Detection of Orthologs via Genetic Mapping Augmentation

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Andrew Couperthwaite

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Researchers interested in examining a given species of interest (or target species) that lacks complete sequence data can infer some knowledge of that species from one or more related species that has a complete set of data. To infer knowledge, it is desired to compare the available sequence data between the two species to find orthologs. However, without complete data sets, one cannot be certain of the validity of the detected orthologs.

Using ortholog detection systems in concert with species' mapping data, researchers can find regions of shared synteny, allowing for more certainty of the detected orthologs as well as allowing inference of some genetic information based on these regions of shared synteny. A pipeline software solution, Detection of Orthologs via Genetic Mapping Augmentation (DOGMA), was developed for this purpose.

DOGMA's functionality was tested using a target species, *Phaseolus vulgaris*, which only had partial sequence data available, and a closely related species, *Glycine max*, which has a fully sequenced genome. On sequence similarity alone, which is the standard technique for detecting orthologs, 205 potential orthologs were detected. DOGMA then filtered these results using mapping data from each species to determine that 121 of the 205 were quite likely true orthologs, referred to as putative orthologs, and the remaining 84 were categorized as reduced orthologs as there was either insufficient information present or were clearly outside a noted region of shared synteny. This provides evidence that DOGMA is capable of reducing false positives versus traditional techniques, such as applications based on Reciprocal Best BLAST Hits. If we interpret the output of the Ortholuge program as the correct answer, DOGMA achieves 95% sensitivity. However, it is possible that some of the reduced orthologs classified by DOGMA are actually Ortholuge's false positives, since DOGMA is using mapping data. To support this idea, we show DOGMA's ability to detect false positives in the results of Ortholuge by artificially creating a paralog and removing the real ortholog. DOGMA properly classifies this data as opposed to Ortholuge.

Acknowledgements

Many thanks to Dr. Ian McQuillan and Dr. Kirstin Bett without whom this research would not have been possible.

For my wife and family whose support and patience were invaluable.

Contents

Permission to Use i

Abstract ii

Acknowledgements iii

Contents v

List of Tables vii

List of Figures viii

List of Abbreviations ix

1 Introduction 1
- 1.1 Problem . . . 1
- 1.2 Motivation . . . 1
- 1.3 Objectives . . . 2

2 Background 3
- 2.1 Genetics . . . 3
  - 2.1.1 Genes . . . 3
  - 2.1.2 Linkage . . . 3
  - 2.1.3 Recombination . . . 4
  - 2.1.4 Genetic/Linkage Mapping . . . 4
  - 2.1.5 Physical/Sequence Maps . . . 9
  - 2.1.6 Speciation, Common Ancestry and Phylogeny . . . 10
  - 2.1.7 Orthology . . . 10
  - 2.1.8 Orthology and Synteny . . . 11
  - 2.1.9 Ploidy . . . 13
- 2.2 Comparative Genetics . . . 14
  - 2.2.1 Specific Model and Target Species . . . 14
- 2.3 Bioinformatics Algorithms . . . 15
  - 2.3.1 Needleman-Wunsch and Smith-Waterman alignment algorithms . . . 15
  - 2.3.2 BLAST . . . 17
- 2.4 Ortholog Detection Systems . . . 19

3 A Pipeline Solution 22
- 3.1 Pipeline Design . . . 23
  - 3.1.1 Data Sets . . . 23
  - 3.1.2 Ortholog Detection . . . 27
  - 3.1.3 Ortholog Filtering and Shared Synteny Detection . . . 28
  - 3.1.4 Map Display . . . 32
- 3.2 Differences in Ploidy Number . . . 32
  - 3.2.1 Modularity . . . 35

4 Data Selection, Results, and Discussion 36
- 4.1 Data Selection Tool . . . 36
- 4.2 Data Sources . . . 38
- 4.3 Test Run and Results . . . 38

4.4 Discussion 45
4.5 Validation 49

5 Summary and Future Directions **54**
5.1 Summary 54
5.1.1 Problem 54
5.1.2 Solution 54
5.1.3 Testing 55
5.1.4 Generality 55
5.1.5 Validation 55
5.2 Future Iterations 55
5.3 Future Directions 56

References **60**

A Custom Perl Scripts **61**
A.1 Join Sequence Data to Map Data 61
A.2 Make Correspondences 63
A.3 Find Syntenic Markers 67
A.4 Create Matrix 72

List of Tables

4.1 Set of 121 putative orthologs as detected by DOGMA 39
4.2 Set of reduced orthologs as detected by DOGMA 43

List of Figures

2.1 A hypothetical single crossover event . . . 5
2.2 The likelihood of recombination occurring between two genes is correlated to the distance between those genes . . . 6
2.3 A hypothetical double crossover event . . . 7
2.4 A sample genetic map produced by CMAP [24] . . . 8
2.5 Diagram illustrating the differences between orthologs, out-paralogs and in-paralogs. 12
2.6 A region of collinear orthologs from two species is referred to as *shared synteny* . . . 13
2.7 A simple scoring matrix for global alignment algorithm . . . 16
2.8 Three examples demonstrating match, mis-match, and gap values. . . . 17
2.9 An example of a global alignment dynamic programming matrix. . . . 18
2.10 Ortholuge augments the detection of putative orthologs by including phylogenetic data 21

3.1 Filtering orthologs based on the shared synteny between two species . . . 24
3.2 Knowledge of missing genetic markers in one species may be inferred from another due to the conservation of gene order . . . 25
3.3 Data flow chart describing the DOGMA pipeline . . . 26
3.4 Hypothetical maps showing the filtering of putative orthologs into sets of putative and reduced orthologs . . . 30
3.5 Ortholog filtering general algorithm. . . . 31
3.6 Orthologs and shared synteny as displayed by CMap . . . 33
3.7 Varying ploidy levels across multiple species causes difficulties in locating shared synteny. . . . 34

4.1 Three dimensional chart illustrating the change in distance parameters. . . . 37
4.2 Three regions of shared synteny as detected by DOGMA. . . . 47
4.3 Two regions of shared synteny on a single map. . . . 48
4.4 Two regions of shared synteny on a single map. . . . 49
4.5 Two regions of shared synteny detected via DOGMA using complete data sets. . . . 52
4.6 DOGMA detects artificially introduced paralogs as reduced orthologs. . . . 53

List of Abbreviations

DOGMA	Detection of Orthologs via Genetic Mapping Application
BLAST	Basic Local Alignment Search Tool
AFLP	Amplification Fragment Length Polymorphism
RFLP	Restriction Fragment Length Polymorphism
EST	Expressed Sequence Tag
CMap	The Comparative Map Viewer
RBBH	Reciprocal Best BLAST Hit

Chapter 1

Introduction

1.1 Problem

Life scientists often study model organisms, so named because there is an abundance of biological data available about them, with the expectation that they can use results to infer knowledge in other, related organisms.

For example, with regards to crop plants, there is an abundance of genetic data pertaining to several species, such as *Medicago truncatula*; *Arabidopsis thaliana*; and *Glycine max*, but relatively little genomic data on *Pisum sativum* (pea) or *Lens culinaris* (lentil), crops with obvious importance. This opens the door to the field of *comparative genetics*. In essence, researchers can look to related model species and infer genomic knowledge of their species of interest.

In situations where the complete genetic sequence or complete mapping data of a species of interest (or target species) is not available, a closely related model species can be used to infer a degree of genetic information pertaining to a species of interest. One task comparative geneticists would like to do is to detect orthologs from related species and then use the orthologs to locate and compare regions of shared synteny between organisms (the concepts of orthology and synteny are defined in Chapter 2). This allows the geneticist to make inferences regarding locations of genes on the genomes of interest based on the regions of shared synteny of a model species. There does not yet exist any single pipeline which will perform all of these operations.

1.2 Motivation

As described above, life scientists often study their target species by way of a model organism using comparative genomics. A plant breeder, for example, might wish to improve efficiency in selection by selecting for a genotype (an organisms genetic makeup) rather than a phenotype (an organisms outwardly displayed characteristics). Hypothetically, selecting for a genotype as opposed to a phenotype could allow for a more precise breeding program, where selecting by phenotype alone may have un-intended consequences as selecting for a phenotype may involve masking of the true genotype by environmental influences thereby causing over or under expression of some key genes.

The breeder of a crop for which there is little genomic information may be looking for a few key traits to manipulate and the most efficient way to search for these could be by comparisons to a closely related species for which more genomic information is available. One method for locating key genes in a target species is by comparing the target species to a related model species and searching for copies of genes, called orthologs, and then inferring target gene locations based on surrounding orthologs. Though it is a smaller piece of the overall solution, it is this type of accurate ortholog detection that plant breeders would depend on for improving a plant species.

"A major challenge for comparative legume genomics is to translate information gained from model species into improvements in crop legumes" [45]. The motivation for this thesis is an attempt to improve upon existing ortholog detection techniques to better solve this problem.

1.3 Objectives

The overall objective of this thesis project is to design a tool to allow life scientists to compare syntenic relationships between a model species and related species of interest. A second objective of this project is to deal with the possibility of varying ploidy levels that may arise between the model species and a related species. Further, the relationship between model and related species can present several issues, such as dealing with large sets of heterogeneous data and ensuring correct detection of orthologs and the detection of regions of shared synteny, which must be addressed. The developed pipeline coordinates multiple data sets from three species (target species, model species, and more distantly related species), detects potential orthologs, filters orthologs using genetic mapping data, and produces a graphical map displaying the correlations between species.

Essentially, the pipeline automatically detects putative orthologs and produces graphical representations of the comparable portions of the target genomes, thereby enabling life scientists to compare syntenic relationships between species. Ultimately, this could allow researchers a more narrow scope to use for locating specific genes on a target species. In the absence of a pipeline, the user would have to find and use an ortholog detection program, analyze the results manually, find and use a map visualizing program and manually filter the results. This system will remove the need for such manual labour and present data to the user with a high degree of accuracy.

This thesis builds on existing ortholog detection algorithms and additionally attempts to lower the number of false positives found by using regions of shared synteny of genetic maps to infer additional knowledge regarding the true nature of the orthologs.

The pipeline developed for this thesis is species independent, so that it can be widely used by all life scientists wishing to compare model and related species. We test this tool using specific plant species, but are not restricted to these plants, or even restricted to plants in general. The purpose of this thesis is to develop and test a pipeline solution for accurate ortholog detection.

Chapter 2

Background

In this chapter, we describe all biological and computational preliminaries needed for this thesis. Section 2.1 illustrates the basic biological concepts needed. Section 2.2 provides an overview of the necessary bioinformatics algorithms. Then, Section 2.3 provides an overview of existing software solutions in the same genre of those used in this thesis. Lastly, Section 2.4 provides an overview of the species chosen for testing the pipeline developed in this thesis.

2.1 Genetics

2.1.1 Genes

A *gene* is a sequence of nucleotides located on a chromosome which is the functional unit of inheritance and controls transmission and expression of one or more physical traits [4]. Nucleotides are the base molecules for deoxyribonucleic acid (DNA) and ribonucleic acid (RNA); they are adenine, cytosine, guanine and thymine for DNA and in the case of RNA thymine is usually replaced with uracil. The abbreviations A, C, G, T and U are used to represent each of the nucleotides in a sequence.

The central dogma of molecular biology states the procedure by which DNA is converted, via transcription and translation, into proteins. The sequence of nucleotides (genes) can be transcribed from DNA into RNA which, in turn, can sometimes be translated into a string of amino acids known as a protein. There are regions of DNA found between the genes which are not transcribed, formerly known as 'junk DNA', but are now referred to as non-coding DNA.

2.1.2 Linkage

Genetic linkage, refers to the association of genes on a chromosome [31]. For the concept of linkage, rather than considering gene sequence, we consider gene locus (plural loci)– the location of a gene on its chromosome. Linked genes do not usually follow the Mendelian principle of independent assortment [20], which states that genes assort independently of each other during meiosis. Instead, linked genes appear together in a set of progeny more frequently than is expected [9]. That is, genes

on separate chromosomes will be separated or grouped with equal likelihood during meiosis, and genes which group together more than this expected frequency are said to be linked and are very likely to be located close to one another on the same chromosome.

2.1.3 Recombination

A species is said to be *diploid* when there are 2 complements (2n) of chromosomes in its somatic cells and one complement (n) in its gametes. For example, *Phaseolus vulgaris* is a diploid species as it has 22 chromosomes (n=11) [5]. Each chromosome in a pair of chromosomes, called *homologous chromosomes*, carries the same set of genes as the other, however, there may be some variation within a given gene [9]. These two variations of the same gene are referred to as *alleles* [9]. Typically, these alleles are represented with single letters, such as those in Figure 2.1. Diploid species can have up to 2 alleles per gene in an individual.

Recombination is an event in which one portion of a chromosome is traded with its homolog during meiosis. These recombination events, also known as 'crossing over', are processes which increases genetic variability [9]. The gametes resulting from these crossing over events are referred to as *recombinants*. They will have characteristics different from its parents. An example of recombination events is illustrated in Figure 2.1. Intuitively, if a pair of genes are closer together, then they are less likely to have a crossing over event between them than a pair of genes that are farther apart on a given chromosome, as shown in Figure 2.2.

Multiple crossover events, such as the example displayed in Figure 2.3, are important considerations with regards to genetic mapping, see Section 2.1.4.

2.1.4 Genetic/Linkage Mapping

Intuitively, genetic linkage maps are akin to road maps of a specific genome. Visually, they appear as a set of linkage groups with genetic markers indicating the order and relative genetic distance between genes, as shown in Figure 2.4. A good genetic map should have the same number of linkage groups as there are chromosome pairs in the organism. *Genetic markers* indicate variations in the DNA sequence of different organisms or species [13]. It is often the case that these markers are not genes themselves, but are fragments of DNA in close proximity to a gene. There are three typical categories of genetic markers [13]:

1. *Morphological* markers are phenotypic characteristics. An organism's phenotype is its displayed characteristic. For example, a gene may control flower colour of either red or white, and the colour that is actually displayed on the flower is the phenotype. The difference in phenotype for a given gene is caused by variations in the genes controlling the trait, these are referred to as the dominant or recessive alleles.

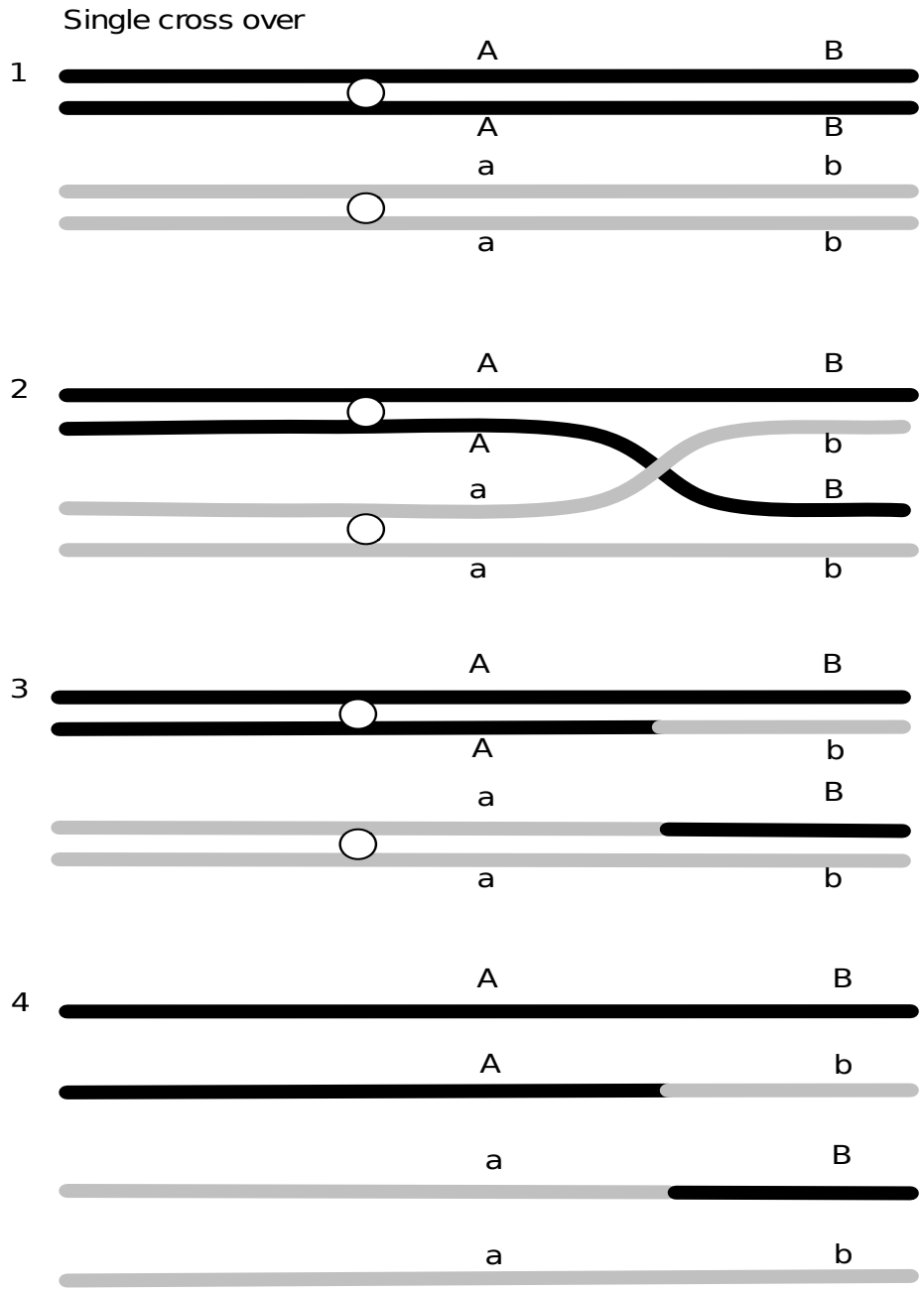

Figure 2.1: This diagram illustrates a hypothetical crossover. Circles represent centromeres. Genes A and B are found on one chromosome and their variants, a and b, in the same location on the sister chromosome (first). During anaphase I of meiosis, a partial exchange of a chromatid occurs (second and third). After meiosis, 4 gametes are produced (fourth), two of which have different genetic combinations than its parent cell. That is, we see the parental genotypes AB and ab, but also recombinant genotypes Ab and aB.

Figure 2.2: Genes located farther apart on the chromosome (left) are more likely to recombine as there are more opportunities for recombination events between them compared with genes which are closer together (right).

2. *Biochemical* markers are the result of variations in the genes, but rather than being displayed as a visual phenotype, they are deduced by the presence of certain enzymes which are observed in biochemical assays [44].

3. *Molecular* or DNA-based markers are the result of slight variations in the sequence information of a segment of DNA compared to another organism [26].

Genetic mapping data is generated from observations of the displayed phenotype of a set of genetic markers from a population of individuals. Each organism is observed for each of the possible phenotypes for each genetic marker (in a diploid organism, there are two possible phenotypes for each marker). The resulting data can appear as a matrix with the individual organisms on the columns, the genetic markers on the rows and each individual record in the matrix an indication of the phenotype observed.

Genetic linkage mapping is the process of taking genetic mapping data and calculating their degree of relatedness and separating subsets of markers into linkage groups. The guiding principle behind genetic mapping is that markers which are found more frequently together among a particular set of offspring from a given cross are likely to be found near each other on the chromosome.

Mapping over short distances is a relatively easy exercise as first done by Morgan [31]. When comparing two genetic markers, the number of recombinants may be calculated. The genetic distance between 2 markers, measured in *centimorgans* (cM), is the fraction of the recombinants over the entire populations as a percentage,

$$\frac{recombinants}{total\,population}100 = distance(cM).$$

When mapping over larger distances, Morgan's formula (above) has poor accuracy compared to mapping over short distances. This is due to increased likelihood of multiple crossover events [31]. In these instances, Haldane's mapping function is used [21]. Consider three genetic markers A, B, and C as well as the distances between them r_{AB},r_{AC} and r_{BC}, measured in centimorgans, where the gene order on the chromosome is ABC and $2r_{AB}r_{BC}$ is the expected frequency of double

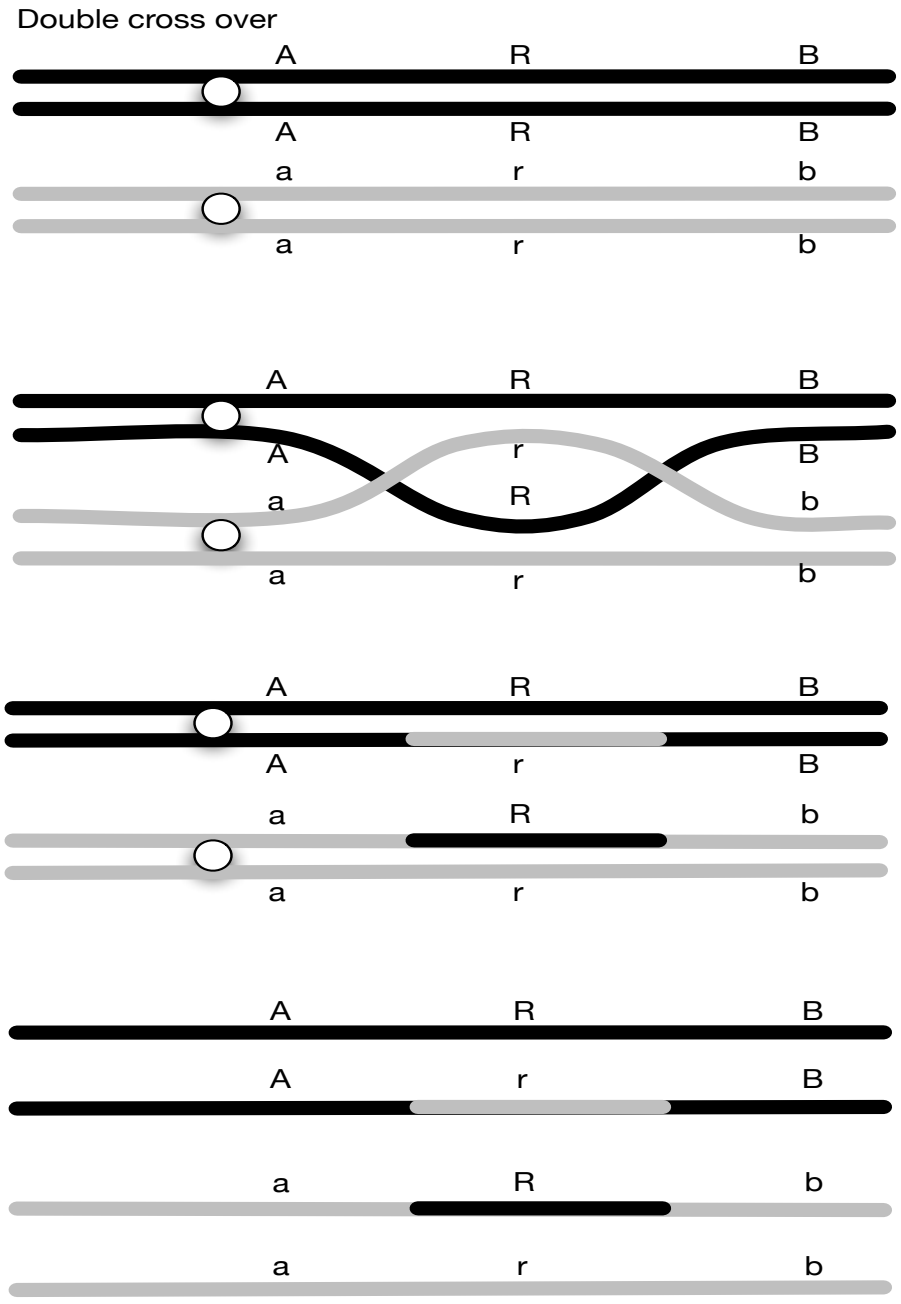

Figure 2.3: This diagram illustrates a hypothetical crossover, similar to Figure 2.1, with the exception that two crossovers take place. Considering only the A and B genes as before, the gametes produced would only have the genotypes AB and ab, and it would be concluded that no crossover has taken place. However, if we look at a third marker, R, we see the parental genotypes ARB and arb as well as the recombinant genotypes ArB and aRb.

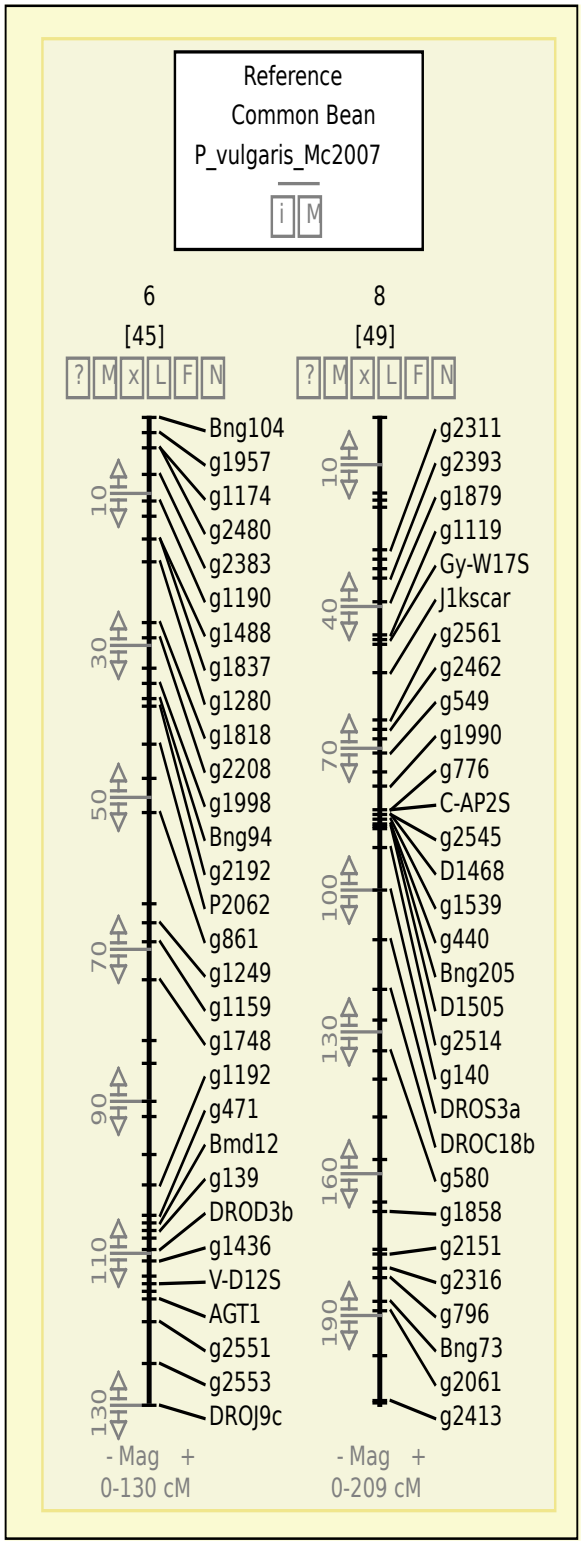

Figure 2.4: A sample genetic (linkage) map displaying two linkage groups (vertical bars) from the species *Phaseolus vulgaris* and the genetic markers (small horizontal lines crossing the linkage groups along with their associated names) in their calculated orders and relative distances. This image was generated using the Comparative Map Viewer (CMAP) [24].

crossovers– two crossover events occur between A and C (C is not pictured but is downstream of B) as illustrated in Figure 2.3. Haldane's formula will calculate the distance between markers A and C as:

$$r_{AC} = r_{AB} + r_{BC} - 2r_{AB}r_{BC}.$$

Mapping is further complicated when crossover interference is considered. This is a phenomenon observed to inhibit multiple crossover events over short segments of chromosomes [23]. Positive interference (the most common form) is the case in which one crossover inhibits the formation of a nearby crossover. Negative interference, whereby one crossover promotes another nearby crossover, rarely occurs. To adjust for this, Kosambi's Map Function is used [22] which is nearly identical to Haldane's with one exception, C is defined as the coefficient of coincidence ($Interference = 1 - C$):

$$r_{AC} = r_{AB} + r_{BC} - 2Cr_{AB}r_{BC}.$$

The value of C depends on the length of the genome segment in question. Kosambi suggested that when $r > 0.5$ then $C = 1$; that is, when the recombination fraction is higher than 0.5 (the markers are unlinked) there is no interference, ($C = 0$ when $r = 0$) [29]. Notice, that when $C = 1$, Kosambi's formula is simplified to Haldane's formula. The value of C should be given as twice the recombination value (i.e. $C = 2r$).

Other related formulae modify Kosambi's formula for specific contexts. Carter & Falconer's mapping function [10] follows nearly identical mathematics to Kosambi's function, but uses a strong value of interference $C = 8r^3$. This function is generally used in areas where interference strongly inhibits other crossover events. Felsenstein's mapping function [15] also follows the same mathematics as Kosambi's except that he accounted for both positive and negative interference by:

$$C = K - (K - 1)2r,$$

where $K = 1$ is an absence of interference, $K < 1$ is positive interference and $K > 1$ is negative interference.

Genetic linkage mapping is useful for determining gene marker order and also the relative genetic distances between those gene markers on chromosomes within a genome. This information can be invaluable in instances where fully sequencing a target species' genome is not practical. This is frequently the case as sequencing an entire genome and then creating a physical map (see Section 2.1.5) is, for the moment, both expensive and time-consuming.

2.1.5 Physical/Sequence Maps

Physical or sequence maps are similar to genetic maps in that they provide a visual description of a species' genome. As with genetic maps, physical maps provide a visual representation of the chromosomes and genes thereon. The major difference between physical and genetic maps lie in the

methods by which they are created. Where genetic maps are calculated based on genetic linkage, physical maps are not measured in centimorgans, but are measured using the number of actual base-pairs separating the markers. Where distances between markers are estimated using statistical methods for genetic maps, on physical maps, these distances are measured physically. Essentially, this means that physical maps are significantly more precise than genetic maps. Physical mapping involves sequencing a species' genome. Then, it is analysed and the various features are plotted on the species' chromosomes. There are many sequencing techniques, such as Sanger-sequencing, and more recently next-generation techniques such as Pyrosequencing and 454-sequencing. For more information on these, we refer the reader to the articles: [40, 39, 14].

2.1.6 Speciation, Common Ancestry and Phylogeny

Over a period of time, the DNA of a species mutates. Speciation occurs when a species undergoes sufficient mutations and large-scale chromosomal rearrangements from its parent species to become genetically isolated and develop into a separate species. This is easily defined for organisms that reproduce sexually, as one can require that two parents are considered to be of the same species if they can produce viable offspring, while separate species cannot produce offspring. It becomes more challenging when concerned with species which reproduce asexually, and in this case speciation is often calculated by the amount of divergence between specific common genes.

Species are said to be related if they have a common ancestor. More precisely, this thesis will concern itself with the most recent common ancestor between two species. Since the two species will have inherited their respective genomes from the ancestor species they will share not only many common genes, but common blocks of genes. The degree to which two species are related can be estimated as the distance between shared DNA via a science known as phylogenetics [9]. In phylogenetics, the distance between two common genes is usually calculated by the number of differences between the two DNA sequences using some scoring mechanisms.

It is intuitive that if two species share a common ancestor species, their genes likely also share a common ancestor gene. We may attempt to find these genes through a study called *orthology*.

2.1.7 Orthology

Orthology is the study of finding copies of the same genes in separate species that were inherited from a common ancestor. Copies of genes in separate, but related, species are referred to as orthologs. That is, an ortholog is a gene shared by separate species having descended from a shared ancestor. A paralog is a similar phenomenon; however, it is a copy of a gene within a single species having descended from a single gene in the shared ancestor usually due to gene or genome duplication.

There has been some contention as to the exact definitions and the use (rather the misuse) of

the terms orthologs and paralogs [25]. The original definitions of the terms ortholog and paralog are as follows [17, 16]:

- orthologs: any two gene copies in different species whose original gene is found in the common ancestor to each species, originated by speciation events,
- paralogs: any two gene copies (in one species) which are resultant from a duplication event.

In a set of articles and responses to articles [42, 25, 35, 28, 34], the definitions of orthologs and paralogs were contested, refined and in one case [42], subdivided. The definitions to be used in this thesis are as follows (as defined in [42]):

- orthologs: copies of genes in separate species derived from a single gene in the last common ancestor of the species,
- paralogs: copies of genes which derive from gene duplication events within a genome. In the same article [42], the definition of a paralog was further refined into:
  - out-paralogs: paralogs which evolved by gene duplications occurring before speciation,
  - in-paralogs: paralogs which evolved by gene duplications occurring after speciation.

Orthologs, in-paralogs and out-paralogs are further explained visually in Figure 2.5. Note that a paralog is either an in-paralog or an out-paralog but not both.

2.1.8 Orthology and Synteny

Synteny refers to all gene loci on a chromosome [34]. It is a term that is related to *linkage* in that both terms make note of sets of genetic markers on a chromosome, except that synteny is not concerned with the observed linkage between them. That is, regardless of whether or not a pair of markers are calculated to be linked, provided they are on the same chromosome, they will be syntenic.

This thesis uses the concepts of orthology and synteny simultaneously. For the purpose of clarity in combining these terms, this thesis will use the term *shared synteny* to refer to a set of orthologs from one chromosome of a species to one chromosome of another (see Figure 2.6).

We can extend this concept to a slightly wider scope by considering regions of chromosomes shared by different species. We will refer to these regions as blocks of shared synteny. This comes from the concept that it is not only genes that are inherited from ancestor species, but large sections of chromosomes up to and including the full chromosome itself. Obviously, some major changes are made to a species' genome in order for speciation to occur, so it may be the case where chromosomal rearrangements have occurred. These rearrangements can involve the breaking and re-associating of various chromosome segments onto other chromosomes, creating new chromosomes from remnants

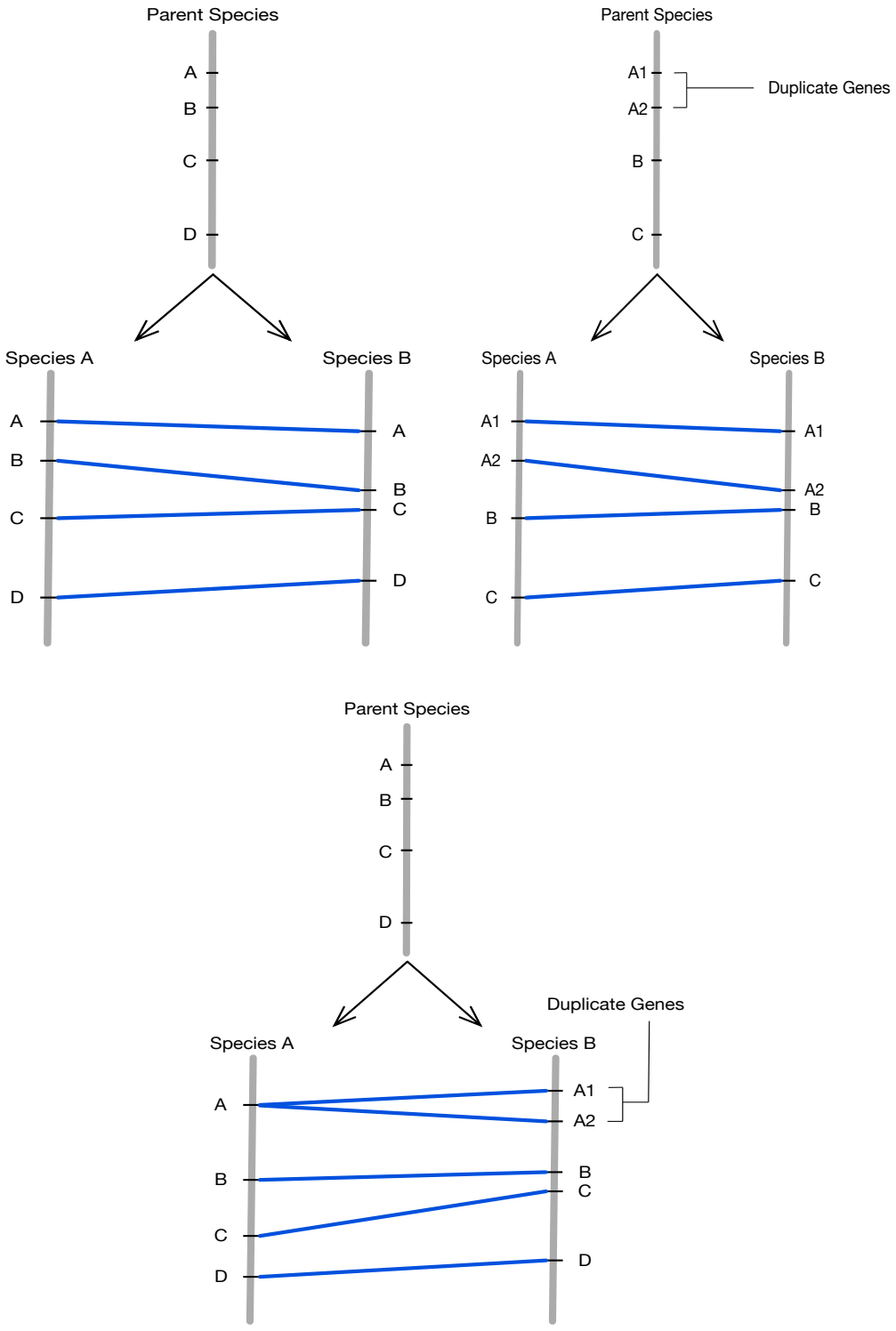

Figure 2.5: The first panel shows a simple one to one relationship of shared genes, these are orthologs. The second panel shows a duplication event in the parent species (A1 and A2) prior to speciation, both of these genes are inherited in both descendant species, these are out-paralogs. The third panel shows a duplication event in one species after a speciation event producing two genes (A1 and A2) are in-paralogs.

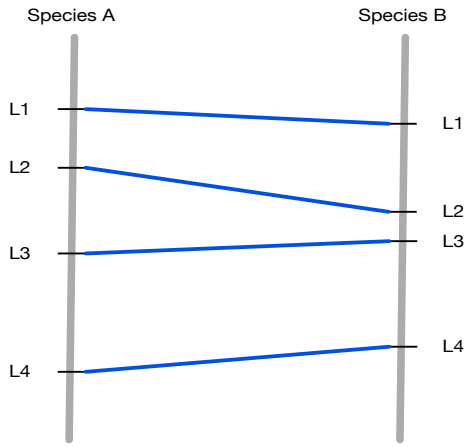

Figure 2.6: This diagram shows a portion of two genetic maps of two hypothetical species. The set of markers L1-L4 on species A are said to be syntenic, as are L1-L4 on species B. Each matching pair (e.g. L1 from A and L1 from B) are said to be orthologous if they originated via speciation. If the gene order is conserved over both regions of the two linkage groups, then it is a region of shared synteny.

of chromosomes, genome duplication, gene duplication etc. Even though a species' chromosomal structure does not appear to completely resemble its related species, it is possible to find blocks of shared synteny between them.

2.1.9 Ploidy

The ploidy of an organism refers to the number of homologous sets of chromosomes contained in somatic cells. We refer to the basic number of chromosomes of an organism's cells by x and the haploid gametic number by n. Then, $2n$ is the number of chromosomes found in zygotic or somatic cells. It is not always the case that $n = x$ [12].

Haploid refers to cells which contain n chromosomes. This occurs most often in reproductive cells, i.e. after a meiotic division. In cases where a organism's zygotic stage is diploid (below) then $n = x$; however, if the zygotic stage is polyploid, then $n > x$ [12].

Diploid organisms are those whose cells contain $2x$ basic chromosomes. *Phaseolus vulgaris* is an example of this; $x = 11$, $2n = 2x = 22$ [12].

Polyploid organisms contain more than $2x$ basic chromosomes, and are common among plants. For example, *Glycine max*, has $x = 10$, but $2n = 4x = 40$ making it a tetraploid organism [12].

Autopolyploid is a type of polyploidy in which copies are sets of chromosomes from the same species. This can occur from a whole genome duplication event [12]. *Glycine max* falls into this category, as it has undergone a complete genome duplication, doubling its $2x$ chromosomal set to $4x$ during evolution.

Allopolyploid is a type of polyploidy in which additional sets of chromosomes are incorporated from a different species via hybridization [12].

With regards to polyploid organisms, we also consider differences in ancient and modern polyploid species. Ancient autopolyploid species have undergone some whole genome duplication, and then continued to produce progeny. During meiosis, however, because there are additional sets of sister chromosomes, chromosomal rearrangements are common. Eventually, the chromosomes stabilize and meiosis performs akin to diploid species, that is, two pairs pair-up. Modern polyploids are much more difficult to analyse as their genomes may not have stabilized. This is an important consideration when choosing species for analysis.

2.2 Comparative Genetics

In the field of comparative genetics, researchers can use related species to infer information about a desired species by making observations about related genomes. The ideal case is when there is an abundance of accurate data available pertaining to the related species from which inferences regarding the desired species can be made. For simplicity, we conform to using the conventional term *model species* when referring to the species with an abundance of available data, and *target species* to refer to the desired species for which we would like to infer some information. A model species is chosen initially because of the relationship formed between its genome and related species. The chosen species is then well studied so that it can be used as a basis for related species in comparative-genetics analysis. For example, *Oryza sativa* (rice) was chosen as a model species as its genome shares a high collinearity with related cereal crops such as maize, barley and wheat [19].

2.2.1 Specific Model and Target Species

Many members of the legume family are used as crop plants and therefore have significant economic value [11]. For this reason, we consider this family worthy of further genetic study. Specifically, within this family, *Phaseolus vulgaris* (the common bean) is of particular interest. *Medicago truncatula* (Barrel medic) is a good model species for the legume family as it shares reasonably good collinearity with the rest of the family and has a relatively small and well characterized genome [11], which makes it easier to fully sequence and analyse. There is, however, a much closer relative to *Phaseolus vulgaris* which is also well studied, *Glycine max* (see Figure 1 of [11]). Because *Glycine max* (soybean) is so closely related to *Phaseolus vulgaris* they are likely to share a high degree of collinearity. As will be seen in Section 2.4, it is sometimes necessary to have two model species for the analysis of the target species. As such we will use all three of the species in this study.

2.3 Bioinformatics Algorithms

Bioinformatics algorithms are used to manipulate and analyse massive amounts of biological data. These can range from sequence assembly to phylogenetic analysis. While there are several types of bioinformatics algorithms, we will concern ourselves with one genre in particular, those for sequence alignment. Sequence alignment is a type of problem where a sequence of DNA, RNA or protein is compared to others and scored based on their similarity. The goal of the problem is to find specific comparisons of the sequences that maximize these scores.

2.3.1 Needleman-Wunsch and Smith-Waterman alignment algorithms

The Needleman-Wunsch global alignment algorithm and the Smith-Waterman local alignment algorithm are two algorithms used for performing sequence alignment [33, 41]. The Needleman-Wunsch algorithm is designed to compare the entirety of two or more DNA (or RNA or protein) sequences and determine the best alignment of them by mapping every possible alignment into a numerical value and choosing the alignment that achieves the highest score. The Smith-Waterman algorithm performs a related task, but is designed to detect the best local alignment between subsequences (smaller segments of the larger sequence) of two or more strings. Both of these algorithms employ a technique known as dynamic programming, which is an algorithmic technique that breaks large problems down into smaller sub-problems, and uses those answers to determine the answers of the larger sub-problems. The main advantage of these two algorithms is that they are completely correct in that they can find an optimal alignment between sequences. An optimal alignment refers to the highest scoring alignment between sequences which can then be used to infer the degree to which the sequences are related. Moreover, as opposed to other algorithms which could find an optimal alignment (trying every alignment and taking highest score), these two algorithms operate relatively efficiently (in polynomial time complexity).

Over time a species genetic makeup may mutate, that is, its DNA may add, delete, or change bases. These changes are reflected in an alignment between two divergent species. To score an alignment between two DNA sequences, the algorithms use three possibilities for each base: a match, a mismatch, or a gap in either species.

For example, consider the following short sequences “ACTGACTGTA” and “ACTAGTGTA”. To align these two sequences we must consider their alignment at every possible position. We will use a simple scoring matrix, Figure 2.7, to describe how well two sequences align. Essentially, we can translate this simple matrix by scoring as follows, scoring a match at each position in the alignment as 1, a mismatch as 0 and a gap as -1.

To find the optimal global alignment score, we will use a matrix with one sequence along the vertical side and the other on the horizontal side, and fill in the scores choosing the highest scoring

	-	A	C	T	G
-	0	-1	-1	-1	-1
A	-1	1	0	0	0
C	-1	0	1	0	0
T	-1	0	0	1	0
G	-1	0	0	0	1

Figure 2.7: A simple scoring matrix for a global alignment algorithm is used to determine the value of a match, mismatch, or gap at any position in a given alignment. In this example, this particular matrix is equivalent to having -1 as a gap penalty, 1 for a match, and 0 for a mismatch score. We could change the values in the matrix to change the score for specific nucleotide matches, mismatches, and gaps.

values for a given cell based on a match, mismatch, or gap. More formally, we can describe each cell $C_{i,j}$ as the maximum of three calculations: the cell above plus the gap penalty $C_{i-1,j} + G$, the cell to the left plus the gap penalty $C_{i,j-1} + G$, or the cell above and to the left plus the match or mismatch score as per the scoring matrix $C_{i-1,j-1} + S(x_i, y_j)$, where x_i is the i'th base of the first sequence x and y_j is the j'th of the second sequence y.

$$C_{i,j} = max \begin{Bmatrix} C_{i-1,j-1} + S(i,j) \\ C_{i-1,j} + G \\ C_{i,j-1} + G \end{Bmatrix}$$

Refer to Figure 2.8 which illustrates how to score a given cell in the matrix and Figure 2.9 for the full alignment matrix for the above sequences.

Using the alignment matrix from the example, Figure 2.9, we can trace the path of the highest scoring global alignment backward to yield the optimal alignment:

```
ACTGACTGTA
ACT-AGTGTA
```

The main downfall of these algorithms is that they are slow and impractical for large data sets. We refer the reader to reference text [27] for general information on local and global sequence alignment, and dynamic programming.

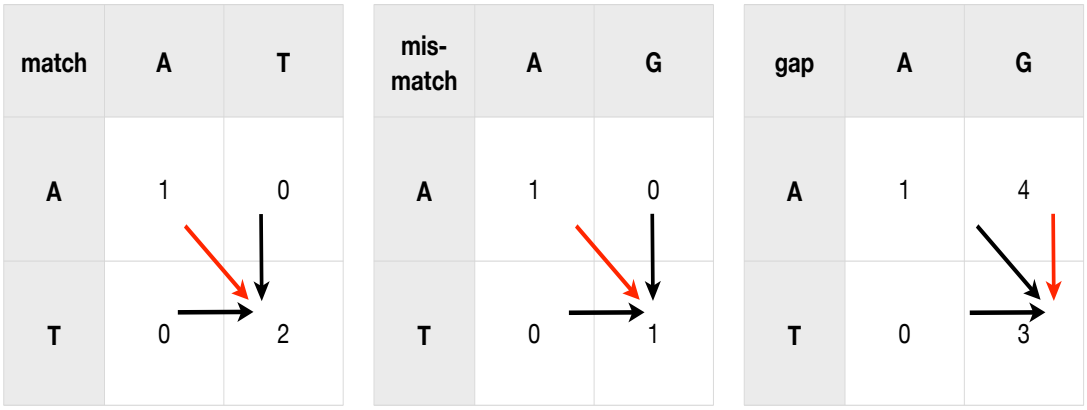

Figure 2.8: Three examples indicating how each one of a match, mismatch, and gap in an alignment is calculated. The left panel shows that the highest score available results from a match. The centre panel shows the highest value will be determined with a mismatch. The right panel shows how the highest score can be determined with a gap.

2.3.2 BLAST

Often, life scientists need to compare a (sometimes large) set of sequences with another large set, such as a database of sequences. This allows the scientist to find specific sequences in the database which are similar to their target sequence. The aforementioned algorithms are often far too slow to perform these tasks in reasonable amounts of time. Therefore a heuristic approach is usually applied instead.

The *Basic Local Alignment Search Tool* (BLAST) [7] is a heuristic algorithm that allows one to search databases for similar sequences and score the similarities between individual pairs of sequences. The heuristic approach performs pair-wise comparisons between sequences, not along their entire sequence, but by short segments called *words*. Once matching words are found, a process called *seeding*, the algorithm extends the comparison in both directions until a certain threshold is met [8]. While there are several parameters that can be used to guide the search in a more precise manner as desired by the user, the parameter most often used is referred to as the *e-value*. The e-value is the expected number of hits occurring by chance in a equally sized database. Therefore, the lower the e-value, the better chance that the current alignment reflects a close evolutionary relationship [6]. The e-value is often set as an input parameter that the user chooses as a threshold cutoff. The failing of this, however, is that if the threshold is set too stringently, weakly aligning sequences may be missed, or set too loosely, can incur an overload of poorly aligning sequences.

	-	A	C	T	G	A	C	T	G	T	A
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
A	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
C	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
T	-3	-1	1	3	2	1	0	-1	-2	-3	-4
A	-4	-2	0	2	3	3	2	1	0	-1	-2
G	-5	-3	-1	1	3	3	3	2	2	1	0
T	-6	-4	-2	0	2	2	2	4	3	3	-2
G	-7	-5	-3	-1	1	2	2	3	5	4	3
T	-8	-6	-4	-2	0	1	2	3	4	6	5
A	-9	-7	-5	-3	-1	1	1	2	3	4	7

Figure 2.9: An example of a global alignment dynamic programming matrix. The scoring matrix in Figure 2.8 was used to calculate each cell in the matrix, and then we are able to trace the path backwards based on how each score was calculated (marked in red) and use that to describe the optimal alignment.

2.4 Ortholog Detection Systems

Ortholog detection across species can be a challenging task. Often, a simplistic approach is taken which involves performing BLAST [7] queries using sequences from a target species against a related species and enforcing some rigid threshold value to ensure the validity of the results [32]. However, this may not be the most efficacious method as it has the potential for returning many false positives, especially when incomplete data sets are used. That is, if the true ortholog is missing from the target species data, then BLAST may return a sequence as an ortholog which is not truly an ortholog. The flaw is magnified when paralogs within the target species are considered; it may easily detect false positives, should the true ortholog not be present in the data set but a paralog is instead present. In this case, it will easily find the paralog with a high degree of similarity, but it will not be the true ortholog. Therefore, other solutions have been devised to make ortholog identification more robust. Tools such as INPARANOID [37], and OrthoMCL [30] make use of a common methodology known as the *Reciprocal Best BLAST Hit* (RBBH) [38].

The RBBH process involves performing BLAST queries in both directions; that is, performing a BLAST query of all the sequences of the first species against that of the second, and then performing the query again with the second species sequences against the first species. Then, if the top hits in both BLAST directions are the same, then the likelihood that the sequences are orthologous is quite high. For example, consider two species A and B, RBBH will perform a BLAST search of all sequences from species A against all sequences from species B and then vice-versa. If the best match for gene A1 from species A is found to be gene B1 in species B and the best match for gene B1 in species B is found to be A1 from species A then they are said to be the reciprocal best BLAST hit and A1 and B1 are considered to be orthologs.

A standard BLAST query may have low specificity with regards to detecting orthologs. It is especially problematic when paralogs are considered, as BLAST may detect paralogs as putative orthologs without any method of verification. RBBH effectively solves the potential low specificity issues that can arise from only doing a single BLAST query by performing the BLAST in both directions.

A significant issue with the above RBBH based tools is that they require complete data sets for each species in order to claim the validity of the results. In comparative genetics, it is often the case that researchers work with incomplete data sets, often due to lack of resources for completing sequence data for a given species. Indeed, this is one of the reasons for making use of comparative genetics.

The problem that arises is again one of sensitivity and specificity. Standard RBBH methods cannot guarantee that the predicted orthologs they detect are indeed 'true' orthologs rather than paralogs when working with incomplete data sets. For example, consider the ortholog detection

of an incomplete data set from species A and a complete data set from species B. An RBBH may detect a copy of a gene which is not a paralog simply because the data for the ortholog was not present in the data set. Extending the above example, consider species A now having a paralog for A1 called A2. Gene A1 is still orthologous with gene B1 of species B, however due to an incomplete data set, the sequence available for A1 is missing. In this case, RBBH will detect A2 and B1 as orthologs as there is no better match present in the data set.

Another tool called *Reciprocal Smallest Distance* RSD [43], avoids the issue of RBBH being mislead by close paralogs by comparing the reciprocal smallest distance between a set of matching sequences. RSD results are produced by performing multiple sequence alignment on the highest matching hits from a BLAST query followed by maximum likelihood estimation of evolutionary distances to detect orthologs. However, RSD might still detect paralogs as orthologs if the original ortholog sequences are missing from a target set.

Ortholuge [18] is an alternative ortholog detection program that attempts to compensate for the issues that arise from dealing with incomplete data sets by incorporating phylogenetics to evaluate the orthology as detected by the RBBH method, as seen in Figure 2.10. To do so, it requires three data sets: an incomplete query species, a complete model species, and a slightly more distantly related species. We will refer to them as 'in-group 1', 'in-group 2' and 'out-group', respectively. As a first step, Ortholuge performs RBBH between the sequences of all pairs of species. Then, using the potential orthologs as detected by the RBBH it performs a phylogenetic analysis of each hit. The phylogenetic analysis will determine the correctness of the ortholog as it will give an indication of whether the distances between the sequences are in line with each species. That is, sequences from in-group 1 ought to be closer with regards to phylogenetic distance to those of in-group 2 than the out-group. If, on the other hand, the distance between an in-group 1 sequence and the out-group sequence was closer than that of in-group 1 to in-group 2, the hit is unlikely to be a true ortholog and the program will remove it from the set of putative orthologs.

The output from Ortholuge is a set of orthologs as well as the distance ratios calculated for each set which can give an indication of the strength of the search. If the distances between the species are too close or too distant, it is possible that the results can be marred; however, the results will still be no worse than RBBH.

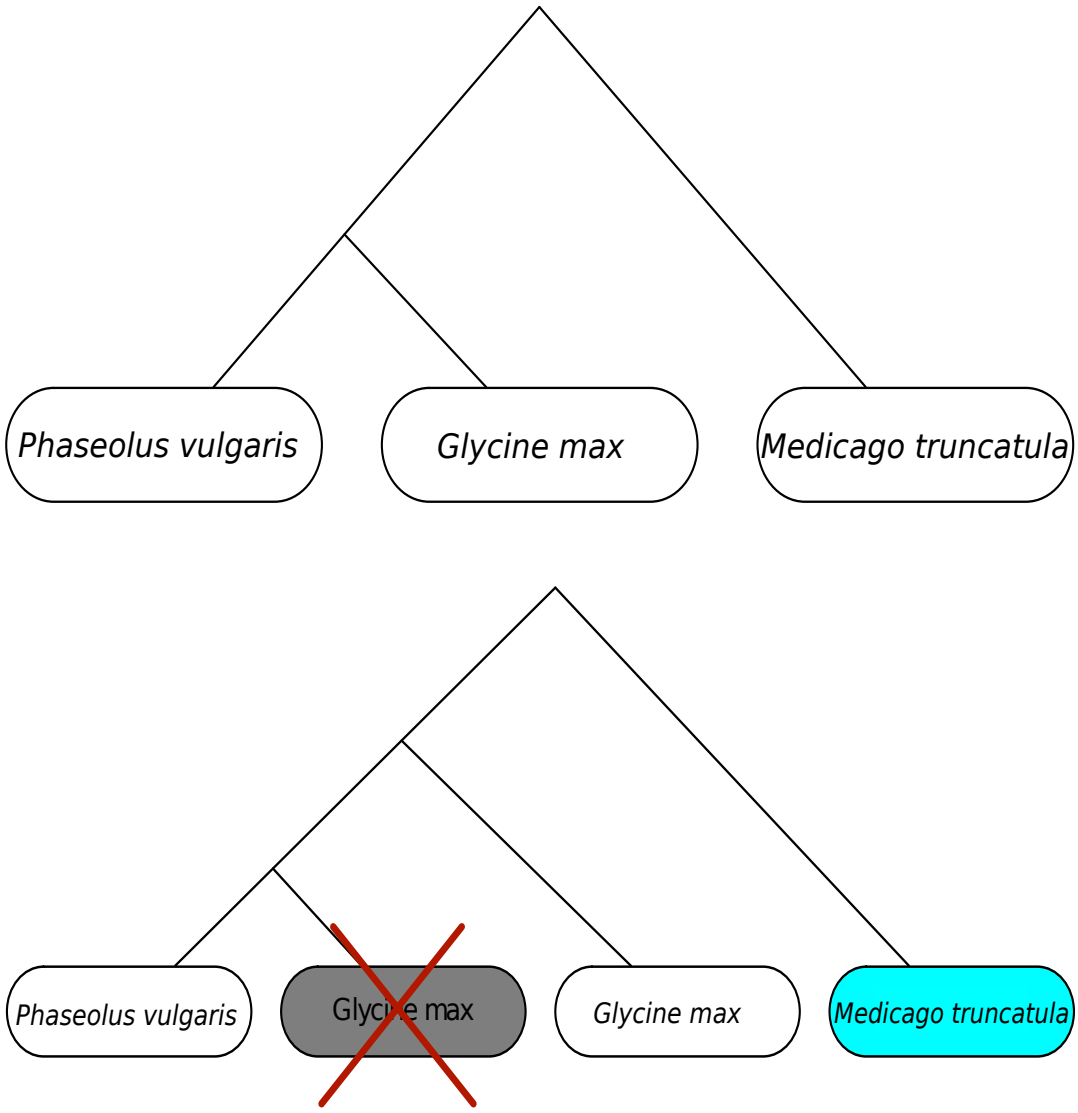

Figure 2.10: The species tree (top picture) shows the ancestral relationship between three species. In this case *Phaseolus vulgaris* is more closely related to *Glycine max* than *Medicago truncatula*. The lower diagram shows a hypothetical gene tree. The crossed out gene is missing, but RBBH detects a match, with the marker highlighted in blue. The highlighted gene is, in fact, an in-paralog. It is possible that a missing ortholog in a data set could allow an in-paralog of that gene to be classified as an ortholog. Ortholuge attempts to compensate for this by including phylogenetic data. If a potential ortholog (blue) is more distantly related than an ortholog from a third related species (*Medicago truncatula*), Ortholuge [18] will not count it among the set of orthologs.

Chapter 3

A Pipeline Solution

The overall objective of this thesis is to develop a tool that uses genetic mapping and sequence data from well-studied organisms to infer knowledge of related target species. Ortholog detection programs that currently exist find orthologs based on their sequence similarity. This technique alone leaves the possibility of false positives; that is, detecting orthologs which are not true orthologs (perhaps paralogs etc.). In instances of gene loss, gene duplication, or incomplete data sets, false positives are more likely to arise as ortholog detection systems may detect other sequences as orthologs even though they are not.

This thesis builds on existing ortholog detection algorithms, but will additionally attempt to lower the number of false positives found by using regions of shared synteny of genetic maps to infer additional knowledge regarding the true nature of the orthologs. To do this, we reclassify the orthologs as detected by Ortholuge into two categories; *putative orthologs* and *reduced orthologs*. A putative ortholog is one which was detected via sequence similarity and whose locus on the genetic map falls into a region of shared synteny, as expected. A reduced ortholog is one which was detected via sequence similarity but whose locus is not within a region of shared synteny, thereby reducing the chance that it is a true ortholog. If, for example (see Figure 3.1 and its caption for additional details), there is a section of shared synteny between two genetic maps, and one ortholog is missing– that is, an ortholog is not detected at a locus as would be expected via shared synteny–, but another sequence is detected which maps to another part of the map (outside the section of shared synteny). Then, the pipeline we develop may classify that detected ortholog as a *reduced* ortholog rather than a *putative* ortholog (see Section 2.1.3). In this way, we filter a set of orthologs into putative and reduced orthologs.

Additionally, because this pipeline specializes in incomplete data sets, users will be able to infer positions of missing genes. For example, if a section of shared synteny between two genetic maps is found, but one of the maps is missing a large number of the genetic markers, a researcher may infer knowledge of the missing markers directly from the more complete genetic map, as shown in Figure 3.2.

A further objective of this thesis is to work with species of varying ploidy levels, as is common in many plant species. An inherent flaw of the RBBH algorithm is that it only detect the best single

hit in either direction. For example, consider searching for orthologs between a diploid species A and a tetraploid species B, where genome B was obtained from an ancestor of species A via full genome duplication. Ideally, for every gene in species A there would be two orthologs detected in species B. However, using the standard RBBH algorithm, this would not be the case. RBBH will only find the first of the orthologs and ignore the other as the algorithm doesn't account for more than one top hit.

From a user's perspective, this system will allow a graphical view of putative orthologs and will highlight regions of shared synteny between two genetic maps. These regions may include such portions of a map whose information regarding markers may now be inferred where it was previously impossible, such as Figure 3.2.

Prior to the development of this pipeline, researchers were performing many of the included tasks manually. There are several programs available for performing the task of ortholog detection, as discussed in Section 2.4. As yet, none of those programs include genetic mapping information as a medium for filtering orthologs in an attempt to lower the number of potential false positives. This pipeline, where possible, filters potential orthologs based on their location in the genome. Further, this thesis attempts to address issues which arise when comparing species of differing ploidy levels.

3.1 Pipeline Design

The pipeline is called "Detection of Orthologs via Genetic Mapping Augmentation" *DOGMA*, and is comprised of three major operations: ortholog detection, ortholog filtering and shared synteny detection, and map display– see Sections 3.1.2, 3.1.3 and 3.1.4 respectively. First, we will elucidate on the required data sets required for each species involved in Section 3.1.1.

Figure 3.3 illustrates the flow of data through the various pieces of the pipeline. At the first phase of DOGMA, various data sources are aggregated and formatted. During the second phase those data sets are input to Ortholuge which produces a set of putative orthologs. The third phase uses genetic (or physical) mapping data along with the putative set of orthologs to locate all of the orthologs on the genetic maps and then proceeds to filter them based on their position in the map. Finally, the fourth phase of the pipeline is a display of the genetic maps and the orthologs illustrating the difference in the type of ortholog present. Each of these phases are explained in detail in the following subsections.

3.1.1 Data Sets

The pipeline makes use of several heterogeneous data sources. These data sources are provided as input and then analyzed at various stages in the pipeline. Depending on the source of the raw data files, some preprocessing may be required in order to ensure correct formatting of the input data.

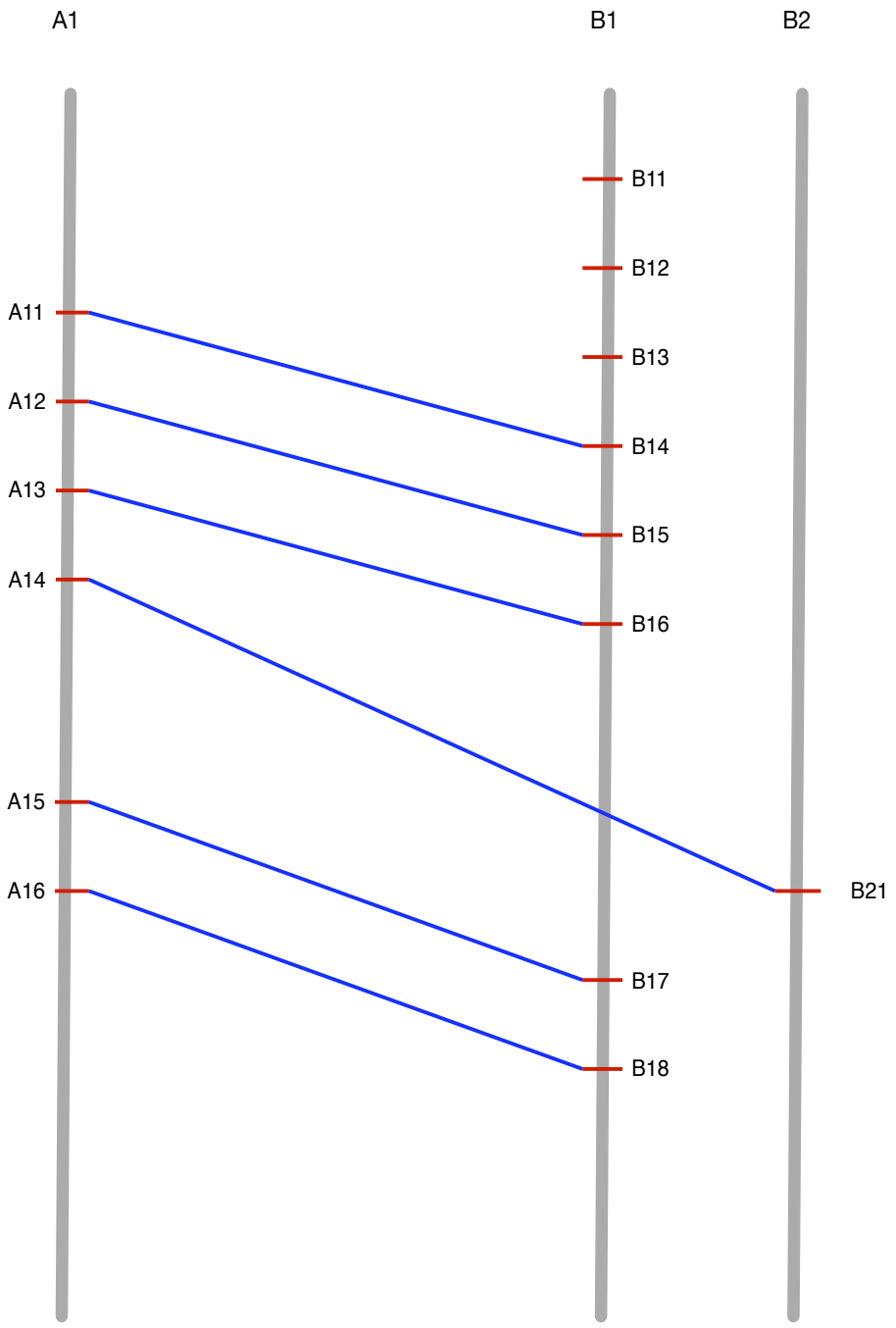

Figure 3.1: If we interpret the section between markers A11 and A16 as a section of shared synteny with the section between markers B14 and B18, then a detected ortholog on linkage group 2 of species B (B21), may be discarded from the set of putative orthologs as it doesn't follow the linearity of the block of shared synteny; however, it would be added to the set of reduced orthologs, as we may be missing data for the real ortholog which could be within the shared synteny.

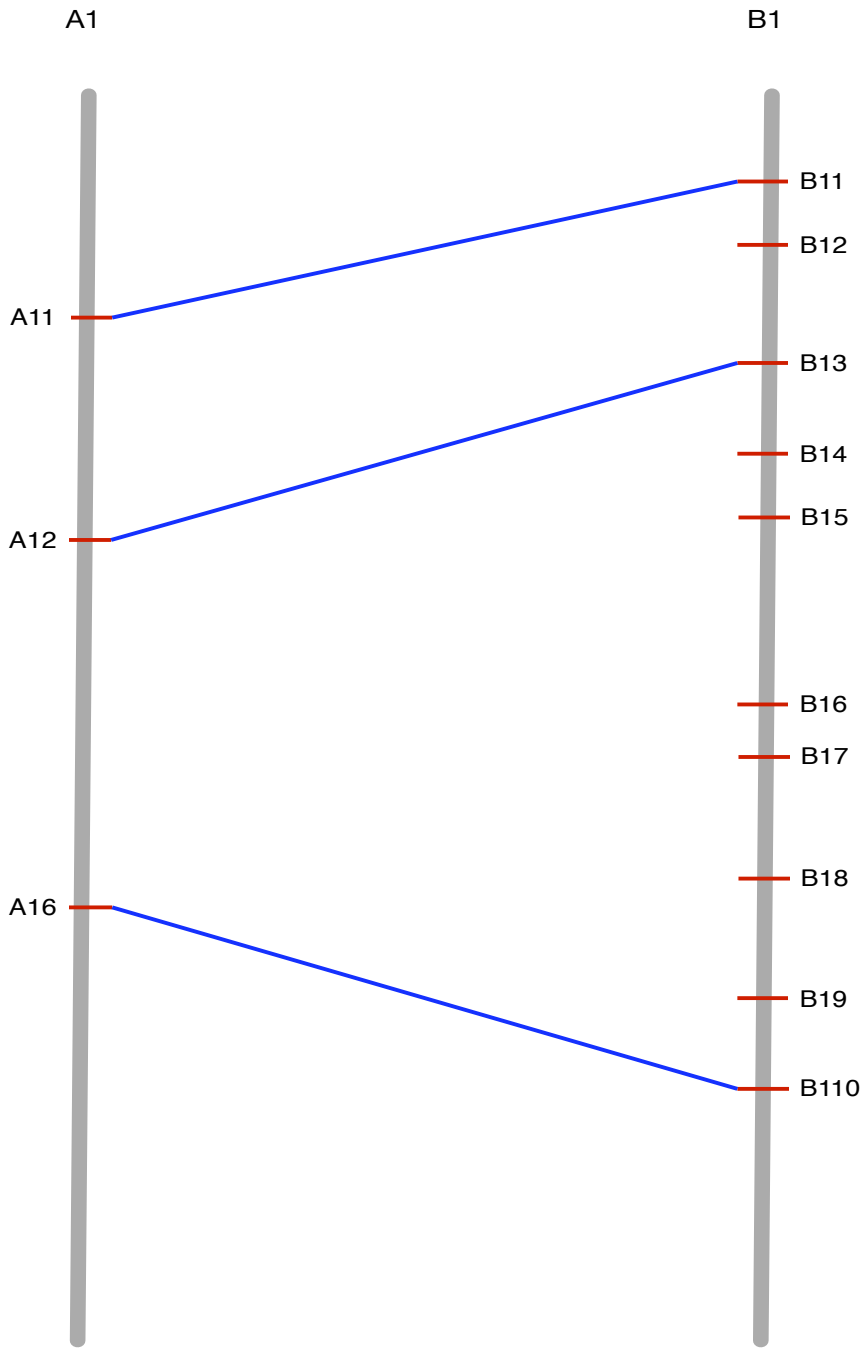

Figure 3.2: A block of shared synteny is shown between two species (A and B). Knowledge of genetic markers missing from the data set of species A may be inferred from species B due to the conservation of gene order. For example, in the above diagram, it is likely that there are six genes not yet found between markers A12 and A16.

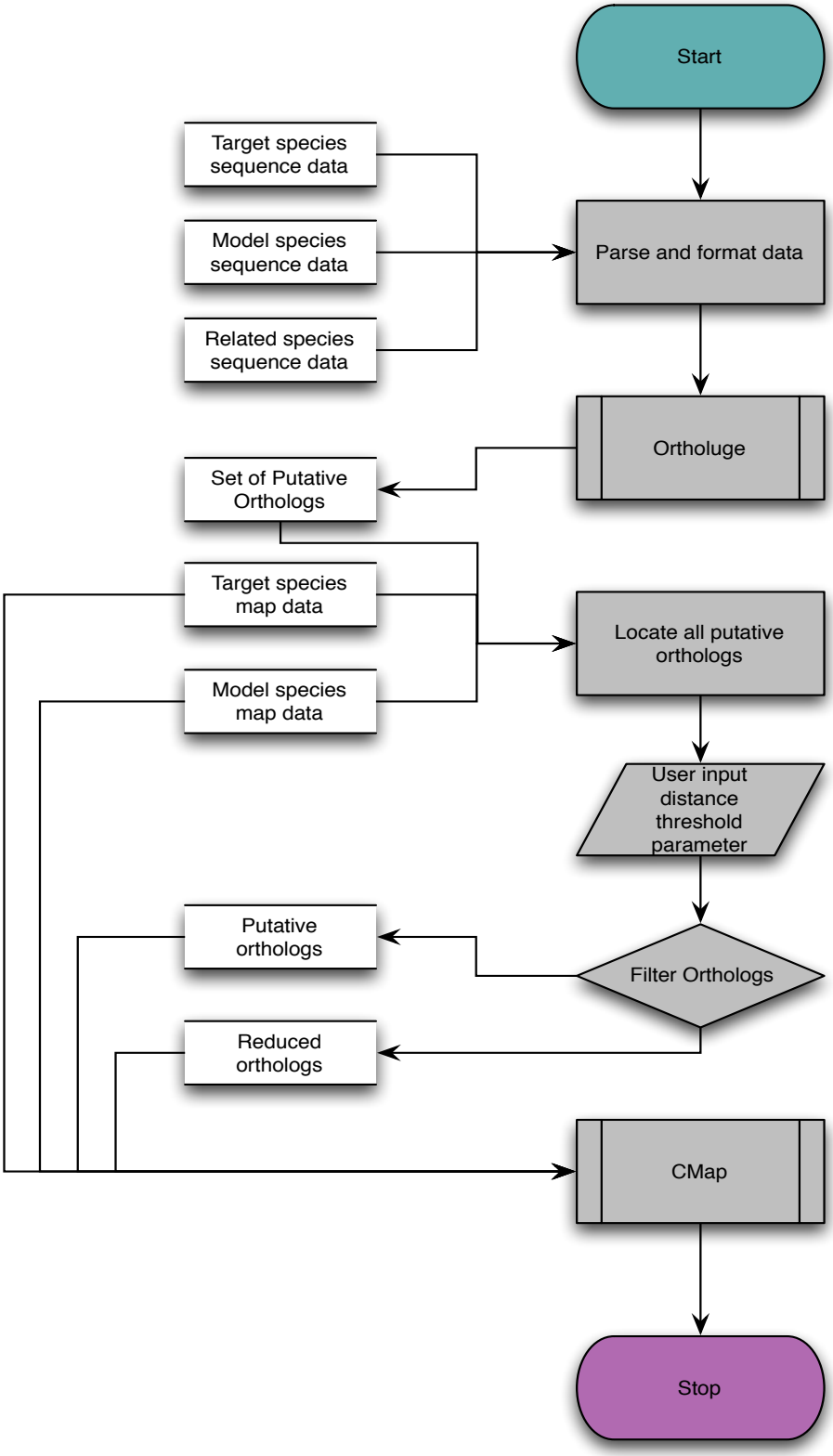

Figure 3.3: Data flow chart describing the DOGMA pipeline.

Customized Perl scripts are used for this task.

The pipeline makes use of five data sets from three related species:

- the target species
  - genetic map data
  - sequence data
- model species
  - genetic map data
  - sequence data
- related species
  - sequence data

The target species is the species of interest to the user. It is not necessary that the data sets be complete for the target species. The model species is a species for which there is an extensive knowledge base and should have complete sequence data available for the highest accuracy. The third species, a slightly more distantly related species, is used as an aid to the ortholog detection program as illustrated in Figure 2.10. The third species is not displayed among the sets of putative or reduced orthologs, but is used as the 'out-group' to enhance the ortholog detection as described in Section 2.4. The order of these three inputs therefore depends on the phylogenetic data as well.

Sequence data of genetic markers for each of the species is necessary as is genetic mapping data for the target and model species. Further, the genetic mapping data must contain some information by which it may be associated with the sequence data for a given species. This is most likely to be the name of the marker in each data set.

The species *Phaseolus vulgaris*, *Glycine max*, and *Medicago truncatula* can be used as the target species, model species and related species, respectively. This is appropriate with the species tree in Figure 2.10, as phylogenetically, *Phaseolus vulgaris* and *Glycine max* are more closely related to each other than *Medicago truncatula* is to either of the former two and thus the three species fit the requirements imposed by the following section. Therefore, we will use these three species for our testing and use them in an ongoing example throughout the rest of this thesis.

3.1.2 Ortholog Detection

Ortholog detection is the first step of the pipeline and is the most critical aspect. Ortholog detection provides the set of putative orthologs which are used throughout the remainder of the pipeline. Therefore, it is essential that it be performed with a robust and accurate system for detecting

putative orthologs. Also, as per the requirements of the pipeline, the detection system must function well in instances with incomplete data sets.

Ortholuge, described in Section 2.4, is used to detect putative orthologs [18]. Ortholuge is a valuable tool, however there are a few caveats. Specifically, when dealing in plant genomics, there are further obstacles to overcome that are not necessarily dealt with in Ortholuge as it stands currently. One such limitation is its inability to effectively analyse polyploid plant species. For example, when searching for orthologs between *Phaseolus vulgaris* (a diploid plant) and *Glycine max* (a tetraploid plant), the results should include two *Glycine max* hits for each *Phaseolus vulgaris* hit. However, only one of these will be found. This limitation is common to RBBH tools. This could be overcome with some customization to the RBBH and phylogenetic analysis portions of the Ortholuge program. However, for this thesis, we overcome this problem with some preprocessing of the species with a higher ploidy value, as discussed in Section 3.2.

In this first step of the pipeline, only the sequence data sets are used. Reiterating Section 2.4, Ortholuge uses three data sets: in-group 1, in-group 2 and, out-group. As applied to the continuing example, we take *Phaseolus vulgaris*, the target species, as in-group 1, *Glycine max* as in-group 2, and *Medicago truncatula* as the out-group. Ortholuge performs a full RBBH between each pair of data sets; that is, it will perform three RBBH searches as follows: in-group 1 against in-group 2, in-group 2 against out-group and, in-group 1 against out-group. Then, Ortholuge finds any putative orthologs shared by each of the three species and then performs a phylogenetic analysis to decide if the putative orthologs fit as expected in the gene tree. If the putative ortholog does not fit the gene tree as expected, such as pictured in the bottom of Figure 2.10, it is discarded. The remaining orthologs are passed to the next step.

3.1.3 Ortholog Filtering and Shared Synteny Detection

The ortholog detection step yields a list of putative orthologs between the target and model species. Using Perl scripts, we join the mapping and sequence data for each species. As per our continued example, we associate the mapping data with the sequence data for *Phaseolus vulgaris* and also do like-wise for *Glycine max*. Then we parse the putative ortholog list and create *correspondences* between the two genetic maps of each species. Pertaining to this thesis, a *correspondence* is defined as a pair of markers, one marker from each of two species, that defines an association of two loci on a visual map as given by the set of putative orthologs. We therefore, name any correspondence by a notation of two markers. When these correspondences are made, we calculate the relative distances between each of the putative orthologs on each linkage group of each species.

We request two parameters, `sp1DistanceParam` and `sp2DistanceParam`, from the user to denote the maximum distance that a marker can be from another and yet possibly fall within the same linkage group– one parameter for each species. These parameters may be given in one of two units,

either centi-Morgans, cM, or in kilo base-pairs, kbp, depending on the types of maps involved, be it genetic maps or physical maps, respectively.

We consider every pair of correspondences. If a correspondence's marker for the first species is less than or equal to the distance parameter for that species from another correspondence's marker and similarly for the second species, then we say that the markers fall within a block of shared synteny and are putative orthologs. We refer to correspondences which do not meet those criteria as reduced orthologs.

Consider the hypothetical maps shown in Figure 3.4. We create correspondences between two maps based on a set of six potential orthologs. The distance parameter for the Pv map is 15 cM and 12 kbp for the Gm map. The algorithm considers each possible pair of correspondences. For example, consider correspondences {Pv1,Gm4} and {Pv2,Gm5}; then $|Pv1 - Pv2|$ is less than the distance parameter of 15 cM for Pv and like-wise $|Gm4 - Gm5|$ is less than the distance parameter for Gm. Hence, we say that these two correspondences are both putative as they form the same region of shared synteny. Consider now correspondences {Pv3,Gm6} and {Pv4,Gm7}. Then $|Pv3 - Pv4|$ meets the distance parameter requirement; however, $|Gm6 - Gm7|$ does not. Therefore ortholog {Pv4,Gm7} remains classified as a reduced ortholog. If `sp1DistanceParam` was extended 30 cM for Pv and `sp2DistanceParam` to 50 kbp for Gm, then all of the markers in Figure 3.4 would be putative. Each additional correspondence which meets the criteria adjacent to the existing region of shared synteny essentially extends that region. There may be ideal fixed values for these parameters, although we allow them to vary so that different values can be studied.

The general algorithm may be found in Figure 3.5.

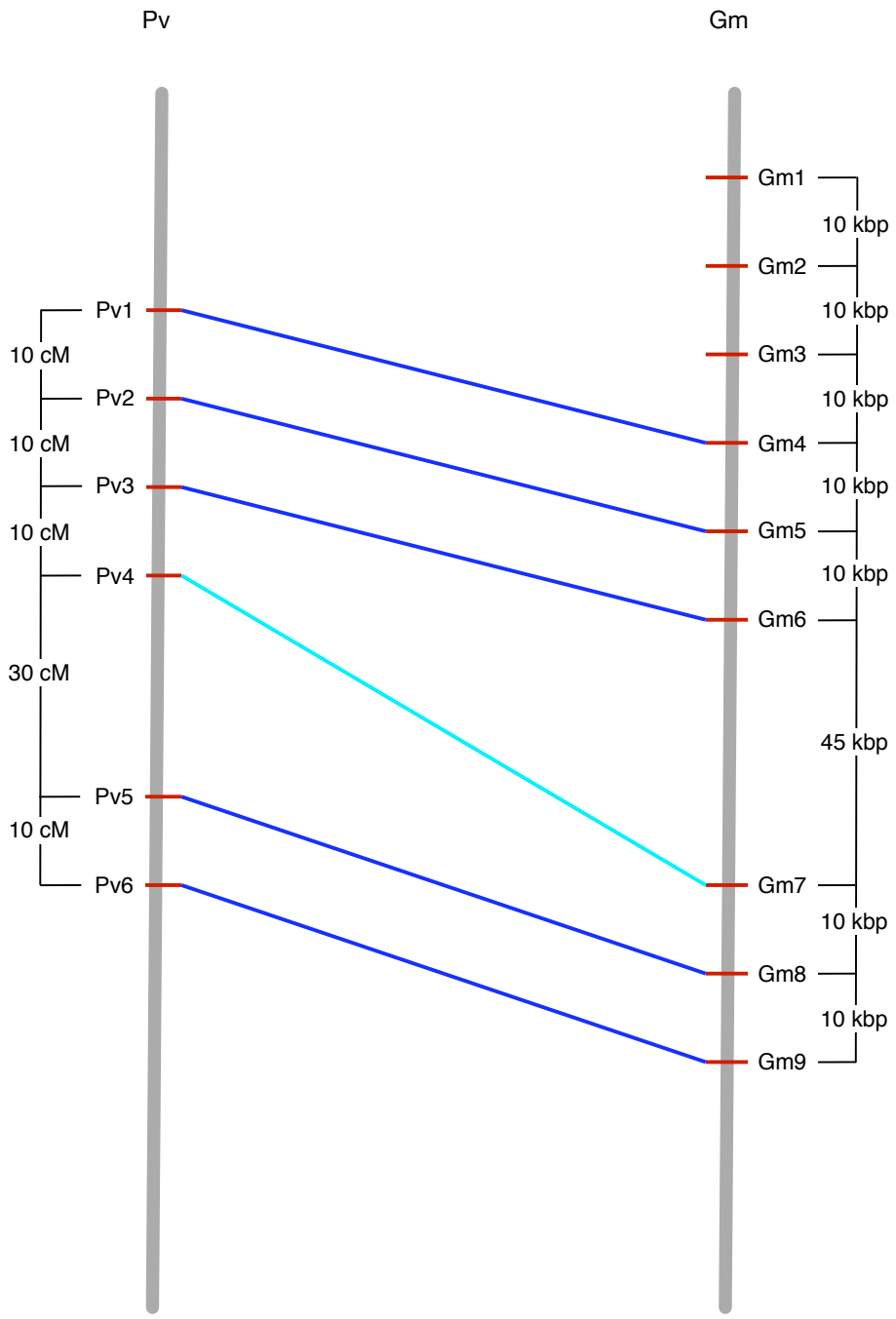

Figure 3.4: Two hypothetical maps to illustrate the filtering of orthologs into a sets of putative and reduced orthologs. The *Phaseolus vulgaris* (Pv) map is displayed with a set of markers, Pv1, ... , Pv6, plotted on it, each separated by the indicated distance measured in centiMorgans. Like-wise the *Glycine max* (Gm) map has its markers, Gm1, ... , Gm6, plotted and separated by the indicated distances measured in kilo base-pairs (kbp). The correspondences in dark blue indicate putative orthologs, correspondences in light blue indicate reduced orthologs.

```
foreach current in @correspondences
do
     foreach next in @correspondences
     do
          if currentCorrespondence != nextCorrespondence
          then
               if |current.sp1Marker.position - next.sp1Marker.position| <= sp1DistanceParam &&
                  |current.sp2Marker.position - next.sp2Marker.position| <= sp2DistanceParam
               then
                    current.putativeFlag = true
                    next.putativeFlag = true
               end if
          end if
     end foreach
end foreach
```

Figure 3.5: Ortholog filtering general algorithm. Variables `current` and `next` are each single elements from the set of correspondence objects. Each correspondence has two marker objects representing a marker from each species pertaining to an ortholog. Each maker object has name and position data. The variables `sp1DistanceParam` and `sp2DistanceParam` are the two input parameters which indicate the maximum desired distance between two orthologs that may belong to the same shared syntenic region.

3.1.4 Map Display

The Comparative Map viewer (CMap) is used to display the maps to the user. CMap uses a mySQL database containing the genetic mapping data for each species. The putative orthologs from the ortholog filtering and shared synteny detection step are incorporated into the CMap database and correspondences between markers, as defined in Section 3.1.3, are created. These correspondences are visual representations of the putative orthologs and are visualized as coloured lines joining markers. Different colours are used to distinguish between putative and reduced orthologs.

Figure 3.6 illustrates putative and reduced orthologs as detected by this pipeline. As shown in the figure, *Glycine max* chromosome 18 was found to have putative orthologs with four separate *Phaseolus vulgaris* linkage groups (1, 6, 8, 11) and regions of shared synteny with three of them (1, 6, 8); the region of shared synteny with linkage group 1 is very small, such that cMap isn't displaying two markers. Correspondences coloured in green are part of a region of shared synteny and are therefore among the set of putative orthologs. Those in red are among the set of reduced orthologs.

3.2 Differences in Ploidy Number

Looking at the example in Section 3.1, we briefly discussed a problem that arose with the choice of species. We note that *Glycine max* and *Phaseolus vulgaris* exist at different ploidy levels. A whole genome duplication has occurred in *Glycine max* causing it to become a tetraploid [36] plant where *Phaseolus vulgaris* remained a diploid species. Intuitively, when we attempt ortholog detection, if both are present, we would assume to find two *Glycine max* markers for each *Phaseolus vulgaris* marker. This is not the case because of the nature of RBBH; it would only find the top one hit. This is a significant limitation, especially with regards to joining the maps of two species as we will not be certain to find regions of shared synteny with any one of two putative orthologs. For example, consider Figure 3.7. The first panel illustrates the correct case in which all putative orthologs are located and regions of shared synteny on all linkage groups are correctly displayed. However, the second panel shows what can occur when RBBH is used to detect orthologs. RBBH overlooks one ortholog from each pair resulting in an incomplete, or possibly missing, region of shared synteny.

In order to overcome this limitation, we preprocess some of the data. In the example from Section 3.1, we would use a BLAST search of *Glycine max* against itself to find putative duplicate genes. We then separate the genes into two files, each file containing one of each copied pair. Then each file is passed through the ortholog detection system separately. Resulting from this will be two sets of putative orthologs which are concatenated together. In this way we essentially by-pass the limitation and are able to continue analyzing regardless of the ploidy level. This method should be modifiable for various polyploid circumstances.

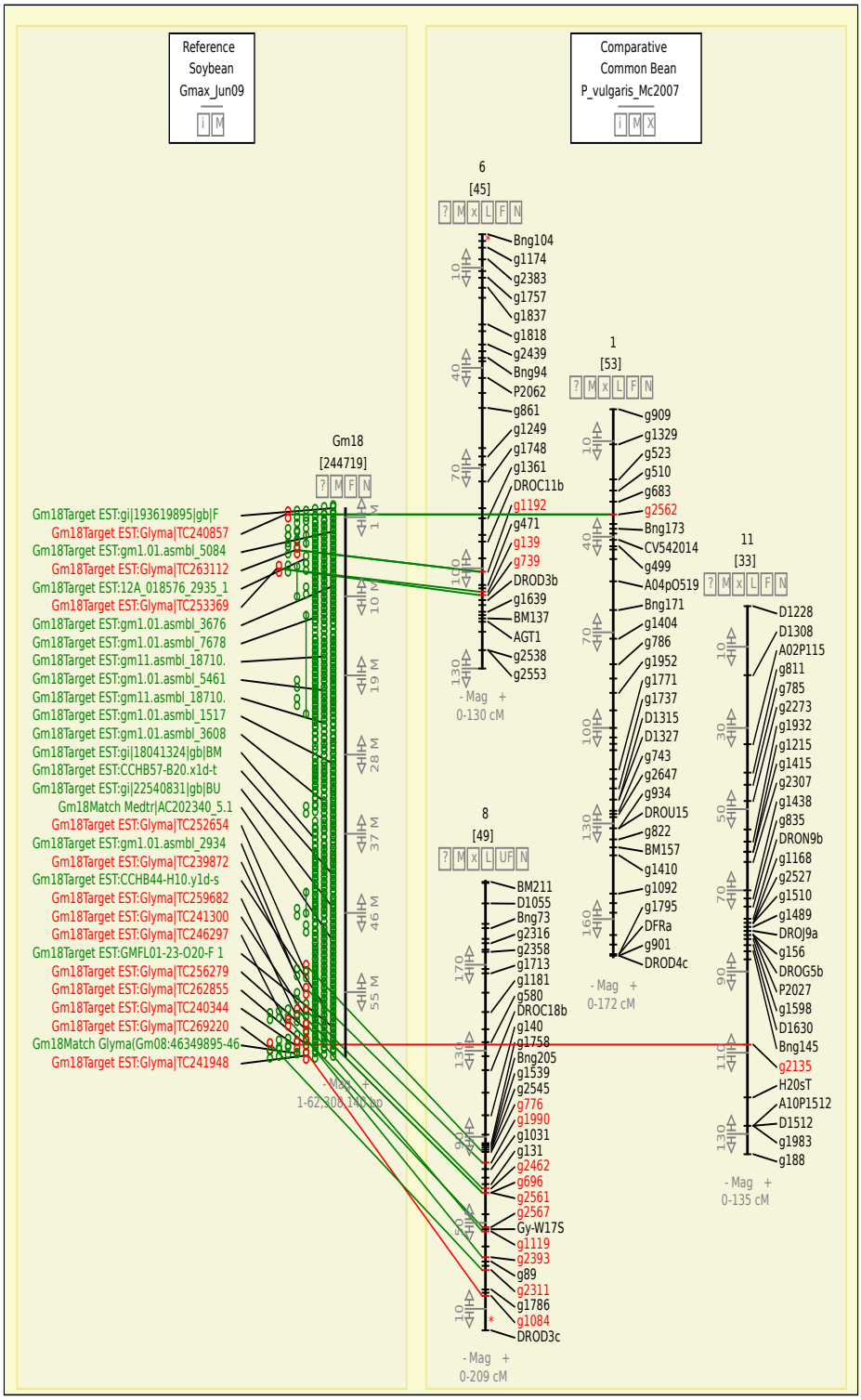

Figure 3.6: Orthologs and shared synteny as displayed by CMap. A single chromosome (number 18) from the species *Glycine max* shares markers with loci on four separate *Phaseolus vulgaris* linkage groups (1, 6, 8, and 11). The red lines indicate reduced orthologs, the green indicates that the markers are in regions of shared synteny. What appears as a solitary green line, such as that between Gm18 and 1, is actually two or more orthologs that are simply too near for the mapping software to illustrate both at this scale. Data sources from [32], [2].

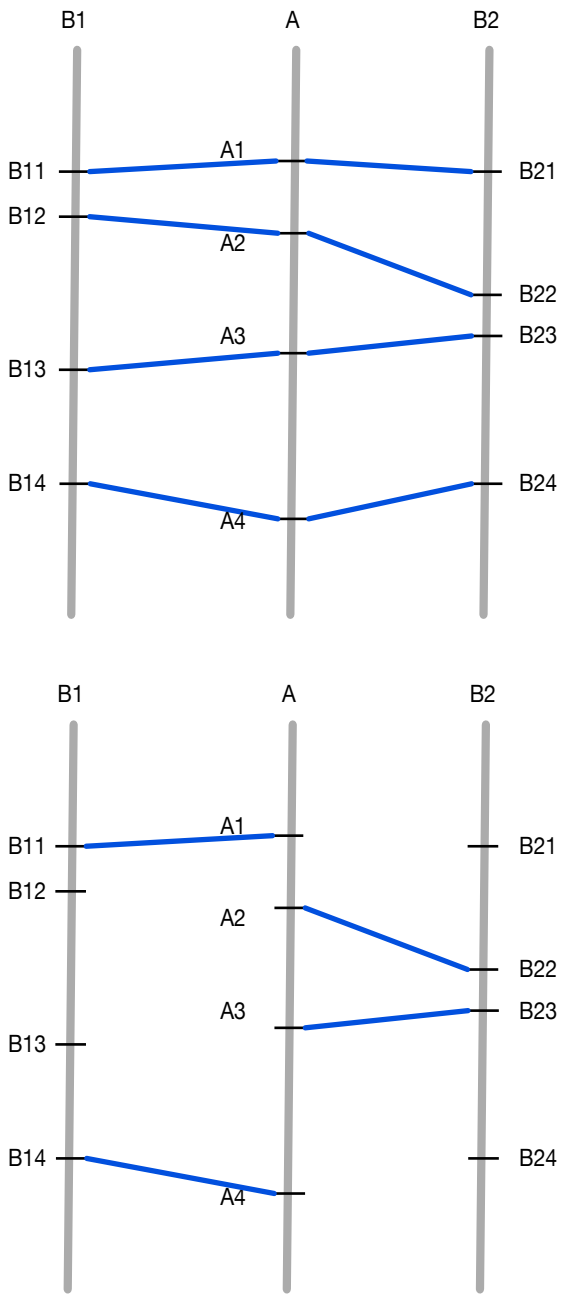

Figure 3.7: Diagram showing putative orthologs between species A, diploid, and species B, tetraploid. The first panel shows the correct case in which for each gene in A two corresponding genes in B are found. The second panel shows what can occur if standard RBBH techniques are applied; only one of each pair will be found, hampering the algorithms ability to detect full regions of shared synteny.

3.2.1 Modularity

The pipeline's modular design allows for the use of various existing programs as well as the ability to easily incorporate customizable scripts. For example, should the user wish to use a different ortholog detection program, it is as simple as removing Ortholuge, replacing it with some other program, and making some minor adjustments to the data files to follow the formats required by the new program. Further, it is relatively easy to update portions of a pipeline when they become out of date; that is, if a new version of one of the modules in the pipeline becomes available, it can be inserted. An unfortunate disadvantage of this type of setup is portability, as it could be difficult to implement on a differing setup, while the custom code portions of the pipeline are easily portable, not all the third party components are easily implemented on all platforms. While there is no clear solution to this issue, it is possible, with good software design principles and portable languages, to create a system which should be relatively portable for anyone with an understanding of the target platform and the pipeline's design.

Pipelined systems, such as this one, are designed for data flow– in situations where data is manipulated in different ways sequentially– a pipeline is advantageous. It is a simple operation to run the entire pipeline or only portions of it as necessary to access the desired data.

Chapter 4

Data Selection, Results, and Discussion

4.1 Data Selection Tool

In Subsection 3.1.3 we refer to two parameters of the algorithm for detecting syntenic correspondences, `sp1DistanceParam` and `sp2DistanceParam`. These two parameters limit the distance between two correspondences to remain classified as a single syntenic region. There is a difficulty presented to the user of DOGMA; what ought these values to be?

In order to assist the user in choosing these values, a Perl script (see Appendix A.4) was written to run the algorithm many times while varying the values of each parameter and to count the number of detected syntenic correspondences at each value of the two parameters. That is, starting at a small value for each parameter and increasing the size each growing until reaching a maximum size which is the total length of their respective maps. The result is a large matrix, where each cell contains the number of syntenic correspondences detected at the coordinates denoted by the two parameters.

Such a matrix is (usually) too large to be read efficiently by human eyes. Therefore, we use the matrix as an input into a spreadsheet program, such as Microsoft's Excel. We can then use the program's charting tools to create a visual representation of the matrix; for example see Figure 4.1. Using a 3-dimensional chart, the user can decide more precisely what distance parameters to use. The tool can easily be re-run with a new range of distance parameters to create a new chart, thus allowing the user to narrow, or expand their parameters as necessary. The distance parameters chosen by the user will depend on how stringent they wish the final output to be. That is, if the user wishes tight stringency in order to be very precise they will look to the lower end of the scales.

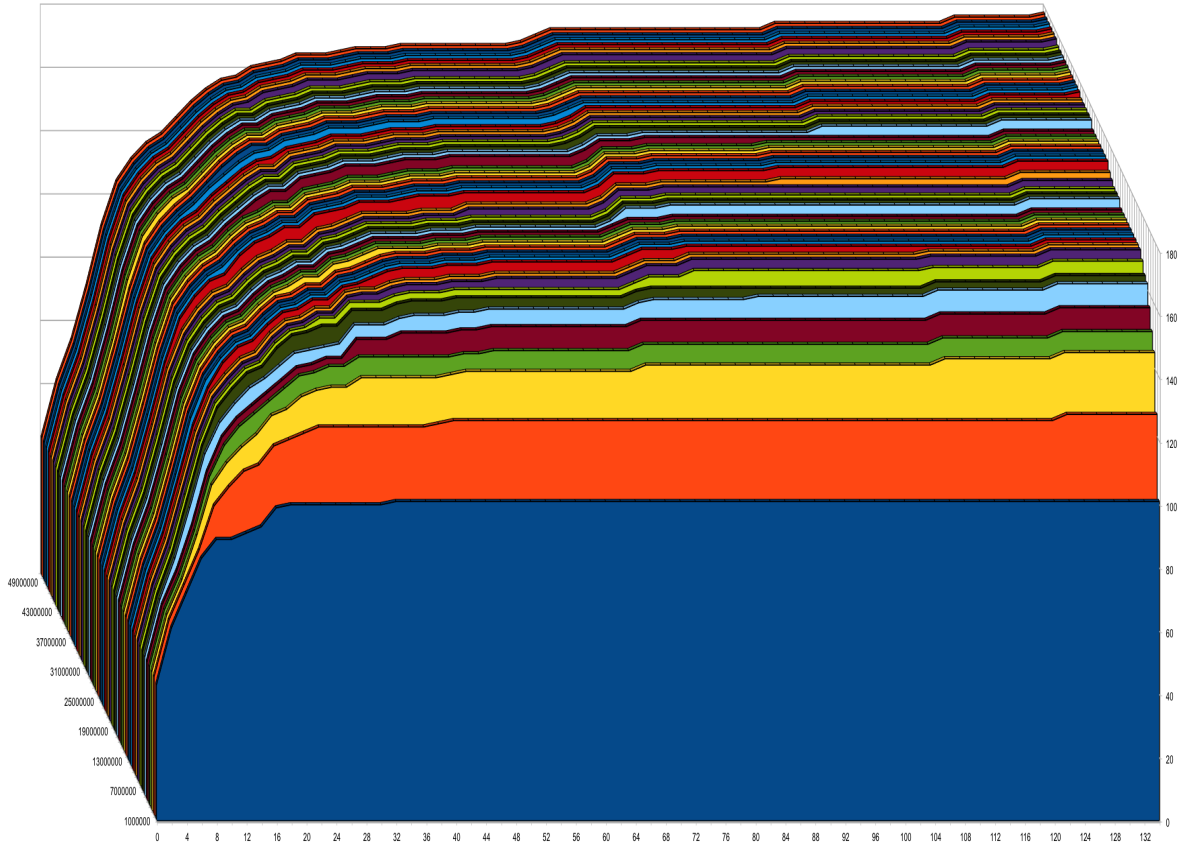

Figure 4.1: `sp1DistanceParam` on x-axis, `sp2DistanceParam` on z-axis, and the number of orthologs as detected by the program on the y-axis. As each distance parameter increases as does the number of probable orthologs. This indicates that there is little change in syntenic correspondence detection beyond a certain distance parameter value.

4.2 Data Sources

The design of this pipeline allows it to function over a variety of related species and is not limited to the example species. For this example we use the previously discussed species since they satisfy the criteria listed in Section 3.1.1 as being appropriate.

As noted in Section 3.1.2 three sets of sequence data are required for Ortholuge to perform the initial ortholog detection. Sequence data for the target species *Phaseolus vulgaris* was acquired through personal correspondence with Dr. Phil McClean from North Dakota State University [32]. Sequence data for the related species *Glycine max* was acquired from the *G. max* Gene Index as hosted at Harvard University [2]. Sequence data for the model species *Medicago truncatula* was acquired from the *M. truncatula* Gene Index and also hosted on the Harvard University servers [1]. The Mapping data for both *P. vulgaris* and *G. max* was acquired from the Legume Information System website [3].

4.3 Test Run and Results

At the first stage, sequence and mapping data were organized and then analysed for correlations. Extraneous metadata was removed from the FASTA sequence headers such that all that remained was the sequence tag identifier. Where sequence data was not found to match any existing markers, those sequences were removed as they can potentially interfere with locating mappable orthologs; orthologs may be detected between sequences which are not mappable due to lack of mapping data. Also reducing the overall number of sequences has an additional positive effect on performance. There was no formatting required for the mapping data as it was retrieved from one CMap environment and was inputted into a local CMap database. In future iterations of DOGMA, the sequences that were removed will be used and and if possible their approximate loci will be estimated based on the surrounding regions of shared synteny, see Section 5.3.

At the second stage, Ortholuge was run with *Phaseolus vulgaris* as the target species, *Glycine max* as the related species, and *Medicago truncatula* as the more distantly related species. The results from Ortholuge yielded 205 potential orthologs.

For the third stage the sequence data and map data for each species was parsed using the PERL script seq_to_map.pl, Appendix A.1, to associate the map and sequence for use in the next stages of the pipeline. At this point the data selection tool as described in Section 4.1 was used to choose the two distance parameters for the Ortholog filtering stage. Looking at the chart, Figure 4.1, we selected the two parameters as 7 000 000pb for `sp1DistanceParam` and 10cM for `sp2DistanceParam`, as this was the point in the graph where there was a drop in the rate of increase of detected orthologs.

Ortholog filtering continued, using the synteny detection script find_synteny.pl, Appendix A.3. Based on the two distance parameters, the 205 orthologs detected by Ortholuge were recategorized as 121 putative orthologs and 84 reduced orthologs. The 121 putative orthologs are listed in Table 4.1 and 84 reduced orthologs in Table 4.2. That is, based on the given data, 84 of the orthologs detected are not in a region of shared synteny denoted by the distance parameters used and therefore have insufficient mapping data to elevate them to the category of putative ortholog.

Table 4.1: Set of 121 putative orthologs as detected by DOGMA using distance parameters of 7 000 000bp and 10cM.

***Glycine max* marker**	***Phaseolus vulgaris* marker**
Gm08Target EST:Glyma—TC258604	g1676
Gm08Target EST:Glyma—TC271959	g2512
Gm08Target EST:Glyma—TC246686	g2596
Gm08Target EST:Glyma—TC239534	g2596
Gm08Target EST:Glyma—TC242261	g1084
Gm08Target EST:Glyma—TC238553	g1786
Gm08Target EST:Glyma—TC271072	g1786
Gm08Target EST:Glyma—TC246110	g1341
Gm08Target EST:Glyma—TC247718	g2260
Gm03Target EST:Glyma—TC243220	g1771
Gm03Target EST:Glyma—TC240408	g1176
Gm10Target EST:Glyma—TC264168	g487
Gm10Target EST:Glyma—TC242831	g1378
Gm10Target EST:Glyma—TC240126	g1233
Gm10Target EST:Glyma—TC250393	g501
Gm10Target EST:Glyma—TC242839	g501
Gm10Target EST:Glyma—TC238177	g1615
Gm01Target EST:Glyma—TC257584	g1801
Gm01Target EST:Glyma—TC244157	g1556
Gm12Target EST:Glyma—TC236451	g2273
Gm12Target EST:Glyma—TC253434	g1932
Gm05Target EST:Glyma—TC260334	D1367
Gm05Target EST:Glyma—TC242764	g2127
Gm05Target EST:Glyma—TC240489	g693
Gm05Target EST:Glyma—TC257721	g774

Continued on next page

Table 4.1 – continued from previous page

Glycine max marker	*Phaseolus vulgaris marker*
Gm05Target EST:Glyma—TC244534	g2540
Gm05Target EST:Glyma—TC240854	g2348
Gm16Target EST:gi—6070924—gb—AW1	g2221
Gm16Target EST:Glyma—TC246331	g2221
Gm14Target EST:Glyma—TC247400	g2061
Gm14Target EST:Glyma—TC241545	g2061
Gm14Target EST:Glyma—TC235394	g2151
Gm14Target EST:Glyma—TC238798	g2413
Gm14Target EST:Glyma—TC257557	g2413
Gm14Target EST:Glyma—TC248427	g1858
Gm14Target EST:Glyma—TC243814	g1713
Gm15Target EST:Glyma—TC241388	g2512
Gm15Target EST:Glyma—TC237789	g1188
Gm15Target EST:Glyma—TC241487	g1188
Gm15Target EST:Glyma—TC252381	g1818
Gm15Target EST:Glyma—TC249027	g2208
Gm15Target EST:Glyma—TC251001	g2439
Gm15Target EST:Glyma—TC241639	g2329
Gm15Target EST:gi—33387671—gb—CA	g1748
Gm15Target EST:Glyma—TC249453	g1852
Gm20Target EST:Glyma—TC251566	g1380
Gm20Target EST:Glyma—TC237732	g1853
Gm20Target EST:Glyma—TC238694	g1615
Gm20Target EST:Glyma—TC257099	g2129
Gm20Target EST:Glyma—TC238296	g2129
Gm20Target EST:Glyma—TC257392	g2531
Gm20Target EST:Glyma—TC239500	g2531
Gm18Target EST:Glyma—TC240857	g2562
Gm18Target EST:Glyma—TC238849	g2562
Gm18Target EST:Glyma—TC263112	g1192
Gm18Target EST:Glyma—TC244953	g1192
Gm18Target EST:Glyma—TC253369	g139
Gm18Target EST:Glyma—TC259162	g739

Continued on next page

Table 4.1 – continued from previous page

Glycine max marker	*Phaseolus vulgaris marker*
Gm18Target EST:Glyma—TC252654	g776
Gm18Target EST:Glyma—TC239872	g1990
Gm18Target EST:Glyma—TC259682	g2561
Gm18Target EST:Glyma—TC271399	g696
Gm18Target EST:Glyma—TC241300	g2462
Gm18Target EST:Glyma—TC246297	g2567
Gm18Target EST:Glyma—TC256279	g1119
Gm18Target EST:Glyma—TC262855	g2393
Gm18Target EST:Glyma—TC269220	g2311
Gm13Target EST:Glyma—TC250879	g1968
Gm13Target EST:Glyma—TC235306	g2557
Gm13Target EST:Glyma—TC247980	g1689
Gm13Target EST:Glyma—TC238619	g2308
Gm13Target EST:Glyma—TC237323	g1664
Gm13Target EST:Glyma—TC257531	g2208
Gm13Target EST:Glyma—TC252730	g1818
Gm13Target EST:Glyma—TC259551	D1861
Gm13Target EST:Glyma—TC251279	D1861
Gm04Target EST:Glyma—TC257987	g2510
Gm04Target EST:Glyma—TC253979	g1126
Gm04Target EST:Glyma—TC263290	g708
Gm06Target EST:Glyma—TC235260	g732
Gm06Target EST:Glyma—TC268509	g2178
Gm06Target EST:Glyma—TC259497	g1884
Gm06Target EST:Glyma—TC252821	g1107
Gm06Target EST:Glyma—TC242077	g1107
Gm06Target EST:Glyma—TC242688	g1126
Gm06Target EST:Glyma—TC242758	g2510
Gm07Target EST:Glyma—TC242299	g2113
Gm07Target EST:gi—151395830—gb—E	g893
Gm07Target EST:Glyma—TC241657	g2268
Gm07Target EST:Glyma—CD414857	g2268
Gm07Target EST:Glyma—TC254896	g2260

Continued on next page

Table 4.1 – continued from previous page

Glycine max marker	*Phaseolus vulgaris marker*
Gm07Target EST:Glyma—TC237248	g1341
Gm09Target EST:Glyma—TC238927	g2581
Gm09Target EST:Glyma—TC241053	g2581
Gm09Target EST:Glyma—TC246807	g2561
Gm09Target EST:Glyma—TC246582	g549
Gm11Target EST:Glyma—TC271437	g2020
Gm11Target EST:gi—5509449—gb—AI8	g797
Gm11Target EST:Glyma—TC277352	g785
Gm11Target EST:Glyma—TC236636	g2273
Gm11Target EST:Glyma—TC249799	g1932
Gm11Target EST:Glyma—TC235231	g2285
Gm11Target EST:Glyma—TC269102	g835
Gm02Target EST:Glyma—TC235255	g2218
Gm02Target EST:gi—23725617—gb—BU	g2218
Gm02Target EST:Glyma—TC255985	g2108
Gm02Target EST:Glyma—TC253293	g1830
Gm02Target EST:Glyma—TC247422	g1925
Gm02Target EST:gi—14009149—gb—BG	g1925
Gm17Target EST:Glyma—TC248658	g2145
Gm17Target EST:Glyma—TC253451	g1645
Gm17Target EST:Glyma—TC243679	g1795
Gm17Target EST:Glyma—TC257905	g2371
Gm17Target EST:Glyma—TC245318	g2371
Gm17Target EST:Glyma—TC236206	g1808
Gm17Target EST:Glyma—TC246873	g2341
Gm17Target EST:Glyma—TC260022	g665
Gm19Target EST:Glyma—TC250348	g1954
Gm19Target EST:Glyma—TC239929	g1176
Gm19Target EST:Glyma—TC250305	g1361
Gm19Target EST:Glyma—TC265797	g1361

Table 4.2: Set of reduced orthologs as detected by DOGMA using distance parameters of 7 000 000bp and 10cM.

***Glycine max* marker**	***Phaseolus vulgaris* marker**
Gm20Target EST:Glyma—TC238249	g1175
Gm19Target EST:Glyma—TC261813	g2303
Gm19Target EST:Glyma—TC264450	g1758
Gm20Target EST:gi—26060100—gb—CA	g2068
Gm19Target EST:Glyma—TC242377	g1786
Gm19Target EST:Glyma—TC250993	g743
Gm01Target EST:Glyma—TC248134	g797
Gm01Target EST:Glyma—TC250804	g1686
Gm02Target EST:Glyma—TC236183	g2274
Gm02Target EST:Glyma—TC244236	g1466
Gm02Target EST:Glyma—TC259352	g2135
Gm02Target EST:Glyma—TC265722	g1858
Gm02Target EST:Glyma—TC276004	g1181
Gm03Target EST:gi—5606222—gb—AI9	g743
Gm03Target EST:Glyma—TC235297	D1580
Gm03Target EST:Glyma—TC237691	g2521
Gm04Target EST:Glyma—TC237156	g544
Gm04Target EST:Glyma—TC243240	g993
Gm05Target EST:gi—151412425—gb—E	g1148
Gm05Target EST:Glyma—TC240679	D1132
Gm05Target EST:Glyma—TC244308	g2341
Gm05Target EST:Glyma—TC248661	g849
Gm05Target EST:Glyma—TC249924	g1808
Gm05Target EST:Glyma—TC259288	g2493
Gm05Target EST:Glyma—TC275291	g1247
Gm06Target EST:Glyma—TC249871	g792
Gm06Target EST:Glyma—TC251150	g1168
Gm07Target EST:Glyma—TC238494	g2521
Gm07Target EST:Glyma—TC242168	g1181
Gm07Target EST:Glyma—TC251071	g2427
Gm07Target EST:Glyma—TC251855	g1994

Continued on next page

Table 4.2 – continued from previous page

***Glycine max* marker**	***Phaseolus vulgaris marker***
Gm07Target EST:Glyma—TC259557	g2068
Gm07Target EST:Glyma—TC260945	g2192
Gm07Target EST:Glyma—TC269182	g776
Gm08Target EST:gi—14991863—gb—BI	g1148
Gm08Target EST:Glyma—TC237131	g1247
Gm08Target EST:Glyma—TC237798	g2113
Gm08Target EST:Glyma—TC239687	g739
Gm08Target EST:Glyma—TC241542	g2348
Gm08Target EST:Glyma—TC243963	g1190
Gm08Target EST:Glyma—TC244401	g909
Gm08Target EST:Glyma—TC250898	g1656
Gm08Target EST:Glyma—TC251632	g2393
Gm08Target EST:Glyma—TC264982	g634
Gm08Target EST:Glyma—TC275312	g1664
Gm09Target EST:Glyma—TC236555	g1159
Gm09Target EST:Glyma—TC240476	g2595
Gm09Target EST:Glyma—TC241216	g1564
Gm09Target EST:Glyma—TC242605	g1852
Gm09Target EST:Glyma—TC255998	g1119
Gm10Target EST:Glyma—TC244326	g1853
Gm11Target EST:Glyma—TC241704	D1367
Gm11Target EST:Glyma—TC243555	g1361
Gm11Target EST:Glyma—TC263523	g511
Gm12Target EST:Glyma—TC237088	g909
Gm12Target EST:Glyma—TC239301	g1395
Gm12Target EST:Glyma—TC251402	g634
Gm12Target EST:Glyma—TC254133	g2285
Gm12Target EST:Glyma—TC259600	g1719
Gm13Target EST:gi—20449746—gb—BQ	D1086
Gm13Target EST:Glyma—TC239765	g417
Gm13Target EST:Glyma—TC240709	g2416
Gm13Target EST:Glyma—TC243861	g1065
Gm13Target EST:Glyma—TC247255	g1341

Continued on next page

Table 4.2 – continued from previous page

Glycine max marker	*Phaseolus vulgaris marker*
Gm13Target EST:Glyma—TC250986	g2551
Gm13Target EST:Glyma—TC254942	g1175
Gm14Target EST:Glyma—TC237651	g2274
Gm14Target EST:Glyma—TC247866	g1645
Gm14Target EST:Glyma—TC261835	g1224
Gm15Target EST:Glyma—TC249996	g2557
Gm15Target EST:Glyma—TC256032	g860
Gm15Target EST:Glyma—TC258624	g1983
Gm16Target EST:Glyma—TC239118	g1758
Gm16Target EST:Glyma—TC240301	g2558
Gm16Target EST:Glyma—TC251574	g2647
Gm17Target EST:Glyma—TC236680	g544
Gm17Target EST:Glyma—TC238919	g1175
Gm17Target EST:Glyma—TC240094	g822
Gm17Target EST:Glyma—TC276082	g2020
Gm18Target EST:Glyma—TC240344	g2135
Gm18Target EST:Glyma—TC241948	g1084
Gm19Target EST:Glyma—TC237422	g1404
Gm19Target EST:Glyma—TC238511	g2467
Gm20Target EST:Glyma—TC244314	g2538

4.4 Discussion

We achieved success in finding several areas of shared synteny in our example described in Section 4.3 and three of these regions are shown in Figure 4.2. As can be seen in each of the panels, using the distance parameters previously specified (10cM for the *Phaseolus vulgaris* linkage map and 7000000bp for the *Glycine Max* physical map) the DOGMA pipeline detected regions with four to five putative orthologs. It is possible that a researcher may infer that all of the remaining sequences on the *Glycine max* genome could and should be found in their respective order on the *Phaseolus vulgaris* map, provided that the distance parameters are stringent enough.

Markers along the *Phaseolus vulgaris* map in black have no detected orthologs. This is due either to the presence of mapping data without sequence data or the resulting data being rejected by Ortholuge in the first step of DOGMA. Examples of these can also be seen in Figure 4.2. Had these sequences been present, the regions of shared synteny may have been even stronger or larger.

Additional examination of the results show multiple regions of shared synteny from one linkage group of *Phaseolus vulgaris* to two chromosomes of *Glycine max*; see Figure 4.3. This is not unexpected as *Glycine max* is an ancient polyploid, and has approximately 2 copies of the *Phaseolus vulgaris* genome in a somewhat scrambled fashion [36].

In a few cases, we notice that DOGMA does not require colinearity between markers, an example of which can be seen in Figure 4.4. Though a marker pair falls within the designated distance parameters, we can see that it is not collinear within its region of shared synteny. DOGMA is checking that consecutive markers are within a certain parameterized distance (see Section 3.1.3), however in this example, the markers are within the parameterized distance but are not collinear. It is possible that these are simple translocations of the ortholog in one species or the other and still warrant the classification of putative ortholog. Future iterations of DOGMA will attempt to account for these and possibly highlight them for manual review.

Of the 84 reduced orthologs, most of them were simply beyond the distance parameters set and were placed in the category of reduced orthologs. Interestingly, a few of them, such as the one in Figure 4.3, were seen as a single marker on the *Phaseolus vulgaris* map which has both a reduced ortholog and a putative ortholog. This is not unexpected because, as noted in Section 3.2, *Glycine max* has double the ploidy level of *Phaseolus vulgaris*. It would be expected to find two *Glycine max* markers for each *Phaseolus marker*. These, however were not detected by Ortholuge in the initial step and were therefore not detected in the latter steps of DOGMA.

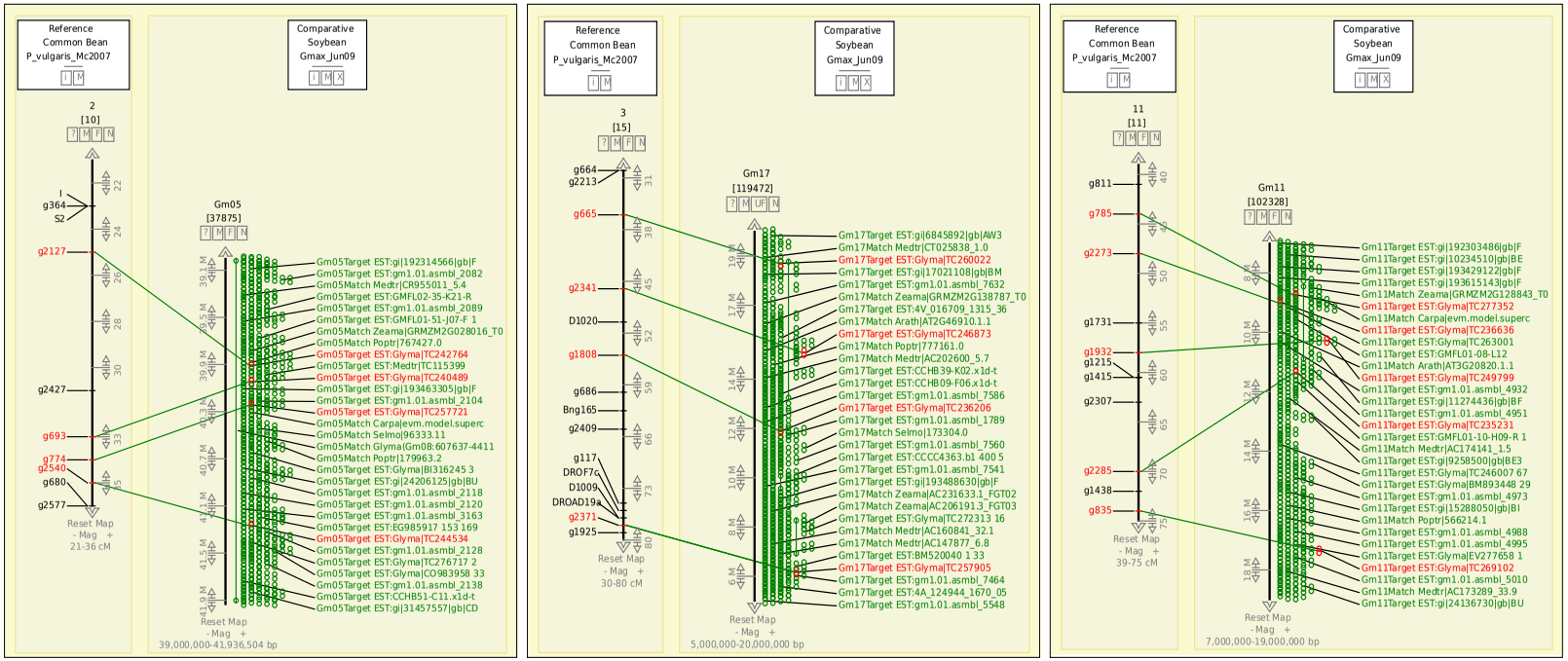

Figure 4.2: Three regions of shared synteny as detected by DOGMA. Each region determined by the distance parameters of 10cM for the *Phaseolus vulgaris* map and 7 000 000bp for the *Glycine max* map. Each map has been enlarged to more clearly show the region of shared synteny. The panels from left to right show portions of linkage group 2, 3, and 5 of *Phaseolus vulgaris* and chromosomes 5, 17, and 11 of *Glycine max*, respectively, with a regions of shared synteny marked by four to five putative orthologs each.

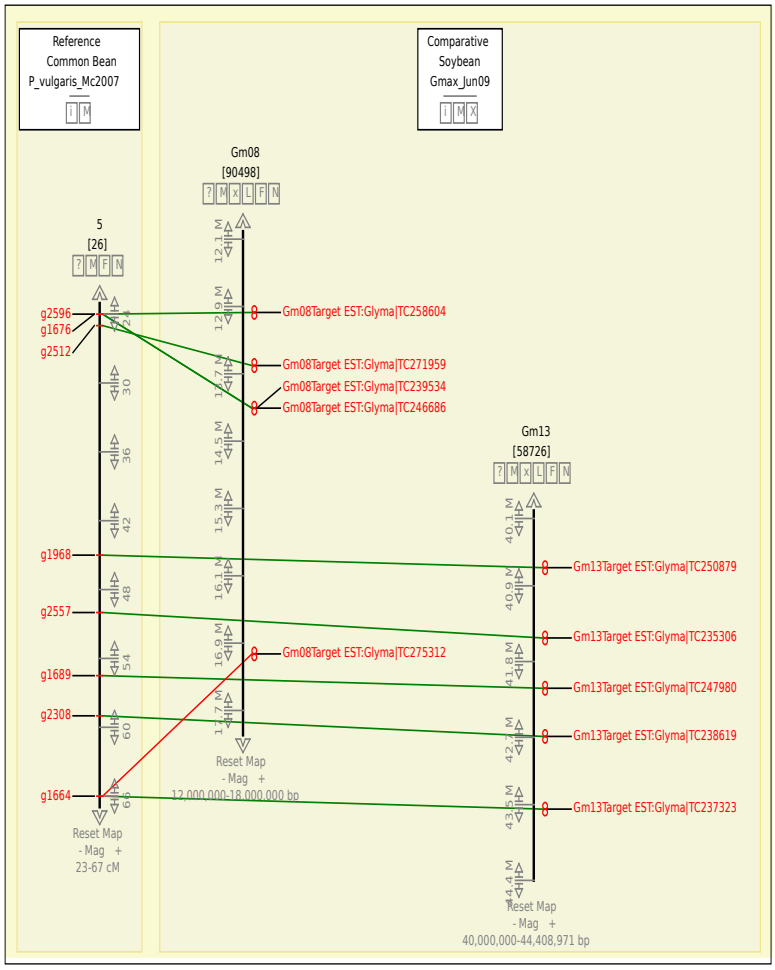

Figure 4.3: DOGMA detected two regions of shared synteny on linkage group 5 of *Phaseolus vulgaris* to chromosomes 8 and 13 of *Glycine max*. Also shown is one marker on the *Phaseolus vulgaris* map with one putative ortholog to chromosome 13 and one reduced ortholog (red) to chromosome 8 of *Glycine max*. Markers with no detected orthologs were removed from this view for clarity of viewing.

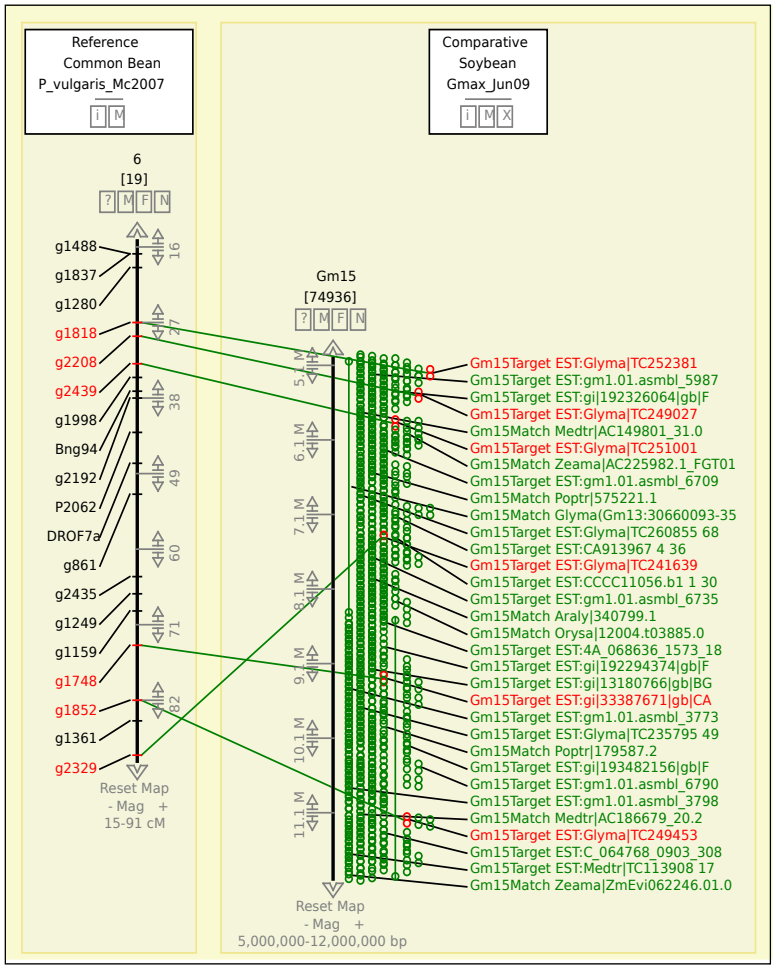

Figure 4.4: DOGMA detected two regions of shared synteny on linkage group 6 of *Phaseolus vulgaris* to chromosome 15 of *Glycine max*. DOGMA detects a non-colinear region as a region of shared synteny.

4.5 Validation

At the time of development of DOGMA, *Phaseolus vulgaris* had not been completely sequenced. It has very recently been sequenced as a "preview release" [5]. As such we are able to use it to more accurately validate the results from DOGMA.

The validation method used to assure the correctness of DOGMA was to use the same species as in our initial test runs (see Section 4.3) but this time using the completed map and sequence data for *Phaseolus vulgaris*. As a control, we perform ortholog detection using DOGMA on the entire set of data. Then we randomly remove 1000 orthologous sequences of the target species, and gauge DOGMA's performance on the results using this reduced set as compared to the results of the control test. This allows us to demonstrate DOGMA's efficacy on incomplete data sets. We use this

validation method to show DOGMA's accuracy on a complete data set and its ability to maintain that accuracy on a known partial data set. Further, we also remove an additional variable by staying within the same type of map; that is, we compare results from a physical mapping control test against a physical mapping validation test, rather than introduce an additional unknown variable by comparing against a genetic map such as the one used in Section 4.3.

Both the control and validation runs were performed using distance parameters of 1 000 000bp for both species. A sample of the control and validation test runs is shown in Figure 4.5.

If we assume that Ortholuge's results on the complete data set are the 'correct results' (we will discuss this assumption below), we can discuss DOGMA's sensitivity and specificity. We take the number of true positives (TP) to be the number of putative orthologs detected by DOGMA that are also in the set of orthologs detected by Ortholuge. In the control run that number is 13 132. We take the number of false negatives (FN) to be the number of reduced orthologs detected by DOGMA which were also in the set of orthologs detected by Ortholuge. In the control run that is 681. By its very design, DOGMA cannot detect more orthologs than Ortholuge, so there cannot be any false positives (FP), and since Ortholuge does not report results as being 'not orthologs' there also cannot be any true negatives (TN). We calculate the sensitivity of DOGMA as: $TP/(TP+FN)$ or $13132/(13132+681) = 0.951$ or approximately 95%, suggesting an error rate of approximately 5% compared with Ortholuge. Since we have no information on true negatives or false positives we cannot speak to the specificity of DOGMA.

Using the same description of true positives and false negatives for the validation run we arrive at 12 937 true positives and 668 false negatives. Again calculating for sensitivity: $12937/(12937+668) = 0.951$ or approximately 95%. This indicates that DOGMA is functioning as well on partial data as on complete data.

Though these validation data appear promising, the validation results should be considered inconclusive. Both the control and validation results appear very similar. The structure of the data sets may be causing this result. There is extensive coverage and overlap of sequences along both maps, and with the removal of one putative or reduced ortholog, another was detected from the overlapping sequences. That is, another putative ortholog was found for many of the removed orthologs. After randomly removing orthologs as detected by the control run, there is no appreciable difference in the number of probable orthologs nor reduced orthologs. Having removed 1000 potential orthologs as detected by Ortholuge, we would have expected to see a total number of orthologs in the validation run of approximately 12 800, and instead the total reduction was only 208. This indicates that even after the removal of a series of probable orthologs, the initial detection by Ortholuge did not yield any orthologs that are truly different from the control set. Moreover, there were no confirmed true negatives, nor a reduction of false positives, since all reduced orthologs present on the validation run were present in the orthologs detected by Ortholuge on the complete

set. A test involving multiple systematically constructed reduced data sets is left as future research.

We will briefly discuss the assumption of interpreting the results of Ortholuge as the 'correct results'. We believe that it is possible that the reduced orthologs are not DOGMA's false negatives, but that they could in fact be Ortholuge's false positives based on DOGMA's knowledge of the orthologs' mapping related data. Indeed, the detection by Ortholuge does not take into account the mapping data, and DOGMA has determined that these reduced orthologs do not fall within a region of shared synteny, providing evidence that they are not actual orthologs. Thus, it is possible that Ortholuge is not providing the 'correct results', but that by adding mapping data to the pool of knowledge, DOGMA is in fact an improvement on Ortholuge's sensitivity.

We attempt to prove DOGMA's ability to detect false positives in Ortholuge's results when using partial data sets by artificially creating a paralog in *Glycine max* and then removing the real ortholog. Specifically, we copy the sequence from one known ortholog and artificially mutate it so that it is different from the original, and create a new map marker that is not in the region of shared synteny. We then removed the original ortholog and all of the overlapping sequences at its locus; that is, we create a partial data set where the original ortholog is missing, and we then run DOGMA. The results in Figure 4.6 show us that the one expected reduced ortholog was detected. This indicates that the paralogous sequence we introduced was detected as an ortholog by Ortholuge, but was reclassified by DOGMA because if was outside a region of shared synteny.

By its very definition, DOGMA, is no worse than Ortholuge as it does not eliminate any data that Ortholuge presents, and as shown in the test and validation runs, it was capable of reclassifying the orthologs as detected by Ortholuge into potential and reduced orthologs. Furthermore, in the case where an ortholog is missing from the target species, but a paralog is present, DOGMA can properly classify it as a reduced ortholog. In contrast, it can be improperly classified as an ortholog by Ortholuge on its own.

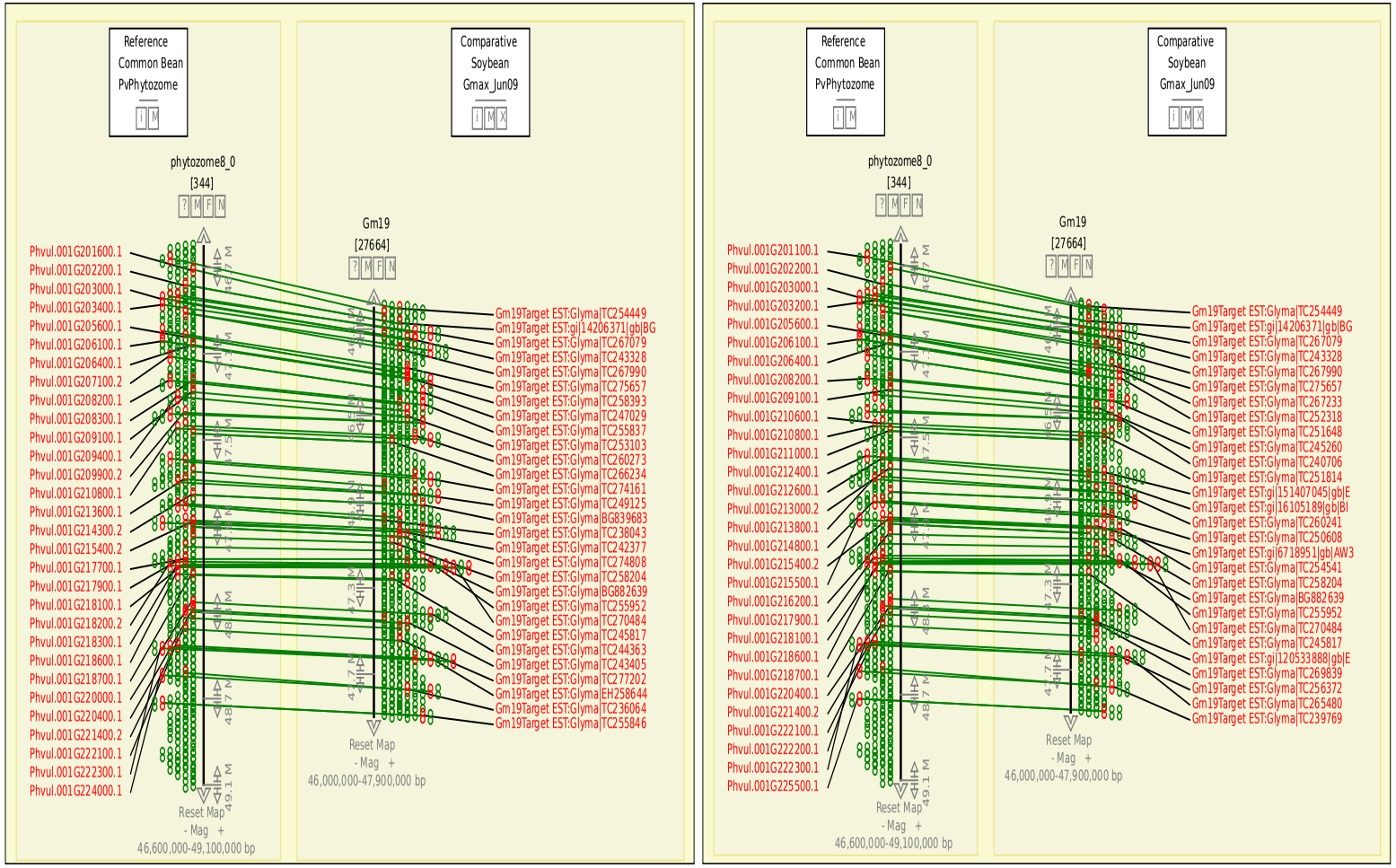

Figure 4.5: Two regions of shared synteny detected via DOGMA using complete data sets. Both maps display the same regions on both species. The left panel illustrates a portion of the resulting map as yielded from a complete set of data. The right panel illustrates the same portion of the map as yielded from a DOGMA run with several ortholog sequences removed.

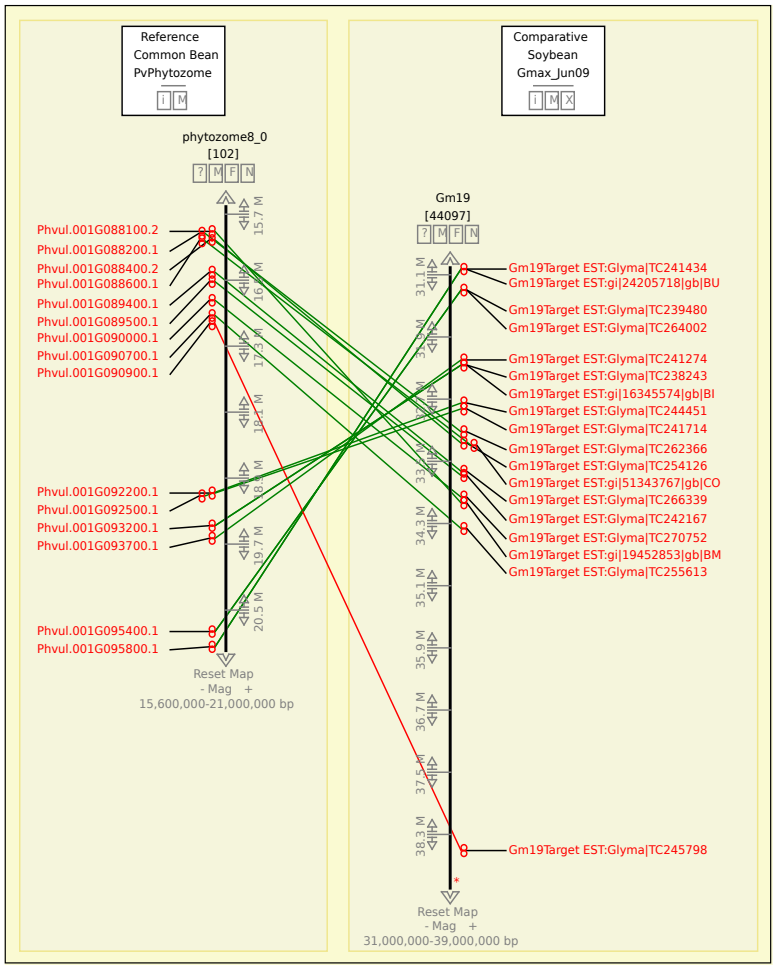

Figure 4.6: DOGMA detects an artificially created paralog as a reduced ortholog. The reduced ortholog, was created by copying the sequence data, mutating it, and creating a new map marker for it outside the region of shared synteny. The original marker was removed from the partial data set.

Chapter 5

Summary and Future Directions

5.1 Summary

5.1.1 Problem

Researchers interested in examining species with limited genetic data and incomplete maps often look to related species that have been completely or nearly completely mapped and or sequenced. While there are several tools that will compare sequence data to find orthologs between two species, they do not use principles of shared synteny to ensure the correctness of their results. In cases of incomplete data, we cannot be sure that a detected ortholog is in fact a true ortholog. Regions of shared synteny can be used to verify the validity of orthologs.

5.1.2 Solution

The pipeline we call DOGMA expands upon the techniques used in other ortholog detection programs by including mapping data to more accurately describe the detected orthologs. DOGMA first uses Ortholuge to detect potential orthologs, then uses mapping data for each species along with user specified distance parameters to filter the potential orthologs into two categories of putative orthologs and reduced orthologs. Putative orthologs have both sequence similarity and regions of shared synteny to ensure their validity.

Further, DOGMA allows users to control the level of filtering by providing them with a data selection tool. This tool performs the filtering stage repetitively and counts the number of putative orthologs resulting by varying the distance parameters.

Also, DOGMA focuses on species which are not completely sequenced, and improves the usefulness of the limited resources allotted to these species by making both sequence comparisons and map comparisons to model species. That is, DOGMA makes species limited genetic data more useful than if putative orthologs remained unknown.

5.1.3 Testing

We tested DOGMA with *Phaseolus vulgaris*, a species of interest to some legume crop researchers. The results, described in Section 4.3, are promising. DOGMA was able to successfully detect several regions of shared synteny incorporating over 120 putative orthologs and lowered a further 84 to the category of reduced orthologs.

5.1.4 Generality

Though we tested DOGMA on only these species, it is designed to function with any set of closely related species, with the expectation that results will vary depending on the degree to which the species are related as well as the amount or completeness of the data present for each species. The only accommodation that needs to be made for alternate species is an accommodate for the specific ploidy differences between the species. For example, in the case of *Glycine max* and *Phaseolus vulgaris*, we accommodated for ploidy differences with an ad hoc filtering of pairs of sequences into one sequence. In this way, we could properly match one ortholog to a pair of matching sequences. To adapt to other sequences, it would be necessary to perform an analogous filtering depending on the ploidy levels.

5.1.5 Validation

We validate DOGMA using the same species as the previous test, but using a more complete set of sequence and map data for *Phaseolus vulgaris*. The validation indicates that DOGMA is functional and maintains the same sensitivity level of 95% compared to Ortholuge; however, it is yet inconclusive as to DOGMA's exact effectiveness due to the overlapping data noted in Section 4.5. We attempt to prove that DOGMA is capable of properly detecting false positives among Ortholuge's results by introducing an artificial paralog and removing it's true ortholog.

5.2 Future Iterations

Future iterations of DOGMA will account for some scenarios observed in the results that were not assessed in it's conception.

Collinearity inside of shared syntenic regions is a necessity by the definition of shared synteny; however, Figures 4.3 and 4.4 show that some putative orthologs are being detected as part of a region of shared synteny where they do not follow the correct order on both species. Future iterations of DOGMA will filter these orthologs based on their collinearity as well as their proximity to other putative orthologs.

Future iterations of DOGMA will make use of more matured data marshalling techniques.

Currently, DOGMA requires manual user intervention and customized scripts for each species and data sets to prepare the data for input into the pipeline. This is due to inconsistencies in the data structures which are naturally found when using data from multiple sources. It will also incorporate an algorithm to accommodate arbitrary ploidy differences without a manual pre-filtering step. In the future, DOGMA's data marshalling will be capable of managing more varied sets of data automatically.

5.3 Future Directions

There are several future steps to be taken to investigate and improve upon DOGMA which are for the time being beyond the scope of this thesis.

As the first of these steps, we will continue investigating the possibility of better defining what constitutes shared synteny. That is, we would like DOGMA to distinguish, perhaps based on marker density, areas of synteny without significant manual intervention as is seen during data selection in Section 4.1. Possibly the distance parameters would vary within a certain range to more concisely define the regions of shared synteny.

Secondly, when performing detection tasks on species of varying ploidy levels, DOGMA will use additional ortholog types to indicate levels of synteny. Hypothetically, DOGMA will be able to detect regions of synteny among reduced orthologs, which should potentially be referred to as paralogs. DOGMA would differentiate the matches based on the relative strength of the matches between the orthologs and the paralogs; that is, the paralogs should have a slightly weaker alignment score. Related to this research question will be a need to investigate the data and DOGMA's initial steps to ensure that it detects multiple putative or reduced orthologs in such cases of varying ploidy levels between the two species.

Thirdly, DOGMA will use the sequence data from the target species that was removed for not having mapping data associated with it, and compare it against sequence data from the related species inside of the regions of shared synteny in an effort to determine an approximate location for these sequences. That is, we will attempt to give approximate map location to sequences of the target species which have no mapping data available at present.

Also beyond the scope of this thesis is the investigation of a more adequate validation experiment to ensure the correctness of DOGMA. One possibility would be to use species for which extensive ortholog data is available, if such data exists, as a control and then test DOGMA verifying that it's results are consistent with said data, together with systematically constructed multiple reduced data sets to properly investigate the effects of having limited data. Furthermore, as DOGMA is a pipeline for use with arbitrary species, it is necessary to test with a variety of different species to properly gauge success.

With DOGMA in its infancy, there are many future steps to be taken, and investigations to be made. Even so, DOGMA represents a functional and potentially useful tool for researchers interested in making better use of model species to improve their knowledge of their own target species.

References

[1] DFCI medicago gene index. Accessed June 06, 2012. http://compbio.dfci.harvard.edu/tgi/cgi-bin/tgi/gimain.pl?gudb=medicago.

[2] DFCI soybean gene index. Accessed June 06, 2012. http://compbio.dfci.harvard.edu/cgi-bin/tgi/gimain.pl?gudb=soybean.

[3] LIS CMap. Legume Information System - www.comparative-legumes.org.

[4] Merriam-Webster Online Dictionary. http://www.merriam-webster.com/dictionary/gene.

[5] Phytozome: *Phaseolus vulgaris*. http://phytozome.net/commonbean_er.php.

[6] The Statistics of Sequence Similarity Scores. http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html.

[7] S.F. Altschul, W. Gish, W. Miller, E.W. Meyers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[8] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389, 1997.

[9] N.A. Campbell, J.B. Reece, L.G. Mitchell, and M.R. Taylor. *Biology: Concepts and Connections*. Benjamin Cummings, 4th edition edition, 2003.

[10] T.C. Carter and D.S. Falconer. Stocks for detecting linkage in the mouse, and the theory of their design. *Journal of Genetics*, 50(2):307–324, 1951.

[11] H.K. Choi, J.H. Mun, D.J. Kim, H. Zhu, J.M. Baek, J. Mudge, B. Roe, N. Ellis, J. Doyle, G.B. Kiss, N.D. Young, and D.R. Cook. Estimating genome conservation between crop and model legume species. *Proceedings of the National Academy of Sciences of the United States of America*, 101(43):15289–15294, 2004.

[12] M. Clark and W.J. Wall. *Chromosomes: The Complex Code*. Chapman and Hall, 1996.

[13] B.C.Y. Collard, M.Z.Z. Jahufer, J.B. Brouwer, and E.C.K. Pang. An introduction to markers, quantitative trait loci (qtl) mapping and marker-assisted selection for crop improvement: The basic concepts. *Euphytica*, 142:169–196, 2005.

[14] M. Margulies et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437:376–380, September 2005.

[15] J. Felsenstein. A mathematically tractable family of genetic mapping functions with different amounts of interference. *Genetics*, 91(4):769–775, 1979.

[16] W.M. Fitch. Distinguishing homologous from analogous proteins. *Systematic Zoology*, 19(2):99–113, 1970.

[17] W.M. Fitch. Homology: a personal view on some of the problems. *Trends in Genetics*, 16(5), 2000.

[18] D.L. Fulton, Y.Y. Li, M.R. Laird, B.G.S. Horsman, F.M. Roche, and F.S.L. Brinkman. Improving the specificity of high-throughput ortholog prediction. *BMC Bioinformatics*, 7:270, 2006.

[19] R. Guyot and B. Keller. Ancestral genome duplication in rice. *Genome*, 47(3):610–614, 2004.

[20] D. L. Hartl. What did gregor mendel think he discovered? *Genetics*, 131(2):245–253, 1992.

[21] T. Helms. Haldane's mapping function. Note: Page from an online genetics course our of North Dakota State University, http://www.ndsu.edu/ndsu/abergstr/webcourses/genetics/mf/mf01.htm, 2000.

[22] T. Helms. Kosambi's mapping function. Note: Page from an online genetics course our of North Dakota State University, http://www.ndsu.edu/ndsu/abergstr/webcourses/genetics/mf/mf09.htm, 2000.

[23] K.J. Hillers. What is crossover interference? *Current Biology*, 14(24):R1036 – R1037, 2004.

[24] http://gmod.org/wiki/Overview. Generic Model Organism Database project.

[25] R.A. Jenson. Orthologs and paralogs - we need to get it right. *Genome Biology*, 2(8):1002–1004, 2001.

[26] N. Jones, H. Ougham, and H. Thomas. Markers and mapping: we are all geneticists now. *New Phytologist*, 137(1):165–177, 1997.

[27] N.C. Jones. *An introduction to bioinformatics algorithems*. The MIT Press, first edition, 2004.

[28] E.K. Koonin. An apology for orthologs - or brave new memes. *Genome Biology*, 2001.

[29] D.D. Kosambi. The estimation of map distances from recombination values. *Ann Eugen*, 16(2):165–192, 1944.

[30] L. Li, C.J. Stoeckert, and D.S. Roos. OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Research*, 13:2178–2189, 2003.

[31] B. H. Liu. *Statistical Genomics: Linkage, Mapping, and QTL analysis*. CRC Press, 1998.

[32] P. McClean. Personal Correspondence. March 17, 2009.

[33] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[34] E. Passarge, B. Horsthemke, and R.A. Farber. Incorrect use of the term synteny. *Nature America Inc.*, 1999.

[35] G.A. Petsko. Homologuephobia. *Genome Biology*, 2(2):comment1002, 2001.

[36] Shoemaker R.C., K. Polzin, J. Labate, J. Specht, E.C. Brummer, T. Olson, and N. Young et al. Genome duplication in soybean (*Glycine* subgeneus *soja*). *Genetics*, 144(1):329–338, 1996.

[37] M. Remm, C.E.V. Storm, and E.L.L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314:1041–1052, 2001.

[38] M.C. Rivera, R. Jain, J.E. Moore, and J.A. Lake. Genomic evidence for two functionally distinct gene classes. *Proceedings of the National Academy of Sciences of the United States of America*, 95(11):6239–6244, 1997.

[39] M. Ronaghi, M. Uhlén, and P. Nyrén. A sequencing method based on real-time pyrophosphate. *Science*, 281(5375):363–365, July 1998.

[40] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of National Academy of Sciences Biochemistry*, 74(12):5463–5467, 1977.

[41] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[42] E.L.L. Sonnhammer and E.K. Koonin. Orthology, paralogy and proposed classification for paralog subtypes. *Trends in Genetics*, 18(12):619–620, 2002.

[43] D. P. Wall, H. B. Fraser, and A. E. Hirsh. Detecting putative orthologs. *Bioinformatics Application Note*, 19(13):1710–1711, 2003.

[44] P. Winter and G. Kahl. Molecular marker technologies for plant improvement. *World Journal of Microbiology and Biotechnology*, 11:438–448, 2004.

[45] H. Zhu, H.K. Choi, D.R. Cook, and R.C. Shoemaker. Bridging moden and crop legumes through comparative genomics. *Plant physiology*, 137(4):1189–1196, 2005.

Appendix A

Custom Perl Scripts

A.1 Join Sequence Data to Map Data

seq_to_map.pl

```
#!/usr/bin/perl

#Author: Andrew Couperthwaite
#email: acc020@mail.usask.ca

#This is the second part of the pipeline.
#This script returns links between the mapping data and the sequence data for a given species.
#This script takes in two parameters, the sequence file followed by the mapping file (in cMap tab-delimited format)
#This script outputs a pairs: sequence identifier and map identifier

my $seq_file = $ARGV[0];
my $map_file = $ARGV[1];

‘grep ’>’ $seq_file | cut -f 1 -d " " | cut -f 3 -d "-" > tmpseq.dat‘;

open(seqdat_handle, "tmpseq.dat") or die "$!";
@seqdat = <seqdat_handle>;
close(seqdat_handle);
‘rm -f tmpseq.dat‘;
open(mapdat_handle, $map_file) or die "$!";
@mapdat = <mapdat_handle>;
close(mapdat_handle);

foreach $seq (@seqdat) {
         chomp($seq);
```

```
        $seq = substr $seq, 1;
        foreach $map (@mapdat){
                 @tokens = split /\t/, $map;
                  if ($seq != ""){
                            if ( $tokens[4] =~ /$seq/ or $tokens[5] =~ /$seq/){
                                    print "g" . $seq . "\t" . $tokens[4] . "\t" . $tokens[5] . "\n";
                            }
                   }
        }
}
```

A.2 Make Correspondences

make_correspondences.pl

```
#!/usr/bin/perl

#Author: Andrew Couperthwaite
#email: acc020@mail.usask.ca

#This script creates correspondence information for input into cMAP based on the otholog detection from otholuge
#This script requires 3 input files: ortholog list, and files for two species as created by seq_to_map.pl


my $ortholog_file = $ARGV[0];
my $in1_mapfile = $ARGV[1];
my $in2_mapfile = $ARGV[2];
#my $ingroup2_file = $ARGV[1];
```



```
`cut -f 1 -d "\t" $ortholog_file | cut -f 3 -d "-" > in1tmp.dat`;
`cut -f 2 -d "\t" $ortholog_file > in2tmp.dat`;

open(in1_handle, "in1tmp.dat") or die "$!";
@in1_ortho_seqs = <in1_handle>;
close(in1_handle);

open(in2_handle, "in2tmp.dat") or die "$!";
@in2_ortho_seqs = <in2_handle>;
close(in2_handle);


`cut -f 1 -d "\t" $in1_mapfile > in1tmp2.dat`;
`cut -f 2 -d "\t" $in1_mapfile > in1tmp3.dat`;
`cut -f 3 -d "\t" $in1_mapfile > in1tmp4.dat`;
```

```
open(in1_handle, "in1tmp2.dat") or die "$!";
@in1_all_seqs = <in1_handle>;
close(in1_handle);

open(in1_handle, "in1tmp3.dat") or die "$!";
@in1_map_acc = <in1_handle>;
close(in1_handle);

open(in1_handle, "in1tmp4.dat") or die "$!";
@in1_map_name = <in1_handle>;
close(in1_handle);

`cut -f 1 -d "\t" $in2_mapfile > in2tmp2.dat`;
`cut -f 2 -d "\t" $in2_mapfile > in2tmp3.dat`;
`cut -f 3 -d "\t" $in2_mapfile > in2tmp4.dat`;

open(in2_handle, "in2tmp2.dat") or die "$!";
@in2_all_seqs = <in2_handle>;
close(in2_handle);

open(in2_handle, "in2tmp3.dat") or die "$!";
@in2_map_acc = <in2_handle>;
close(in2_handle);

open(in2_handle, "in2tmp4.dat") or die "$!";
@in2_map_name = <in2_handle>;
close(in2_handle);

my $i = 0;
my $k = 0;
my $feature_acc1;
my $feature_acc2;
my $feature_name1;
my $feature_name2;
my $feat1 = 0;
```

```
my $feat2 = 0;
print "feature_name1\tfeature_acc1\tfeature_name2\tfeature_acc2\tevidence\tis_enabled\n";
for (0 .. $#in1_ortho_seqs) {
         #chomp $in1_ortho_seqs[$i];
         #chomp $in2_ortho_seqs[$i];
         $k = 0;
         $feat1 = 0;
         $feat2 = 0;

         for (0 .. $#in1_all_seqs){
                   chomp $in1_map_acc[$k];
                   chomp $in1_map_name[$k];
                   if ($in1_ortho_seqs[$i] eq $in1_all_seqs[$k] )
                   {
                            $feature_acc1 = $in1_map_acc[$k];
                            $feature_name1 = $in1_map_name[$k];
                            $feat1 = 1;
                   }
                  $k++;
          }
         $k = 0;
         for (0 .. $#in2_all_seqs){
                  chomp $in2_map_acc[$k];
                  chomp $in2_map_name[$k];
                  if ($in2_ortho_seqs[$i] eq $in2_all_seqs[$k] )
                  {
                            $feature_acc2 = $in2_map_acc[$k];
                            $feature_name2 = $in2_map_name[$k];
                            $feat2 = 1
                   }
                  $k++;
          }
          if ($feat1 = 1 and $feat2 = 1){
                  print "$feature_name1\t$feature_acc1\t$feature_name2\t$feature_acc2\tOrtholog_Detection\t1\n";
          }
```

```
            $feat1 = 0;
            $feat2 = 0;
            $i++;
    }
    `rm -f in1tmp* in2tmp*`;
```

A.3 Find Syntenic Markers

find_synteny.pl

```
#!/usr/bin/perl

#Author: Andrew Couperthwaite
#email: acc020@mail.usask.ca

#This script will locate areas of shared synteny and modify CMAP features to reflect those areas
#This will access the CMAP database an extract features for specific maps, then using a metric, locate and define areas of shared
# synteny and finally modify the database such that those regions on the maps will be reflected.

#inputs: database name, username, password,
               desired distance between map 1 elements, desired distance between map 2 elements
#modify to input species/map set ids

#outputs a list of orthologs in the database to be changed


use DBI;
use warnings;
use strict;

sub zup {
    join "\n" => map {join " " => map {shift @$_} @_} @{$_ [0]}
}


my $databasename = $ARGV[0];
my $username = $ARGV[1];
my $password = $ARGV[2];
my $map1dist = $ARGV[3];
my $map2dist = $ARGV[4];
```

```
my $dbh = DBI->connect("dbi:mysql:$databasename","$username","$password") or die "Connection Error: $DBI::errstr\n";

#find map pairs that have correspondences
my $sql = "select map_id1, map_id2 from cmap_correspondence_lookup GROUP BY map_id1, map_id2 ORDER BY map_id1, map_id2";
my $sth = $dbh->prepare($sql);
$sth->execute or die "SQL error: $DBI::errstr\n";
my @map_id1;
my @map_id2;
my @row;
while (@row = $sth->fetchrow_array){
  push(@map_id1,$row[0]);
  push(@map_id2,$row[1]);
}


#for each map pair with correspondences...
my $mapid1;
my $mapid2;
my $i = 0;
my $j = 1;
my $featid1;
my $featid2;
my $featstart1;
my $featstart2;
my $featstop1;
my $featstop2;

#orthologues
my $olog;
my @ologs;
my $found = 0;
my $rec;

foreach $mapid1 (@map_id1){
  $mapid2 = $map_id2[$i];
```

```
  #foreach $mapid2 (@map_id2){
    #print "$mapid1\t$mapid2\n-------------------\n";
    $sql = "select feature_id1, feature_id2, feature_start1, feature_start2, feature_stop1, feature_stop2, feature_correspondence_id
          from cmap_correspondence_lookup
          where map_id1 = $mapid1 and map_id2 = $mapid2 ORDER BY feature_start1, feature_stop1, feature_start2, feature_stop2";
    $sth = $dbh->prepare($sql);
    $sth->execute or die "SQL error: $DBI::errstr\n";
    while (@row = $sth->fetchrow_array){
      #print "@row" . "\n";
      $olog = {
           id1 => $row[0],
           id2 => $row[1],
           start1 => $row[2],
           start2 => $row[3],
           stop1 => $row[4],
           stop2 => $row[5],
           correspondence_id => $row[6],
           mapid1 => $mapid1,
           mapid2 => $mapid2,
           synteny => 0
      };

      $found = 0;
      foreach $rec (@ologs){
          if ($olog->{id1} == $rec->{id2} && $olog->{id2} == $rec->{id1}){
              $found = 1;
          }
      }
      if (not $found){
           push @ologs, $olog;
           $j++;
      }
    }
      $i ++;
}
```

```
my \%seen1 = ();
my \%seen2 = ();
my @mapid1 = ();
my @mapid2 = ();
foreach $rec (@map_id1) {
  push (@mapid1, $rec) unless $seen1{$rec}++;
}
foreach $rec (@map_id2) {
  push (@mapid2, $rec) unless $seen2{$rec}++;
}


my $count;
my @current_map;
foreach $mapid1 (@mapid1){
  foreach $mapid2 (@mapid2){
    @current_map = ();
    $count = 0;
    foreach $olog (@ologs){
      if ($olog->{mapid1} == $mapid1 && $olog->{mapid2} == $mapid2){
          push(@current_map, $olog);
          $count++;
      }
    }
    #if more than one othologue on the current map
    if ($count > 1){
      #do stuff with current_map
         my @sorted_cur_map = sort({$$a{start1} >= $$b{start1}}@current_map);
         for($i = 0; $i <= $count - 1; $i++) {
                   for($j = $i+1; $j <= $count -1; $j++) {
                           if (abs($sorted_cur_map[$i]->{start1} - $sorted_cur_map[$j]->{start1}) <= $map1dist  &&
                               abs($sorted_cur_map[$i]->{start2} - $sorted_cur_map[$j]->{start2}) <= $map2dist){
                                     $sorted_cur_map[$i]->{synteny} = 1;
                                     $sorted_cur_map[$j]->{synteny} = 1;
```

```
            }
        }
    }
    foreach $olog (@sorted_cur_map){
            if ($olog->{synteny}){
                print $olog->{correspondence_id} . "\n";
            }
        }
    }
}
```

A.4 Create Matrix

find_synteny_matrix.pl

```
#!/usr/bin/perl

#find_synteny_matrix.pl
#Author: Andrew Couperthwaite
#email: acc020@mail.usask.ca
my $databasename = $ARGV[0];
my $username = $ARGV[1];
my $password = $ARGV[2];

my $dist1 = 50000000;
my $dist2 = 0;

print "\t";
for ($dist1 = 1000000; $dist1 <= 50000000; $dist1 = $dist1 + 1000000){
      print $dist1 . "\t";
}
print "\n";

for ($dist2 = 0; $dist2 <= 135; $dist2 = $dist2 + 2){
print $dist2 . "\t";
    for ($dist1 = 1000000; $dist1 <= 50000000; $dist1 = $dist1 + 1000000){
         $cur = `./find_synteny.pl $databasename $username $password $dist1 $dist2 | wc -l`;
         chomp($cur);
         print $cur . "\t";#. " " . $dist1 . " " . $dist2 . " ";
     }
     print "\n";
}
```