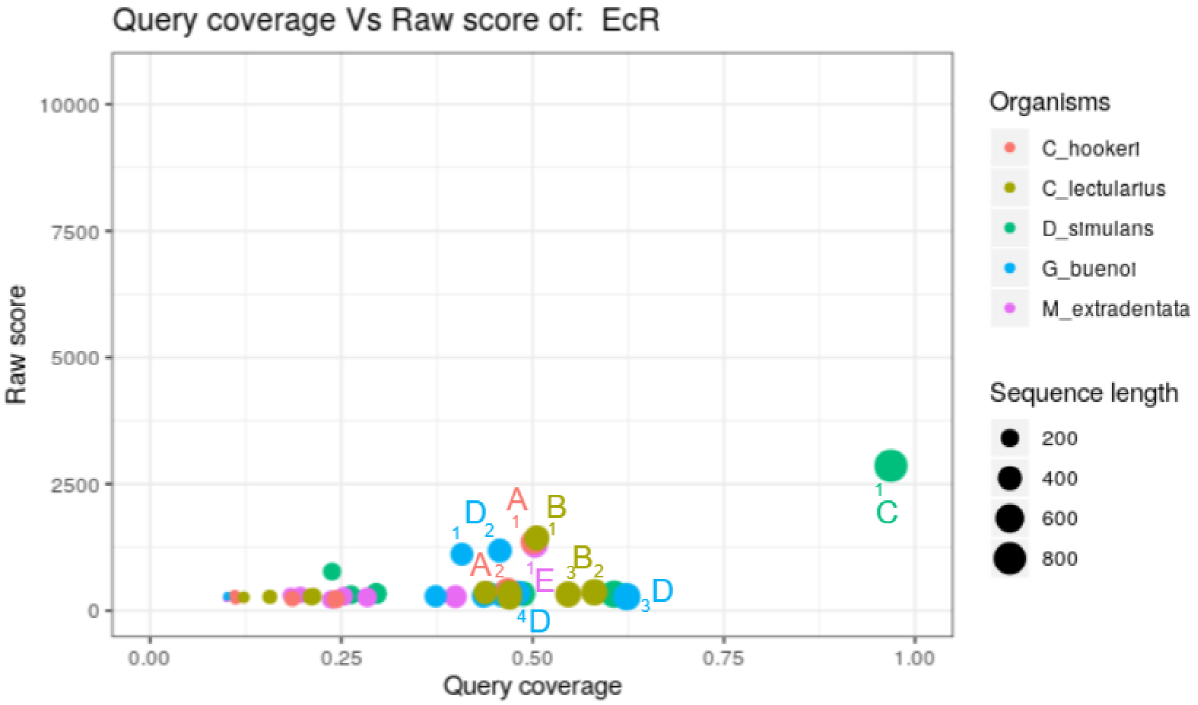
B. Query coverage vs raw score of organisms

Source: Exonerate

Figure 3. Ten best matches with respect to raw score of exonerate searches against *C. lectularius, D. simulans, C. hookeri, G. buenoi* and *M. extradentata* annotated polypeptide multifasta sequences. The horizontal axis is query

coverage which is calculated as alignment length in the target sequence over corresponding *Drosophila melanogaster* protein query sequence length. Vertical axis is the raw score of the exonerate match which is the sum of transition scores (i.e. substitution matrix scores and the gap penalties) used in the dynamic programming. The size of each dot indicates the length of the alignment in the target sequence. Upper figure A has colour coding according to which genes were matched whereas lower figure B's colour coding is according to which organisms the same matches were.

Figure 4 illustrates one example of exonerate matches, more specifically the Ecdysone receptor protein sequence search results. In it is also included which sequences were selected into the first multiple protein alignment before any manual curation of sequences. In addition, the figure exhibits one example of the heuristic used in picking out the putative homological sequences for initial MPAs. The decisions of which putative homologs were chosen as candidates based on figures such as Figure 4 showed out to be successful because the sequences in subsequent MPAs seemed to mostly "fit" with the others. Notable from the exonerate found protein sequences is that the best match found for *D. simulans* in annotated polypeptide file (marked with C1 in Figure 4) was not included in the initial MPA but rather a sequence annotated with the same name in FlyBase. Section 2.4.2 Multiple protein alignment evaluation and refinement mentions also about joining together of two translated Ecdysone receptor isoforms from *G. buenoi*. The actual protein sequences joined were alignment D1 in Figure 4 which is annotated as ”ecdysone receptor isoform A” and with a regular expression matched protein sequence annotated as ”ecdysone receptor C-term” (which it did not get high enough hit to be included in Figure 4).



Source: Exonerate

Figure 4. One example of exonerate matches with Drosophila melanogaster lexicographically first translated isoform of Ecdysone receptor gene as query sequence and annotated polypeptide multifasta sequences of five species as target sequences. The horizontal axis is query coverage which is calculated as alignment length in the target sequence over Drosophila melanogaster Ecdysone receptor protein query sequence length (which is 849 amino acids long). Vertical axis is the raw score of the exonerate match which is the sum of transition scores (i.e. substitution matrix scores and the gap penalties) used in the dynamic programming. The size of each dot indicates the length of the alignment in the

target sequence. The letters A-E with numbers varying between 1 and 4 indicate which aligned protein sequences were chosen from which species. Letters A-E represent alignments against *C. hookeri, C. lectularius, D. simulans, G. buenoi* and *M. extradentata* respectively.

3.3 Matches of exonerate searches in genome assemblies

The output of exonerate searches by default contain a raw score and length of each alignment. In addition, as the length of the query (i.e. the *D. melanogaster* protein sequence: engrailed, Ultrabithorax or Eip74EF) is known, query coverage can be easily calculated. These three, raw score, alignment length and query coverage, were used as the criteria for selecting the aligned polypeptide sequences for MPAs from matches in genome assemblies.

In order to describe the improvement in alignment of the exonerate searches between the searches against annotated polypeptide sequences and genome assemblies, average and median of alignment lengths, query coverages and raw scores are presented for exonerate runs against both annotated polypeptide sequences and genome assemblies. These results are summarised in Table 2.

Table 2. The average and median values of best results of exonerate searches using *D. melanogaster* gene sequences: engrailed, Eip74EF and ultrabithorax against *C. hookeri* (see entry 3 in Table B1 in Appendix B) and *M. extradentata* (see entry 16 in Table B1 in Appendix B) genome assemblies and polypeptides.

Query gene	Exonerate search target	Alignment length	Query coverage	Raw score
Engrailed	*C. hookeri* genome	537.7/538	0.97/0.97	370.3/320
	C. hookeri polypeptides	473/469.5	0.86/0.85	149.2/146
Eip74EF	*M. extradentata* genome	602.8/592.5	0.72/0.71	375.5/279.5
	M. extradentata polypeptides	253/230	0.31/0.28	152.7/153
Ultrabithorax	*M. extradentata* genome	372.2/371.0	0.95/0.95	348.0/354.0
	M. extradentata polypeptides	197.3/223	0.51/0.57	178.7/176
Ultrabithorax	*C. hookeri* genome	353.3/359.0	0.91/0.92	332.3/311.0
	C. hookeri polypeptides	184/133	0.40/0.34	147.1/144

3.4 Multiple protein alignments

For all the eight multiple protein alignments some manual curation was necessary. The differences in amount of manual curation depended on how many false positives the heuristic used for picking out putative homological sequences had caught. That there were false positives, for some alignments more than others (e.g. Ecdysone receptor had eight sequences which were manually filtered out) was not problematic since the manual curation and subsequent MPA were easily performed on each iteration of this procedure.

Figure E1 in Appendix E illustrates one example of multiple protein alignments, namely Ecdysone receptor, produced from the eight genes with found putative homologous protein sequences in the nine species. In the MPAs, each putative homologous sequence is also labelled by the wing morphology of the species in which the putative homology is from.

3.5 Phylogenetic trees of wing development genes

In order to visualise one example of phylogenetic trees produced from the MPAs, Figure 5 illustrates the phylogenetic tree of Ecdysone receptor putative homologous protein sequences. The trees were not rooted because we are only interested in relationships between the species, not in the directionality of evolution, i.e. we wish to determine which putative homologs are evolutionarily closer to each other.

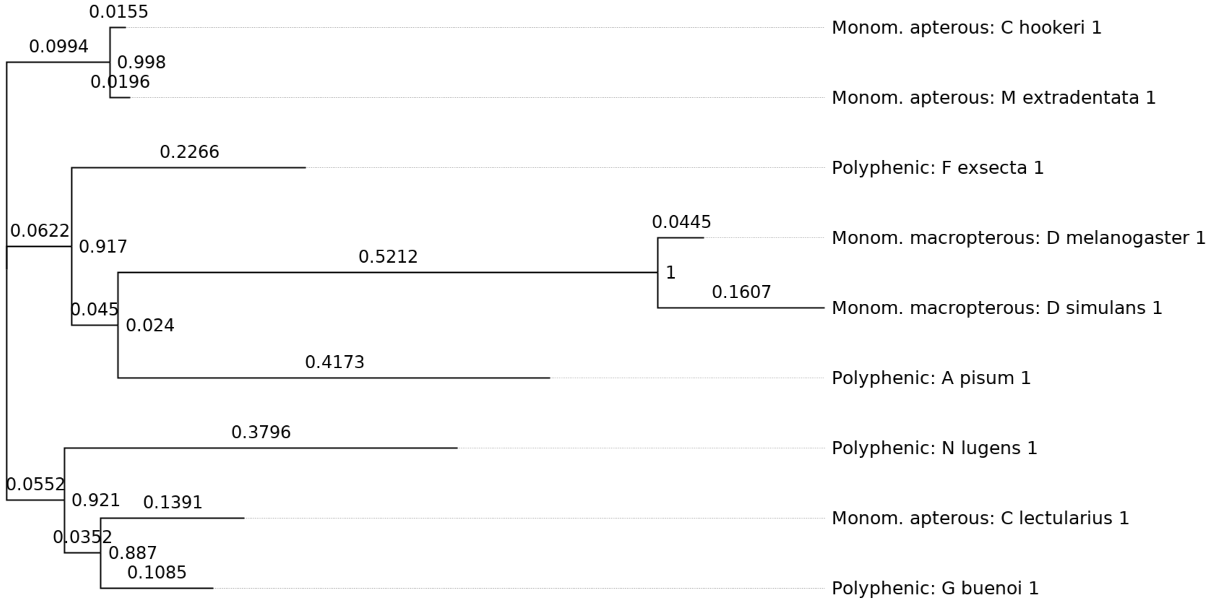


Figure 5. Phylogenetic tree of putative homologous Ecdysone proteins of nine species with varying wing morphologies. Branch lengths and internal node support values (local support values by resampling the site likelihoods 1000 times and executing the Shimodaira Hasegawa test (Shimodaira & Hasegawa 1999)) are included. This illustration was implemented with FigTree v. 1.4.4 (Rambaut 2018).

As was the case for the example presented in Figure 5, no discernible patterns of clustering between winged and wingless species could be recognised on the basis of manually evaluating all the eight phylogenetic trees constructed for this study. The patterns of clustering could indicate that species with wings (i.e. monomorphic macropterous and polyphenic) would be evolutionarily closer to each other than wingless species (monomorphic apterous).

4 Discussion

One of the main aims of this project was to develop a bioinformatic pipeline for identifying homologous genes with poorly annotated and often fragmented genomes, a frequent feature of non-model organisms' genomes. This pipeline was then tested to identify homologous wing development genes for *D. melanogaster* in eight other published insect genomes to provide potential insights into the evolutionary genetic mechanism and history behind the observed loss of wings in some lineages.

Out of the 21 genes of interest outlined in Table A1 in Appendix A, eight were tested on the bioinformatic pipeline and found to be homologous. These were found through a combination of results from text searches in annotations, exonerate searches in annotated polypeptide files and in genome assemblies as well as manual curation of MPAs of putative homologs. That these eight were found and selected for further analyses, was based on that for all these genes, there were found either name matches in annotation files or some number of exonerate matches in annotated polypeptide files. As can be seen from Table C1 in Appendix C, there were found name matches in also in the other 13 genes for several species, but they were not pursued further with the developed pipeline (i.e. the pipeline was not fully applied to them). This was mainly due to two factors. Firstly, the pipeline was in continuous development and the eight genes were selected at an earlier stage when it was not clear yet that the pipeline would be expanded to exonerate searches against genome assemblies. Secondly, and more importantly, there was not enough time in the project to neither search if there were more annotated polypeptide files available for the species of interest which lacked name matches for the 13 genes nor search with exonerate against the genome assemblies with the 13 protein sequences. Hence, this complementary work for the 13 other genes remains something to be done some time in the future.

As discussed earlier, the pipeline developed contained text searching in annotation files, searching with exonerate in annotated polypeptide files and searching with exonerate in genome assemblies as well as manual curation of MPAs produced by candidates of the previous three steps. Interestingly, the pipeline containing all these steps did not use the most common tool for identifying homologues, BLAST. One of the main goals for searching the homologous *Drosophila melanogaster* proteins in other species was to be able to compare in multiple protein alignment the whole proteins of all species. With BLAST obtaining whole proteins can be challenging. BLAST calculates local sequence alignments which identify the most similar region between two sequences, such as a similar domain, but that wouldn’t necessarily guarantee that the protein where the domain exists is homologous. One solution for trying to solve the multitude of matches produced by BLAST is through tiling (for which there are implementation made available e.g. in Bioperl (Stajich *et al.* 2002, Bioperl Community 2019)) but the challenge is how to know where the proteins end when there can be introns in the middle too. Further complications come from the fact that HSPs returned by BLAST don’t necessarily correspond to the exons. These considerations led to the choice of using exonerate which can handle introns and thus more likely find whole homologous proteins because all domains would be detected in the alignment.

Finding similarly named proteins was the first step in pipeline used for putative homology detection. Benefit of this approach is that it builds on already established annotation data and there is less need for reinventing the wheel. However, building on already established annotation data can also be its weakness because annotations can sometimes be ominous in containing errors especially if they are produced by automated annotation pipelines and lack any external support such as RNA-sequencing data. This was largely the case for *C. hookeri* and *M. extradentata* annotations. Another drawback of name matching approach is that not all homologous proteins are named consistently. Overall, as can be seen in Table D1 in Appendix D, most matching proteins were incorporated into MPAs through having been found in annotations.

Next step in the pipeline for finding putative homologous proteins was searching in annotated polypeptide sequences of a species. Again, same strengths and weaknesses of building on already found data can be stated about this step as the previous one. In contrast to the next step (and for that matter for BLAST search results too), the greatest benefit of this step is in that the whole protein can be obtained if the match with exonerate can be deemed significant enough with respect to query coverage and alignment raw score.

Due to not as high quality in gff-annotation files available for *C. hookeri* and *M. extradentata*, searching for homologues in their genome assemblies was necessary. As was noted earlier the results produced with this method were significantly better than those found in the previous step. However, the greatest drawback of this method is that it can return only what is matched in the alignment and that the matches are biased by *D. melanogaster* protein sequences.

The manual curation was carried out by attempting to preview the whole alignment and trying to identify sequences which seemed to be “off” by not having well aligned residues in most parts of its length. This curation was one of the most challenging parts in the project as well as the least objective and thus development of an automated and unbiased approach would be desirable both from an objective viewpoint as well as from a time saving perspective. One way to accomplish this could be by using a program called TrimAl to remove spurious sequences from the alignment (Capella-Gutierrez *et al.* 2009). TrimAl uses two user-given parameters, minimum score of overlapping residues and minimum percentage of how much of the sequence should overlap (with the previous mentioned minimum score) with others, to filter out non-fitting sequences.

One of the questions of interest of this project was the application area of studying wing development differences in genotypes of species with different wing morphologies. MPAs and trees obtained for this project aim to attempt to start shed light on that. What was searched was if there were e.g. indels in the sequences of wingless species which could be hypothised to be correlating with loss of wings in them. However, at this stage no definitive answer can be given to that question based on the results so far. All of the MPAs, one good example of them being the MPA of Ecdysone receptor putative homologs in Figure E1 in Appendix E, were quite patchy. In order to draw conclusions with some level of confidence from the MPAs some type of more quantitative way to assess the MPAs than eyeballing would be necessary. Further work in finding the correct tool for this is needed.

Figure 5 exemplifies well also how trees made from the eight genes in general became. No clear clustering of sequences from winged and unwinged species was detectable. FastTree v. 2.1.10 (Price *et al.* 2010) was used for constructing them. No trimming of low-confidence regions of the produced MPAs was carried out beforehand because FastTree is able to trim the alignments on its own. However, as the constructed trees were not as consistent as was hoped, trimming with an external trimming software would have been worth a try. One possible candidate of such trimming software is TrimAl (Capella-Gutierrez *et al.* 2009). This task of experimentation with trimming software is left for future as well.

When searching homologous protein sequences in the annotated polypeptide files and genome assemblies, lexicographically first translated isoforms of *D. melanogaster* genes presented in Table A1 in Appendix A were used as query sequences. The choice of just choosing one translated isoform was based on the assumption that as the evolutionary distances between *D. melanogaster* and the other species were large, in comparison, a choice of one translated isoform over another wouldn’t make much of a difference. Further, just choosing the first one was computationally simplest as there is always at least one translated isoform available.

The pipeline developed for homology detection can be easily expanded to include other species. For instance, at a later stage *Timema cristinae* annotated polypeptide sequences and genome assembly could be used as target sequences for homology searches with exonerate. Perhaps homology searching genome assemblies could be done with closer homologous protein query sequences than the ones from *D. melanogaster.* This might alleviate the problems caused by being biased by what is found only in *D. melanogaster*. Furthermore, to gain more confidence in the annotations and having picked the right proteins, proteins matched with exonerate searches against *D. simulans* could be compared with those obtained from FlyBase (and which ended up being used).

Right from the beginning of the project a personal goal of mine was to learn and apply practices that increase reproducibility in computational biology. Hence, practices such as use of computational notebooks and version control with Git and Github were in use right from the outset.

In contrast, use of Docker containers came along slightly later in the project. The tipping point in driving this change came from at times experienced extreme sluggishness of Packrat R dependency management system which was initially used. After some searching online, Docker containers for R called Rocker (Boettiger & Eddelbuettel 2017) arose as the best alternative for Packrat and the use of Docker was soon after adopted into my daily workflow.

Due to the ease of use of Rocker and the vast possibilities of Dockerfiles to set up of almost any computational environment needed in bioinformatics, Docker virtualisation techniques, and possibly in the future Singularity in HPC clusters, will likely become a solid part of my bioinformatics workflows.

In summary, the main goal of this project was to two-fold, firstly to develop a pipeline for homology detection in genomic data of vastly varying quality and secondly to apply the pipeline to finding homologous wing-development genes from *D. melanogaster* in nine other species in order to shed light on whether genotypic differences can explain the loss of wings in some of the nine species. The developed pipeline was partially applied to 21 *D. melanogaster* protein sequences, out of which eight were found to have putative homologs in eight other species and for the 13 other genes the pipeline remains to be applied in its full extent. The differences in putatively homologous protein sequences of the eight genes discovered in *D. melanogaster* and eight other species, were illustrated using multiple protein alignments and phylogenetic trees. No biological inferences were able to be drawn from these, as they in their current form did not indicate any conclusive differences between the winged and non-winged species. However, further work with both the multiple protein alignments and phylogenetic trees is required for a better understanding of the involvement of potential functional changes in homologues genes underlying wing development in explaining the loss of wings in many insects.

5 Acknowledgements

I would like to extend my gratitude to my supervisors, Arild, Erik and Douglas for all the help and inspiration you have given me in this project. It has been a delight to work with you! I would also like to thank my subject reader, Sandra for valuable feedback and the time for meeting with me regarding this project.

References

Abouheif E. 2002. Evolution of the Gene Network Underlying Wing Polyphenism in Ants. Science 297: 249–252.

AJC1 from UK. Bed bug, Cimex lectularius - Wikimedia Commons. online: https://upload.wikimedia.org/wikipedia/commons/3/35/Bed_bug%2C_Cimex_lectularius_%289627010587%29.jpg. Accessed October 3, 2019.

Altschul SF. 2014. BLAST Algorithm. online June 16, 2014: http://doi.wiley.com/10.1002/9780470015902.a0005253.pub2. Accessed October 3, 2019.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. Journal of Molecular Biology 215: 403–410.

Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Research 25: 3389–3402.

André Karwath aka Aka. Drosophila melanogaster - Wikimedia Commons. online: https://commons.wikimedia.org/wiki/File:Drosophila_melanogaster_-_top_(aka).jpg. Accessed October 3, 2019.

Bioperl Community. 2019. BioPerl Tiling HOWTO. online 2019: https://bioperl.org/howtos/Tiling_HOWTO.html. Accessed October 3, 2019.

Boettiger C, Eddelbuettel D. 2017. An Introduction to Rocker: Docker Containers for R. R Journal 9: 527–536.

Capella-Gutierrez S, Silla-Martinez JM, Gabaldon T. 2009. trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. Bioinformatics 25: 1972–1973.

Katoh K, Standley DM. 2013. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. Molecular Biology and Evolution 30: 772–780.

Kelly SA, Panhuis TM, Stoehr AM. 2012. Phenotypic Plasticity: Molecular Mechanisms and Adaptive Significance. Comprehensive Physiology, pp. 1417–1439. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Koonin E V. 2005. Orthologs, Paralogs, and Evolutionary Genomics. Annual Review of Genetics 39: 309–338.

Li W-H. 2006. Homologous, Orthologous and Paralogous Genes. Encyclopedia of Life Sciences 2005.

Merkel D. 2014. Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux Journal 2014:

Mesilaakso L. 2019. In this repository you can find the developed pipeline and most of input data. online 2019: https://github.com/ljmesi/MS-thesis. Accessed October 3, 2019.

Mott R. 2005. Smith-Waterman Algorithm. online September 23, 2005: http://doi.wiley.com/10.1038/npg.els.0005263. Accessed October 3, 2019.

Pearson WR. 2013. An Introduction to Sequence Similarity (“Homology”) Searching. Current Protocols in Bioinformatics 42: 3.1.1-3.1.8.

Pearson WR. 2005. Similarity Search. online September 23, 2005: http://doi.wiley.com/10.1038/npg.els.0005262. Accessed October 3, 2019.

Price MN, Dehal PS, Arkin AP. 2010. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. PLoS ONE 5: e9490.

Rambaut A. 2018. FigTree.

Roff DA. 1990. The evolution of flightlessness in insects. Ecological Monographs 60: 389–421.

RStudio Team. 2019. RStudio: Integrated Development Environment for R.

Shimodaira H, Hasegawa M. 1999. Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference. Molecular Biology and Evolution 16: 1114–1116.

Simpson SJ, Sword GA, Lo N. 2011. Polyphenism in Insects. Current Biology 21: R738–R749.

Slater GSC, Birney E. 2005. Automated generation of heuristics for biological sequence comparison. BMC bioinformatics 6: 31.

Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, Fuellen G, Gilbert JGR, Korf I, Lapp H, Lehväslaiho H, Matsalla C, Mungall CJ, Osborne BI, Pocock MR, Schattner P, Senger M, Stein LD, Stupka E, Wilkinson MD, Birney E. 2002. The Bioperl toolkit: Perl modules for the life sciences. Genome Research 12: 1611–1618.

Thompson JD, Poch O. 2006. Sequence Alignment. Encyclopedia of Life Sciences 81–115.

Thurmond J, Goodman JL, Strelets VB, Attrill H, Gramates LS, Marygold SJ, Matthews BB, Millburn G, Antonazzo G, Trovisco V, Kaufman TC, Calvi BR, Perrimon N, Gelbart SR, Agapite J, Broll K, Crosby L, Santos G dos, Emmert D, Gramates LS, Falls K, Jenkins V, Matthews B, Sutherland C, Tabone C, Zhou P, Zytkovicz M, Brown N, Antonazzo G, Attrill H, Garapati P, Holmes A, Larkin A, Marygold S, Millburn G, Pilgrim C, Trovisco V, Urbano P, Kaufman T, Calvi B, Czoch B, Goodman J, Strelets V, Thurmond J, Cripps R, Baker P. 2019. FlyBase 2.0: the next generation. Nucleic Acids Research 47: D759–D765.

Vellichirammal NN, Gupta P, Hall TA, Brisson JA. 2017. Ecdysone signaling underlies the pea aphid transgenerational wing polyphenism. Proceedings of the National Academy of

Sciences 114: 1419–1423.

Xie Y. 2014. knitr: A Comprehensive Tool for Reproducible Research in R. In: Stodden V, Leisch F, Peng RD (ed.). Implementing Reproducible Research, p. 448. Chapman and Hall/CRC,

Xu H-J, Xue J, Lu B, Zhang X-C, Zhuo J-C, He S-F, Ma X-F, Jiang Y-Q, Fan H-W, Xu J-Y, Ye Y-X, Pan P-L, Li Q, Bao Y-Y, Nijhout HF, Zhang C-X. 2015. Two insulin receptors determine alternative wing morphs in planthoppers. Nature 519: 464–467.

Appendix A

Table A1. The names of Drosophila melanogaster wing development genes with FlyBase gene IDs used for homology search. The gene symbols are in parentheses.

No	Name	FlyBase gene ID
1	Crustacean cardioactive peptide (CCAP)	FBgn0039007
2	Eclosion hormone (Eh)	FBgn0000564
3	Bursicon (Burs)	FBgn0038901
4	Ecdysone receptor (EcR)	FBgn0000546
5	ultraspiracle (usp)	FBgn0003964
6	Imitation SWI (Iswi)	FBgn0011604
7	broad (br)	FBgn0283451
8	ftz transcription factor 1 (ftz-f1)	FBgn0001078
9	Ecdysone-induced protein 74EF (Eip74EF)	FBgn0000567
10	Death-associated APAF1-related killer (Dark)	FBgn0263864
11	Death related ICE-like caspase (Drice)	FBgn0019972
12	wingless (wg)	FBgn0284084
13	Distal-less (Dll)	FBgn0000157
14	engrailed (en)	FBgn0000577
15	Ultrabithorax (Ubx)	FBgn0003944
16	extradenticle (exd)	FBgn0000611
17	scalloped (sd)	FBgn0003345
18	spalt major (salm)	FBgn0261648
19	spalt-adjacent (sala)	FBgn0003313
20	spalt-related (salr)	FBgn0000287
21	Insulin-like receptor (InR)	FBgn0283499

Appendix B

Table B1. Description of source data used in homology searching.

No	Description of source data
1	*A. pisum* genome annotation of assembly: GCF_005508785.1
2	*C. hookeri* annotated polypeptide sequences of assembly: GCA_002778355.1
3	*C. hookeri* genome annotation of assembly: GCA_002778355.1
4	*C. hookeri* genome assembly GCA_002778355.1
5	*C. lectularius* annotated polypeptide sequences of assembly: GCF_000648675.2
6	*C. lectularius* genome annotation of assembly: GCF_000648675.2
7	*D. melanogaster* genome annotation of assembly: GCF_000001215.4
8	*D. melanogaster* wing protein sequences (See Appendix A for which genes)
9	*D. simulans* annotated polypeptide sequences of assembly: GCA_000754195.3
10	*D. simulans* genome annotation of assembly: GCF_000754195.2
11	*F. exsecta* genome annotation of assembly: GCF_003651465.1
12	*G. buenoi* annotated polypeptide sequences of official gene set version 1.1
13	*G. buenoi* genome annotation of official gene set version 1.1.1
14	*M. extradentata* annotated polypeptide sequences of assembly: GCA_003012365.1
15	*M. extradentata* genome annotation of assembly: GCA_003012365.1
16	*M. extradentata* genome assembly: GCA_003012365.1
17	*N. lugens* genome annotation of assembly: GCA_000757685.1
18	*T. cristinae* genome annotation of assembly: GCA_002928295.1

Appendix C

Table C1. The text search terms used in regular expressions in searching annotation files of ten species and the number of found in these species.

	N. lug	***F. ext***	***D. mel***	***A. pis***	***C. hoo***	***D. sim***	***T. cris***	***M. ext***	***C. lec***	***G. bue***
crustacean cardioactive	0	0	3	0	0	0	0	0	3	0
eclosion hormone	2	0	2	3	0	6	0	0	1	0
bursicon	2	1	2	2	0	0	0	0	2	0
prothoracicostatic peptide	1	0	0	1	0	0	0	0	0	0
ecdysone receptor	2	6	6	3	0	0	0	0	1	3
ultraspiracle	0	0	2	4	0	0	0	0	1	0
imitation	0	0	3	0	0	0	0	0	0	0
broad	4	39	14	11	0	1	0	0	1	2
ftz transcription factor 1	0	0	4	0	0	0	0	0	0	1
chromatin-remodeling complex ATPase chain Iswi-like	0	0	0	1	0	0	0	0	0	0
Ecdysone-induced protein 74EF	2	4	5	2	0	0	0	0	0	0
Death-associated APAF1-related killer	0	0	2	0	0	0	0	0	0	0
death related ICE-like caspase	0	0	1	0	0	0	0	0	0	0

wingless	2	0	9	0	0	1	0	0	0	1
distal-less	2	3	3	1	0	0	0	0	0	1
homeobox protein engrailed	5	4	0	3	0	0	0	0	0	0
ultrabithorax	2	2	6	4	0	0	0	0	1	1
homeobox protein extradenticle	2	9	0	2	0	0	0	0	0	0
scalloped	0	0	21	0	0	0	0	0	0	1
spalt	1	2	7	3	0	1	0	0	0	1
(insulin-like receptor|insulin receptor)	7	11	8	10	0	0	0	0	1	5
forkhead box	22	24	18	16	0	0	0	0	0	0

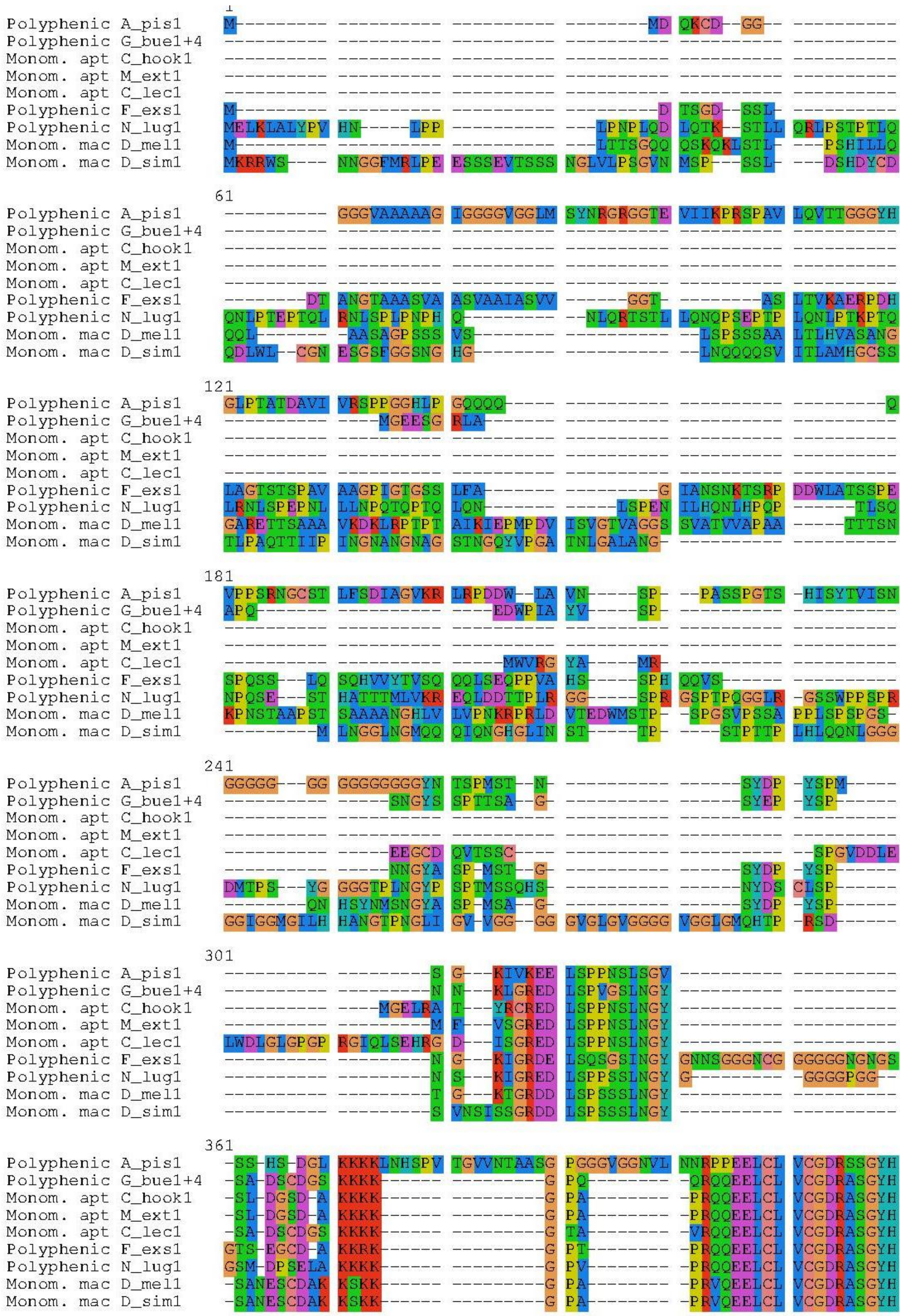
Appendix D

Table D1. Gene names of genes for which were found putative homologues in most of the species of interest for further analyses with multiple protein alignment and phylogenetic tree construction. The sources from which the putative homologous protein sequences were found are listed as well. Regex means that the homologous protein was found through searching with a regular expression on the annotation gff-file. FlyB means that the protein sequence was found by searching directly the homologous sequence in FlyBase online database. PP means that the homologous sequence was found by searching in annotated polypeptide files using exonerate. Lastly, GA means that the homologous sequence was found by searching against the genome alignment of the species with exonerate.

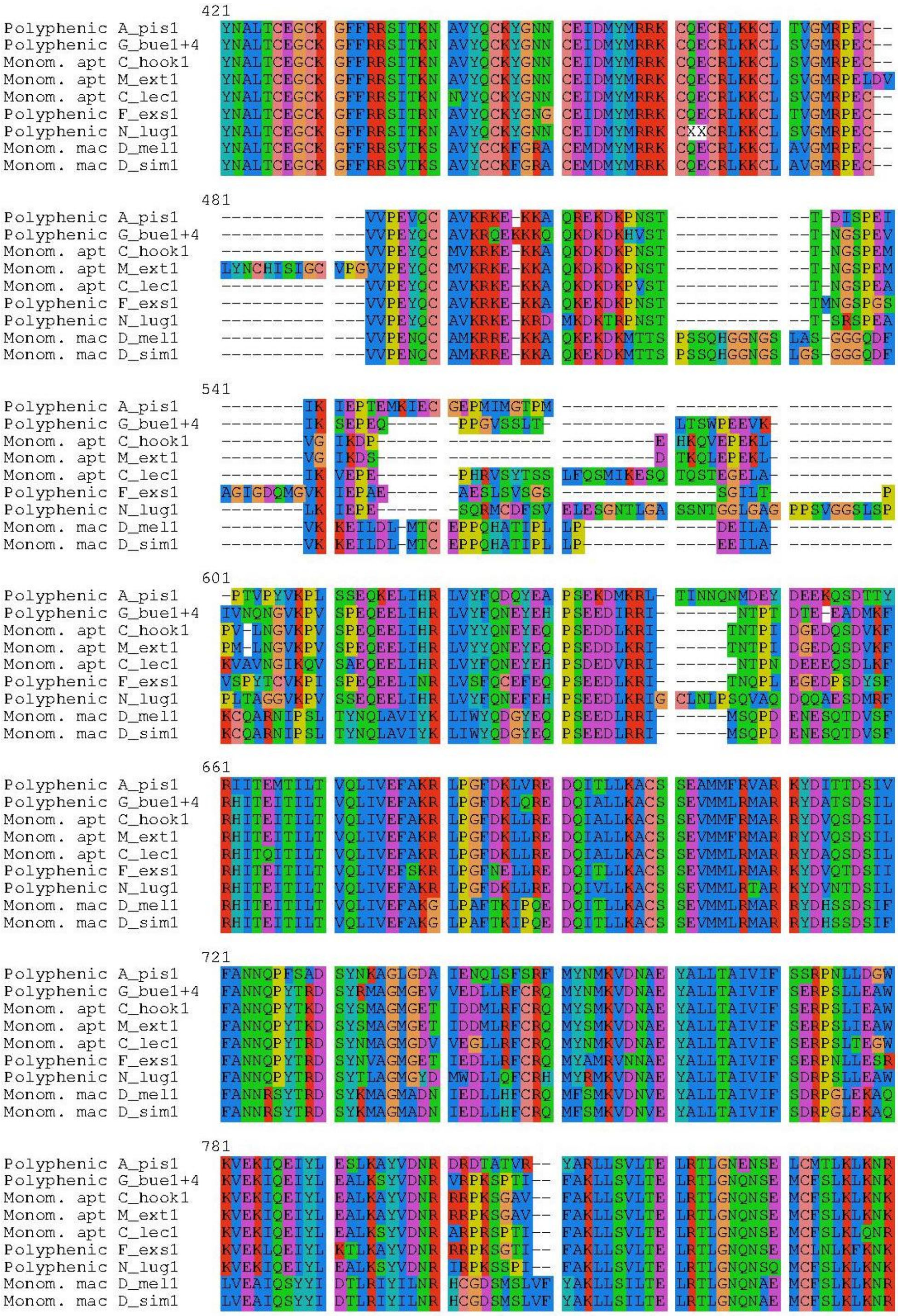
Gene name	N. lug	F ext	D. mel	A. pis	C. hoo	D. sim	M. ext	C. lec	G bu
Ecdysone receptor (EcR)	Regex	Regex	Regex	Regex	PP	FlyB	PP	Regex	Regex
Distal-less (Dll)	Regex	Regex	Regex	Regex	PP	FlyB	PP	PP	Regex
Ultrabithorax (Ubx)	Regex	Regex	Regex	Regex	GA	FlyB	GA	Regex	Regex
Engrailed (en)	Regex	Regex	FlyB	Regex	GA	FlyB	PP	PP	PP
Eip74EF	Regex	Regex	Regex	Regex	PP	FlyB	GA	PP	PP
Extradenticle (exd)	Regex	Regex	FlyB	Regex	PP	FlyB	PP	PP	PP
Insulin receptor (InR)	Regex	Regex	FlyB	Regex	PP	FlyB	PP	Regex	Regex
Broad (br)	Regex	Regex	Regex	Regex	PP	FlyB	PP	Regex	Regex

Appendix E

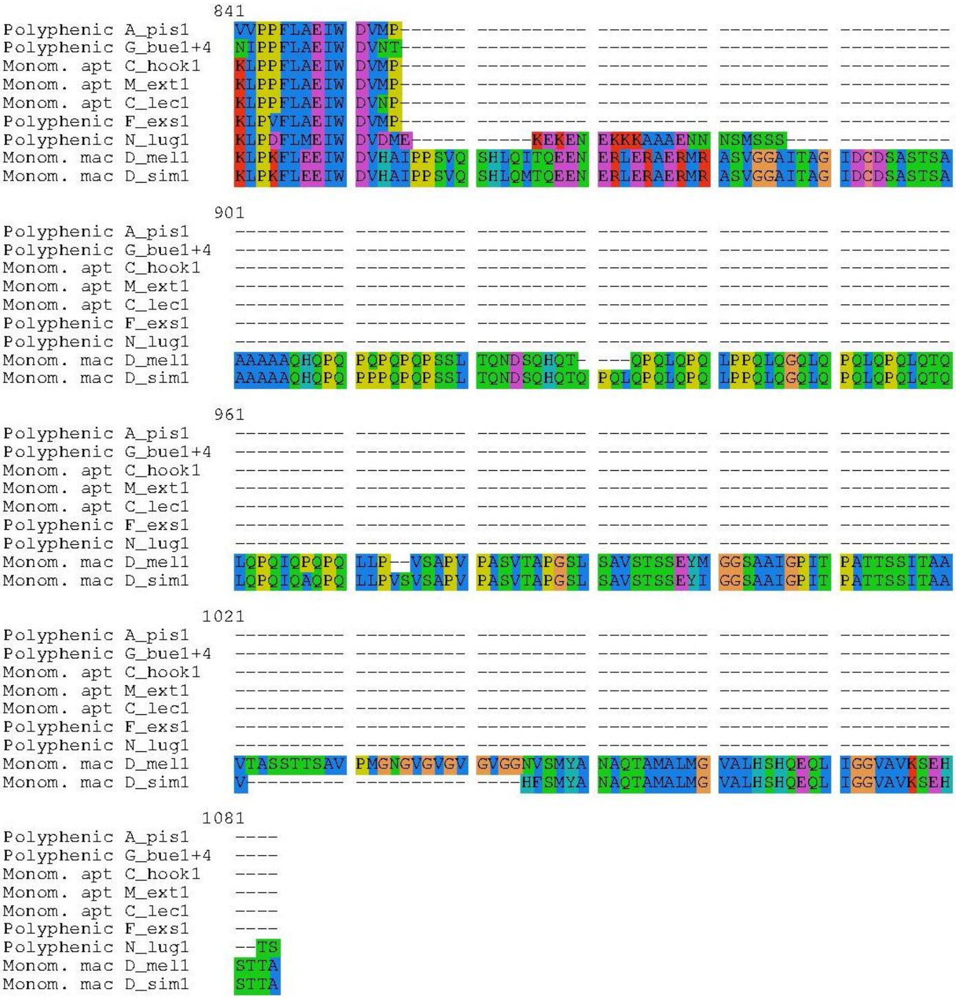
Figure E1. Multiple protein alignment of putative homologous proteins of nine species *A. pisum* (with A_pis1), *G. buenoi* (with G_bue1+4), *C. hookeri* (with C_hook1), *M. extradentata* (with M_ext1), *C. lectularius* (with C_lec1), *F. exsecta* (with F_exs1), *N. lugens* (with N_lug1), *D. melanogaster* (with D_mel1) and *D. simulans* (with D_sim1). The multiple protein alignment was executed with MAFFT v 7.407 which also reordered the sequences according to the guidetree it built for the alignment so that more similar sequences are closer to each other and dissimilar further away.



```
                          1
Polyphenic A_pis1        M--------- ---------- ---------- --------MD QKCD--GG-- ----------
Polyphenic G_bue1+4      ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic F_exs1        M--------- ---------- ---------- ---------D TSGD--SSL- ----------
Polyphenic N_lug1        MELKLALYPV HN-----LPP ---------- ---LPNPLQD LQTK--STLL QRLPSTPTLQ
Monom. mac D_mel1        M--------- ---------- ---------- ---LTTSGQQ QSKQKLSTL- ---PSHILLQ
Monom. mac D_sim1        MKRRWS---- NNGGFMRLPE ESSSEVTSSS NGLVLPSGVN MSP---SSL- ---DSHDYCD

                         61
Polyphenic A_pis1        ---------- GGGVAAAAAG IGGGGVGGLM SYNRGRGGTE VIIKPRSPAV LQVTTGGGYH
Polyphenic G_bue1+4      ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic F_exs1        --------DT ANGTAAASVA ASVAAIASVV ------GGT- --------AS LTVKAERPDH
Polyphenic N_lug1        QNLPTEPTQL RNLSPLPNPH Q--------- --NLQRTSTL LQNQPSEPTP LQNLPTKPTQ
Monom. mac D_mel1        QQL------- -AASAGPSSS VS-------- ---------- --LSPSSSAA LTLHVASANG
Monom. mac D_sim1        QDLWL--CGN ESGSFGGSNG HG-------- ---------- --LNQQQQSV ITLAMHGCSS

                        121
Polyphenic A_pis1        GLPTATDAVI VRSPPGGHLP GQQQQ----- ---------- ---------- ---------Q
Polyphenic G_bue1+4      ---------- ----MGEESG RLA------- ---------- ---------- ----------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic F_exs1        LAGTSTSPAV AAGPIGTGSS LFA------- ---------G IANSNKTSRP DDWLATSSPE
Polyphenic N_lug1        LRNLSPEPNL LLNPQTQPTQ LQN------- -----LSPEN ILHQNLHPQP ------TLSQ
Monom. mac D_mel1        GARETTSAAA VKDKLRPTPT AIKIEPMPDV ISVGTVAGGS SVATVVAPAA -----TTTSN
Monom. mac D_sim1        TLPAQTTIIP INGNANGNAG STNGQYVPGA TNLGALANG- ---------- ----------

                        181
Polyphenic A_pis1        VPPSRNGCST LFSDIAGVKR LRPDDW--LA VN-----SP- --PASSPGTS -HISYTVISN
Polyphenic G_bue1+4      APQ------- ---------- ----EDWPIA YV-----SP- ---------- ----------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- ---------- -----MWVRG YA-----MR- ---------- ----------
Polyphenic F_exs1        SPQSS---LQ SQHVVYTVSQ QQLSEQPPVA HS-----SPH QQVS------ ----------
Polyphenic N_lug1        NPQSE---ST HATTTMLVKR EQLDDTTPLR GG-----SPR GSPTPQGGLR -GSSWPPSPR
Monom. mac D_mel1        KPNSTAAPST SAAAANGHLV LVPNKRPRLD VTEDWMSTP- -SPGSVPSSA PPLSPSPGS-
Monom. mac D_sim1        ---------M LNGGLNGMQQ QIQNGHGLIN ST-----TP- ----STPTTP LHLQQNLGGG

                        241
Polyphenic A_pis1        GGGGG---GG GGGGGGGGYN TSPMST--N- ---------- ------SYDP -YSPM-----
Polyphenic G_bue1+4      ---------- -----SNGYS SPTTSA--G- ---------- ------SYEP -YSP------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- -----EEGCD QVTSSC---- ---------- ---------- --SPGVDDLE
Polyphenic F_exs1        ---------- -----NNGYA SP-MST--G- ---------- ------SYDP -YSP------
Polyphenic N_lug1        DMTPS---YG GGGTPLNGYP SPTMSSQHS- ---------- ------NYDS CLSP------
Monom. mac D_mel1        --------QN HSYNMSNGYA SP-MSA--G- ---------- ------SYDP -YSP------
Monom. mac D_sim1        GGIGGMGILH HANGTPNGLI GV-VGG--GG GVGLGVGGGG VGGLGMQHTP -RSD------

                        301
Polyphenic A_pis1        ---------- ---------S G---KIVKEE LSPPNSLSGV ---------- ----------
Polyphenic G_bue1+4      ---------- ---------N N---KLGRED LSPVGSLNGY ---------- ----------
Monom. apt C_hook1       ---------- ----MGELRA T---YRCRED LSPPNSLNGY ---------- ----------
Monom. apt M_ext1        ---------- ---------M F---VSGRED LSPPNSLNGY ---------- ----------
Monom. apt C_lec1        LWDLGLGPGP RGIQLSEHRG D---ISGRED LSPPNSLNGY ---------- ----------
Polyphenic F_exs1        ---------- ---------N G---KIGRDE LSQSGSINGY GNNSGGGNCG GGGGGNGNGS
Polyphenic N_lug1        ---------- ---------N S---KIGRED LSPPSSLNGY G--------- -GGGGPGG--
Monom. mac D_mel1        ---------- ---------T G---KTGRDD LSPSSSLNGY ---------- ----------
Monom. mac D_sim1        ---------- ---------S VNSISSGRDD LSPSSSLNGY ---------- ----------

                        361
Polyphenic A_pis1        -SS-HS-DGL KKKKLNHSPV TGVVNTAASG PGGGVGGNVL NNRPPEELCL VCGDRSSGYH
Polyphenic G_bue1+4      -SA-DSCDGS KKKK------ ---------G PQ-------- -QRQQEELCL VCGDRASGYH
Monom. apt C_hook1       -SL-DGSD-A KKKK------ ---------G PA-------- -PRQQEELCL VCGDRASGYH
Monom. apt M_ext1        -SL-DGSD-A KKKK------ ---------G PA-------- -PRQQEELCL VCGDRASGYH
Monom. apt C_lec1        -SA-DSCDGS KKKK------ ---------G TA-------- -VRQQEELCL VCGDRASGYH
Polyphenic F_exs1        GTS-EGCD-A KKRK------ ---------G PT-------- -PRQQEELCL VCGDRASGYH
Polyphenic N_lug1        GSM-DPSELA KKKK------ ---------G PV-------- -PRQQEELCL VCGDRASGYH
Monom. mac D_mel1        -SANESCDAK KSKK------ ---------G PA-------- -PRVQEELCL VCGDRASGYH
Monom. mac D_sim1        -SANESCDAK KSKK------ ---------G PA-------- -PRVQEELCL VCGDRASGYH
```



```
                         421
Polyphenic A_pis1        YNALTCEGCK GFFRRSITKN AVYQCKYGNN CEIDMYMRRK CQECRLKKCL TVGMRPEC--
Polyphenic G_bue1+4      YNALTCEGCK GFFRRSITKN AVYQCKYGNN CEIDMYMRRK CQECRLKKCL SVGMRPEC--
Monom. apt C_hook1       YNALTCEGCK GFFRRSITKN AVYQCKYGNN CEIDMYMRRK CQECRLKKCL SVGMRPEC--
Monom. apt M_ext1        YNALTCEGCK GFFRRSITKN AVYQCKYGNN CEIDMYMRRK CQECRLKKCL SVGMRPELDV
Monom. apt C_lec1        YNALTCEGCK GFFRRSITKN NVYQCKYGNN CEIDMYMRRK CQECRLKKCL SVGMRPEC--
Polyphenic F_exs1        YNALTCEGCK GFFRRSITRN AVYQCKYGNG CEIDMYMRRK CQECRLKKCL TVGMRPEC--
Polyphenic N_lug1        YNALTCEGCK GFFRRSITKN AVYQCKYGNN CEIDMYMRRK CXXCRLKKCL SVGMRPEC--
Monom. mac D_mel1        YNALTCEGCK GFFRRSVTKS AVYCCKFGRA CEMDMYMRRK CQECRLKKCL AVGMRPEC--
Monom. mac D_sim1        YNALTCEGCK GFFRRSVTKS AVYCCKFGRA CEMDMYMRRK CQECRLKKCL AVGMRPEC--

                         481
Polyphenic A_pis1        ---------- ---VVPEVQC AVKRKE-KKA QREKDKPNST ---------- --T-DISPEI
Polyphenic G_bue1+4      ---------- ---VVPEYQC AVKRQEKKKQ QKDKDKHVST ---------- --T-NGSPEV
Monom. apt C_hook1       ---------- ---VVPEYQC MVKRKE-KKA QKDKDKPNST ---------- --T-NGSPEM
Monom. apt M_ext1        LYNCHISIGC VPGVVPEYQC MVKRKE-KKA QKDKDKPNST ---------- --T-NGSPEM
Monom. apt C_lec1        ---------- ---VVPEYQC AVKRKE-KKA QKDKDKPVST ---------- --T-NGSPEA
Polyphenic F_exs1        ---------- ---VVPEYQC AVKRKE-KKA QKEKDKPNST ---------- --TMNGSPGS
Polyphenic N_lug1        ---------- ---VVPEYQC AVKRKE-KRD MKDKTRPNST ---------- --T-SRSPEA
Monom. mac D_mel1        ---------- ---VVPENQC AMKRRE-KKA QKEKDKMTTS PSSQHGGNGS LAS-GGGQDF
Monom. mac D_sim1        ---------- ---VVPENQC AMKRRE-KKA QKEKDKMTTS PSSQHGGNGS LGS-GGGQDF

                         541
Polyphenic A_pis1        --------IK IEPTEMKIEC GEPMIMGTPM ---------- ---------- ----------
Polyphenic G_bue1+4      --------IK SEPEQ----- -PPGVSSLT- ---------- LTSWPEEVK- ----------
Monom. apt C_hook1       --------VG IKDP------ ---------- ---------E HKQVEPEKL- ----------
Monom. apt M_ext1        --------VG IKDS------ ---------- ---------D TKQLEPEKL- ----------
Monom. apt C_lec1        --------IK VEPE------ -PHRVSYTSS LFQSMIKESQ TQSTEGELA- ----------
Polyphenic F_exs1        AGIGDQMGVK IEPAE----- -AESLSVSGS ---------- ----SGILT- ---------P
Polyphenic N_lug1        --------LK IEPE------ -SQRMCDFSV ELESGNTLGA SSNTGGLGAG PPSVGGSLSP
Monom. mac D_mel1        --------VK KEILDL-MTC EPPQHATIPL LP-------- ----DEILA- ----------
Monom. mac D_sim1        --------VK KEILDL-MTC EPPQHATIPL LP-------- ----EEILA- ----------

                         601
Polyphenic A_pis1        -PTVPYVKPL SSEQKELIHR LVYFQDQYEA PSEKDMKRL- TINNQNMDEY DEEKQSDTTY
Polyphenic G_bue1+4      IVNQNGVKPV SPEQEELIHR LVYFQNEYEH PSEEDIRRI- ------NTPT DTE-EADMKF
Monom. apt C_hook1       PV-LNGVKPV SPEQEELIHR LVYYQNEYEQ PSEDDLKRI- -----TNTPI DGEDQSDVKF
Monom. apt M_ext1        PM-LNGVKPV SPEQEELIHR LVYYQNEYEQ PSEDDLKRI- -----TNTPI DGEDQSDVKF
Monom. apt C_lec1        KVAVNGIKQV SAEQEELIHR LVYFQNEYEH PSDEDVRRI- ------NTPN DEEEQSDLKF
Polyphenic F_exs1        VSPYTCVKPI SPEQEELINR LVSFQCEFEQ PSEEDLKRI- -----TNQPL EGEDPSDYSF
Polyphenic N_lug1        PLTAGGVKPV SSEQEELIHR LVYFQNEFEH PSEEDLKRIG CLNLPSQVAQ DQQAESDMRF
Monom. mac D_mel1        KCQARNIPSL TYNQLAVIYK LIWYQDGYEQ PSEEDLRRI- -----MSQPD ENESQTDVSF
Monom. mac D_sim1        KCQARNIPSL TYNQLAVIYK LIWYQDGYEQ PSEEDLRRI- -----MSQPD ENESQTDVSF

                         661
Polyphenic A_pis1        RIITEMTILT VQLIVEFAKR LPGFDKLVRE DQITLLKACS SEAMMFRVAR KYDITTDSIV
Polyphenic G_bue1+4      RHITEITILT VQLIVEFAKR LPGFDKLQRE DQIALLKACS SEVMMLRMAR RYDATSDSIL
Monom. apt C_hook1       RHITEITILT VQLIVEFAKR LPGFDKLLRE DQIALLKACS SEVMMFRMAR RYDVQSDSIL
Monom. apt M_ext1        RHITEITILT VQLIVEFAKR LPGFDKLLRE DQIALLKACS SEVMMFRMAR RYDVQSDSIL
Monom. apt C_lec1        RHITQITILT VQLIVEFAKR LPGFDKLLRE DQIALLKACS SEVMMLRMAR RYDAQSDSIL
Polyphenic F_exs1        RHITEITILT VQLIVEFSKR LPGFNELLRE DQITLLKACS SEVMMLRMAR KYDVQTDSII
Polyphenic N_lug1        RHITEITILT VQLIVEFAKR LPGFDKLLRE DQIVLLKACS SEVMMLRTAR KYDVNTDSIL
Monom. mac D_mel1        RHITEITILT VQLIVEFAKG LPAFTKIPQE DQITLLKACS SEVMMLRMAR RYDHSSDSIF
Monom. mac D_sim1        RHITEITILT VQLIVEFAKG LPAFTKIPQE DQITLLKACS SEVMMLRMAR RYDHSSDSIF

                         721
Polyphenic A_pis1        FANNQPFSAD SYNKAGLGDA IENQLSFSRF MYNMKVDNAE YALLTAIVIF SSRPNLLDGW
Polyphenic G_bue1+4      FANNQPYTRD SYRMAGMGEV VEDLLRFCRQ MYNMKVDNAE YALLTAIVIF SERPSLLEAW
Monom. apt C_hook1       FANNQPYTKD SYSMAGMGET IDDMLRFCRQ MYSMKVDNAE YALLTAIVIF SERPSLIEAW
Monom. apt M_ext1        FANNQPYTKD SYSMAGMGET IDDMLRFCRQ MYSMKVDNAE YALLTAIVIF SERPSLIEAW
Monom. apt C_lec1        FANNQPYTRD SYNMAGMGDV VEGLLRFCRQ MYNMKVDNAE YALLTAIVIF SERPSLTEGW
Polyphenic F_exs1        FANNQPYTRD SYNVAGMGET IEDLLRFCRQ MYAMRVNNAE YALLTAIVIF SERPNLLESR
Polyphenic N_lug1        FANNQPYTRD SYTLAGMGYD MWDLLQFCRH MYRMKVDNAE YALLTAIVIF SDRPSLLEAW
Monom. mac D_mel1        FANNRSYTRD SYKMAGMADN IEDLLHFCRQ MFSMKVDNVE YALLTAIVIF SDRPGLEKAQ
Monom. mac D_sim1        FANNRSYTRD SYKMAGMADN IEDLLHFCRQ MFSMKVDNVE YALLTAIVIF SDRPGLEKAQ

                         781
Polyphenic A_pis1        KVEKIQEIYL ESLKAYVDNR DRDTATVR-- YARLLSVLTE LRTLGNENSE LCMTLKLKNR
Polyphenic G_bue1+4      KVEKIQEIYL EALKSYVDNR VRPKSPTI-- FAKLLSVLTE LRTLGNQNSE MCFSLKLKNK
Monom. apt C_hook1       KVEKIQEIYL EALKAYVDNR RRPKSGAV-- FAKLLSVLTE LRTLGNQNSE MCFSLKLKNK
Monom. apt M_ext1        KVEKIQEIYL EALKAYVDNR RRPKSGAV-- FAKLLSVLTE LRTLGNQNSE MCFSLKLKNK
Monom. apt C_lec1        KVEKIQEIYL EALKSYVDNR ARPRSPTI-- FAKLLSVLTE LRTLGNQNSE MCFSLKLQNR
Polyphenic F_exs1        KVEKLQEIYL KTLKAYVDNR RRPKSGTI-- FAKLLSVLTE LRTLGNQNSE MCLNLKFKNK
Polyphenic N_lug1        KVEKIQEIYL EALKSYVDNR IRPKSSPI-- FAKLLSVLTE LRTLGNQNSQ MCFSLKLKNK
Monom. mac D_mel1        LVEAIQSYYI DTLRIYILNR HCGDSMSLVF YAKLLSILTE LRTLGNQNAE MCFSLKLKNR
Monom. mac D_sim1        LVEAIQSYYI DTLRIYILNR HCGDSMSLVF YAKLLSILTE LRTLGNQNAE MCFSLKLKNR
```



```
                         841
Polyphenic A_pis1        VVPPFLAEIW DVMP------ ---------- ---------- ---------- ----------
Polyphenic G_bue1+4      NIPPFLAEIW DVNT------ ---------- ---------- ---------- ----------
Monom. apt C_hook1       KLPPFLAEIW DVMP------ ---------- ---------- ---------- ----------
Monom. apt M_ext1        KLPPFLAEIW DVMP------ ---------- ---------- ---------- ----------
Monom. apt C_lec1        KLPPFLAEIW DVNP------ ---------- ---------- ---------- ----------
Polyphenic F_exs1        KLPVFLAEIW DVMP------ ---------- ---------- ---------- ----------
Polyphenic N_lug1        KLPDFLMEIW DVDME----- -----KEKEN EKKKAAAENN NSMSSS---- ----------
Monom. mac D_mel1        KLPKFLEEIW DVHAIPPSVQ SHLQITQEEN ERLERAERMR ASVGGAITAG IDCDSASTSA
Monom. mac D_sim1        KLPKFLEEIW DVHAIPPSVQ SHLQMTQEEN ERLERAERMR ASVGGAITAG IDCDSASTSA

                         901
Polyphenic A_pis1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic G_bue1+4      ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic F_exs1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic N_lug1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. mac D_mel1        AAAAAQHQPQ PQPQPQPSSL TQNDSQHQT- ---QPQLQPQ LPPQLQGQLQ PQLQPQLQTQ
Monom. mac D_sim1        AAAAAQHQPQ PPPQPQPSSL TQNDSQHQTQ PQLQPQLQPQ LPPQLQGQLQ PQLQPQLQTQ

                         961
Polyphenic A_pis1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic G_bue1+4      ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic F_exs1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic N_lug1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. mac D_mel1        LQPQIQPQPQ LLP--VSAPV PASVTAPGSL SAVSTSSEYM GGSAAIGPIT PATTSSITAA
Monom. mac D_sim1        LQPQIQAQPQ LLPVSVSAPV PASVTAPGSL SAVSTSSEYI GGSAAIGPIT PATTSSITAA

                        1021
Polyphenic A_pis1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic G_bue1+4      ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_hook1       ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt M_ext1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. apt C_lec1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic F_exs1        ---------- ---------- ---------- ---------- ---------- ----------
Polyphenic N_lug1        ---------- ---------- ---------- ---------- ---------- ----------
Monom. mac D_mel1        VTASSTTSAV PMGNGVGVGV GVGGNVSMYA NAQTAMALMG VALHSHQEQL IGGVAVKSEH
Monom. mac D_sim1        V--------- ---------- ----HFSMYA NAQTAMALMG VALHSHQEQL IGGVAVKSEH

                        1081
Polyphenic A_pis1        ----
Polyphenic G_bue1+4      ----
Monom. apt C_hook1       ----
Monom. apt M_ext1        ----
Monom. apt C_lec1        ----
Polyphenic F_exs1        ----
Polyphenic N_lug1        --TS
Monom. mac D_mel1        STTA
Monom. mac D_sim1        STTA
```