Development of fusion and duplication finder BLAST (fdfBLAST): a systematic tool to detect differentially distributed gene fusions and resolve trifurcations in the tree of life

Submitted by **Guy Leonard**

to the **University of Exeter** as a thesis for the degree of **Doctor of Philosophy in Biological Sciences** in **December 2010**

I certify that all material in this thesis, which is not, my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signature:

Guy Leonard 17/07/2011

Acknowledgements

I would firstly like to thank Dr. Thomas A. Richards for continued support, both during my PhD and in my MSc before that. Your time, motivation, discussion and ideas throughout the last 5 years have been invaluable. Secondly, I would like to thank Dr. Jamie Stevens. I have very much appreciated your supervision and support throughout my PhD research.

I am also grateful for the help and expertise offered by Darren Soanes (University of Exeter) for the use of his program 'Darren's Orchard', Bill Wickstead (University of Oxford) for his bioinformatics scripts and Peter Foster (Natural History Museum) for help with advanced phylogenetics.

I would like to thank everyone in room 310 (and all the offices before) and the whole CEEM group for friendship, encouragement and for putting up with me for four whole years! Special thanks also go to Theresa Hudson, Konrad Paszkiewicz and Kate Le Cocq for looking after me.

Finally, I would like to thank my mum and sister for nonstop support in my continued education, excellent proof reading services and lovely Sunday dinners. Thanks also go to my dad for proof reading and support throughout.

Abstract

The construction of a tree of life and the placing of taxa into their correct phylogenetic context is the underpinning of modern evolutionary biology. However, many parts of the tree are currently unresolved due to conflicts within the sequence data. These sources of conflict include: horizontal gene transfer (HGT), hidden paralogy, and the effects of methodological artefacts such as Long Branch attraction (LBA). These limitations are further compounded by absence of key taxa that are yet to be sampled. Therefore, whilst phylogenetic methods are fundamentally useful for the reconstruction of the tree of life, given their current limitations, additional strategies are needed in order to fully resolve the tree of life. Gene fusions represent a potential source of evolutionary synapomorphies useful for resolving contentious branching relationships in the tree of life. I therefore, built a program to analyse whole genome datasets for the presence of differentially distributed gene fusion events (shared derived characters - SDCs). These putative SDCs can then be polarised with the help of traditional phylogenetic techniques and used as synapomorphies on the tree of life. Having constructed this program and tested it on established fusion datasets, I analysed five sets of four genomes from across the tree of life (the Deuterostomia, Fungi, Vertebrata, Viridiplantae and Discicristata). I used this data to identify the relative rates of gene fusion events. Previous studies have suggested that fission events occurred more often than gene fusion events. However, our analysis broadly suggests the opposite (albeit with a higher rate of fissions in the Deuterostomia). This result has direct implications for the use of gene fusions as evolutionary informative synapomorphies because the identification of a lower rate of reversion suggests that

these characters are less likely to be homoplasious and therefore represent useful tools for polarising evolutionary relationships. Six phylogenetically informative synapomorphies were recovered, three in the Discicristata which resolve the monophyly of the Kinetoplastida and four in the Fungi, one of which represented a HGT event and was independently discovered and previously published. Thus, this thesis reports the development and testing of a new tool to identify differentially distributed gene fusion events. The datasets analysed demonstrate that the program can be used to find phylogenetically informative gene fusion characters that can help resolve the tree of life in conjunction with traditional phylogenetic methods.

Table of Contents

Table of Figures

"Nothing in biology makes sense except in the light of evolution"

- *Theodosius Dobzhansky, 1973*

-

"Nothing in biology makes any sense except in the context of its place in phylogeny, its context in the tree of life … reconstructing that tree is critical to understanding the living world"

- *(Sterelny & Griffiths, 1999).*

1 Introduction

1.1 Early Interpretations of the Tree of Life

The idea of a Tree of Life (TOL) 'stems' back to an early representation in Charles Darwin's book, On the Origin of Species (Darwin, 1876). Darwin presents the idea of grouping taxa into a tree topology by stating:

"…the affinities of all the beings of the same class have sometimes been represented by a great tree…"

However, the only diagram presented in his work was a species tree for a large unnamed genus. The more famous, 'I think' Darwin-tree, and so a much closer representation to a modern un-rooted phylogenetic tree topology, appears much earlier in one of Darwin's notebooks from 1837 (Darwin, 1837-1838). Ernst Haeckel, an important early microbiologist, also presented several early phylogenetic topologies with the most notable possessing three kingdoms, the Plantae, Protista and Animalia (Haeckel, 1866).

Since then the tree of life has been revised multiple times in order to attempt to resolve ancient evolutionary relationships and so to understand how the evolution of life on Earth proceeded. Initial trees were drawn by the grouping of morphological characteristics present in different extinct (based on the fossil record) and extant species. Woese (C. R. Woese, Kandler, & Wheelis, 1990) claimed that these early attempts at phylogenetic reconstruction, which included only the metazoa and

‘metaphyta’ (all plants) based on morphological characters alone, barely covered 20 percent of all evolutionary history, although that is likely to be overestimated; given the large diversity of eukaryotic and prokaryotic microbes (Rappé & Giovannoni, 2003). The use of nucleotide and protein sequence data has been incorporated into this effort (e.g. Page & Holmes, 1998). This has occurred as nucleotide and protein sequences are large-scale discrete characters that can be analysed statistically, whereas in comparison the continuous characters of morphology can suffer from homoplasy (correspondence by evolutionary convergence), difficulty in classifying discrete characters, and bias during the selection of data. The original morphologically based trees, see reviews in W. F. Doolittle & Brown, 1994; C R Woese, et al., 1990, suffered from poor resolution especially for the prokaryotic topologies due to the inconsistent structural and morphological data present between similar and yet very different species. These differences lead to a great deal of confusion in the placement and the grouping of those species. To illustrate this point single-celled organisms were, for a time, split into the Protoctista, consisting of single-celled eukaryotes, and the Monera (presently prokaryotes), consisting of bacteria and blue-green algae (Copeland, 1938). But the Protoctista were unable to be classified as a strict monophyly as they lacked a defining set of characteristics (Whittaker, 1969) leading to a reclassification of the grouping (largely based on the three main methods of nutrition; ingestion, photosynthesis and absorption). This classification resulted in the grouping of the fungi to its own group, putting some algae within the plants, and renaming the remaining microbes as the Protista (Whittaker, 1969).

Therefore, a revision to the natural system of order was needed and partly achieved by the inclusion of molecular data. Nevertheless, the problem of misplaced species and an unresolved positioning of the root of the TOL are still present; nonetheless, the use of sequence data has been vital in the restructuring of the branches of the TOL to resolve ancient evolutionary relationships, examples are discussed below.

1.1.1 Single-Gene Phylogenies

One of these first molecular phylogenetic trees was based upon a small subunit ribosomal RNA (SSU rRNA) (Carl R. Woese, 1996) gene which was chosen for analysis due to their conserved presence across all cellular organisms. These early SSU rRNA phylogenies described the tree of life as three major clades named by Woese and colleagues as Bacteria, Archaea and Eukarya. This led to earlier taxonomies being radically revised as with the integration of the Archaea (C. R. Woese, et al., 1990). It therefore became clear that splitting the tree of life into two major groups (between the eukaryotes and prokaryotes) was a faulty assumption (G. Olsen & Woese, 1993; C. R. Woese, 1987; C R Woese, et al., 1990). These early phylogenetic reconstructions had based their approach on the use of nucleotide sequences (Z. Yang & Roberts, 1995; Zuckerkandl & Pauling, 1965) of SSU rRNA genes (for example, T. Cavalier-Smith, 1993; M. C. Rivera & Lake, 1992; Sogin, 1991) yet this approach only derives one tree for one gene, which unfortunately represents a tiny fraction of the evolution of a species. Moreover, attempts to build single-gene trees resulted in increasingly unresolved topologies with datasets giving highly different phylogenies depending on the taxon sampling and the methods used (e.g. Brown & Doolittle, 1997). Therefore, the use of multi-gene phylogenies started to be seen as an approach to solve these discrepancies.

1.1.2 Multi-Gene Phylogenies

The trend for moving away from single-gene phylogenies was based on the assumption that a larger number of genes should improve the accuracy and resolution of species trees (e.g. Hillis, 1996). Multi-gene phylogenies, are as the name suggests, comprised of more than one gene contributing to the alignment needed to build a tree topology. They are superior to single-gene phylogenies as they attempt to form an understanding based upon a larger set of available data rather than a small subset which can be misleading as single genes can and do undergo different rates of evolution compared to those of the organism itself (Gribaldo, Poole, Daubin, Forterre, & Brochier-Armanet, 2010). Single gene analysis is also more likely to suffer from the effects of: 1) phylogenetic reconstruction artefacts (discussed later in Section 1.2), 2) horizontal gene transfer and, 3) inconsistent paralogue sampling with gene duplication and loss events (hidden paralogy - Figure 1:1).

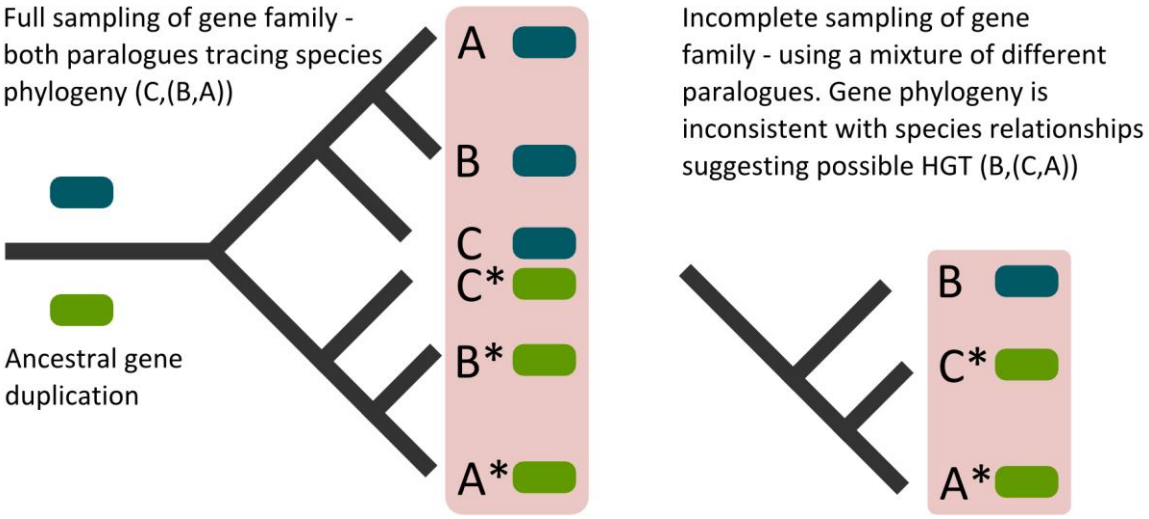


Figure 1:1 - Hidden Paralogy: Gene duplications may be confused with horizontal gene transfer if the sampling of genomes is incomplete or genes have been lost (T. A. Richards, Hirt, Williams, & Embley, 2003).

Inconsistencies between gene phylogeny and species phylogeny may also be attributed to horizontal (lateral) gene transfer (HGT) events, which can be observed as a possibility in the insert of Figure 1:1 (T. A. Richards, et al., 2003) and directly in Figure 1:2 - depicted by the grey arrow. HGT describes the process that transfers genomic material from one organism to another {Doolittle, 1998 #1173;Doolittle, 1999 #884;Keeling, 2008 #1565;Koonin, 2001 #286;Lawrence, 1998 #502;Ragan, 2001 #123}.

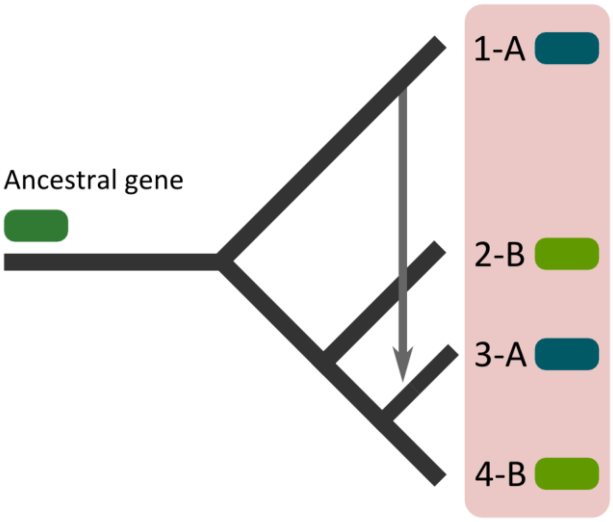


Figure 1:2 - Horizontal Gene Transfer: An ancestral gene (dark green) diverges into four lineages, lineage 1 receives a copy of the gene called 'A' (in blue) and lineages 2-4 (as a clade) receive a copy of the gene 'B' (in green). However, the taxa represented in lineage 3 have undergone a gene loss and acquisition event, loss of the 'B' gene and a horizontal gene transfer of gene 'A' in its place.

Two main ways to produce a multi-gene phylogeny have been developed; firstly, the concatenation method (sometimes called a super-matrix, Frederic Delsuc, Brinkmann, & Philippe, 2005) combines multiple gene sequences from the same taxon into one sequence and secondly, the creation of supertrees (Wilkinson & Thorley, 1998) based on the combination of individual previously computed single and/or multi-gene topologies.

1.1.3 Super-matrices (Concatenation)

Super-matrices or concatenated alignments are groups of gene alignments that have been combined together to create a large multi-gene alignment, any missing taxa or alignment data between the different alignments are normally removed or they can also be coded as missing data (H. Philippe et al., 2004). Many attempts, to use this

approach to build a phylogeny, predominantly appear to focus on smaller groups of associated taxa (Bailey et al., 2006; Bapteste et al., 2002; de Queiroz & Ashton, 2004; Frederic Delsuc, et al., 2005; Driskell et al., 2004; James et al., 2006; McMahon & Sanderson, 2006; Webb & Donoghue, 2005). Nevertheless, larger eukaryotic attempts using gene datasets in the order of 100 orthologue families have also been created (F. Burki & Pawlowski, 2006; Fabien Burki et al., 2007; F. Delsuc, Brinkmann, Chourrout, & Philippe, 2006; Hampl et al., 2009; Rodríguez-Ezpeleta et al., 2005; Rodríguez-Ezpeleta et al., 2007).

For example, Rodríguez-Ezpeleta et al. (Rodríguez-Ezpeleta, et al., 2005) describe an attempt to recover the monophyly of the 'Plantae' (now Archaeplastida; Adl et al., 2005) by creating a concatenated alignment of 143 amino acid sequences. What is most interesting about their experiment is that they attempt to test how many genes are needed in order to test and then resolve the Plantae monophyletic relationship. They do so by removing amino acid positions (Jackknifing; Joseph Felsenstein, 2004) from their concatenated alignment and noting the bootstrap support values for the monophyly of the Plantae. In Figure 1:3, reproduced from (Rodríguez-Ezpeleta, et al., 2005), the thick black line represents the support for the monophyletic relationship of the Plantae, it shows that as the number of amino acid positions decreases the bootstrap support also decreases; most notably it decreases rapidly towards weaker support once the number of positions remaining is close to 5000. This is a prime example for the need to make sure that the signal outweighs that of noise in the data being investigated.

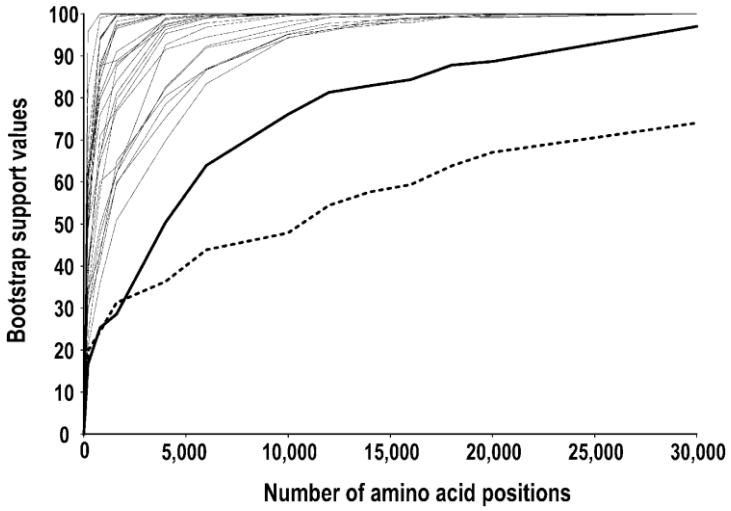


Figure 1:3 - Reproduction of Figure 3 from Rodríguez-Ezpeleta et al. (Rodríguez-Ezpeleta, et al., 2005). The main point of interest is the thick black line, which represents the bootstrap support for the monophyly of the Plantae from their concatenated alignment.

Identifying the shape of the tree of life and polarizing ancient branching relationships among the eukaryotes is a difficult process. A synthesis of data from ribosomal DNA genes (for example, Thomas Cavalier-Smith, 2004; Moreira, Lopez-Garcia, & Vickerman, 2004) and multi-gene phylogenies (Fabien Burki, Shalchian-Tabrizi, & Pawlowski, 2008; Hampl, et al., 2009; Rodríguez-Ezpeleta, et al., 2005; Rodríguez-Ezpeleta, et al., 2007) combined with selective comparisons of ultra-structural characteristics across plants, fungi, animals and protists has led to the identification of six large eukaryotic taxonomic collectives, often called super-groups. These groups include: 1) the opisthokonts, encompassing animals and fungi, 2) the Archaeplastida (Plantae) encompassing plants and some algae, 3) Amoebozoa, 4) Rhizaria, 5) chromalveolates, and 6) the excavates (see Adl, et al., 2005; A. G. Simpson & Roger, 2004). The groupings of the Excavata and the Chromalveolata, together, encompass a range of divergent cellular morphologies and lifestyles, including phototrophs,

parasites, and heterotrophic free-living protists. These two groupings remain amongst the most controversial classifications with inconsistent support in multi-gene phylogenies (A. G. B. Simpson, Y. Inagaki, & A. J. Roger, 2006) and share no consistent morphological and/or genomic characters (T. Cavalier-Smith, 2003; Hampl, et al., 2009; Rodríguez-Ezpeleta, et al., 2007; A. G. Simpson, Y. Inagaki, & A. J. Roger, 2006). Moreover, the classifications of the other groups are also relatively weakly supported causing controversy over their groupings (Roger & Simpson, 2009). Nevertheless, the results from these studies can be combined with the super-tree and/or sequence concatenation methods and used to generate a topology for the tree of life, indeed this is the approach taken by the Tree of Life Web Project (http://tolweb.org/tree).

1.1.4 Super-trees

Some of the earliest supertrees (Gordon, 1986) were built to assist systematicists in the construction of an overall tree topology prepared from several phylogenies where gaps in the data existed, mainly due to low taxon sampling (as the number of available genomes was fewer). These trees are described as informal, meaning that the trees were hierarchically nested without using any form of optimisation technique and so do not necessarily accurately represent evolutionary history (Bininda-Emonds, 2004). In essence, supertrees combine separate gene trees into one single species tree by way of diverse optimisation techniques (Bininda-Emonds, 2004; Bininda-Emonds, Gittleman, & Steel, 2003) with matrix representation using parsimony (MRP) (Baum, 1992; Ragan, 1992) being the most popular (Bininda-Emonds, 2004; F. G. Liu et al., 2001; Pisani & Wilkinson, 2002). In one study a phylogenetic supertree based upon 168 taxa and 5,741 genes was generated to provide insight into the chimerical origins of

the eukaryotes (Pisani, Cotton, & McInerney, 2007). The study compared the results of the supertree to the eight most prominent hypotheses for the origins of the Eukaryota, using a systematic signal stripping methodology allowing the authors to remove the signal derived from the mitochondrial and plastid footprint of endosymbiosis. The remaining data was then used for a subsequent supertree analysis and demonstrated support for both the sulphur-dependent (Searcy & Hixon, 1991) and hydrogen-dependent syntrophy (Martin & Muller, 1998) eukaryotic origin hypotheses which both involve a symbiosis founded on metabolic interactions between an Archaea and a bacterium. These results are not definite but do provide a useful example of the application of the supertree method, indeed numerous multi-gene analyses, both from supertree and concatenated gene analyses, have shown support for conflicting TOL phylogenies (for review please see Table 1 in Gribaldo, et al., 2010). These experiments demonstrate that such methods can be extremely informative for understanding ancient evolution.

1.2 Molecular Sequence Based Phylogeny

Similarly, as with the problems associated with the early attempts to resolve ancient evolutionary relationships, modern molecular sequence based phylogenetic reconstruction is also beset with many problems, which obscure the phylogenetic signal and the measurement of evolutionary change. In the next section, the different processes for phylogenetic reconstruction using molecular sequence based methods and the problems associated with each stage are discussed.

1.2.1 Sequence Data

One of the first problems experienced is portrayed in Figure 1:4 and it illustrates the difference between observed changes and actual changes present in sequence data. In the figure, the dark-blue line signifies the potential or actual mutation rate of a site given uniform patterns of mutation in a nucleotide sequence. The brown line represents the observable mutation rate. For example, consider the existence of a site-specific mutation, this conventionally implies that there has been exactly one mutation and this is indeed our observation at the left side of Figure 1:4 - where the brown and dark-blue lines are merged. Yet, this is not necessarily the actual pattern present across evolutionary time (which is vast); a nucleotide may be homoplasious and have changed once and then reverted to its original state. For example, a nucleotide may change from an *A* to *C* and then back to an *A* again or other sequences may, for example, change from an *A* to *T* to *C* to an *A*. In either case, we are unable to observe any difference in the outcome, but there has been more than one change, consequently obscuring the actual evolutionary history. Over a long period (which partly determines a group's evolutionary distance), ancient changes in the nucleotide sequences of an organism are effectively lost due to the transition and transversion events which cause these multiple site changes. As we are unable to observe these ancient changes on the DNA, the perceivable mutations (the brown line) reach saturation, forming a plateau where it is not possible to discern with confidence, their phylogenetic relationships.

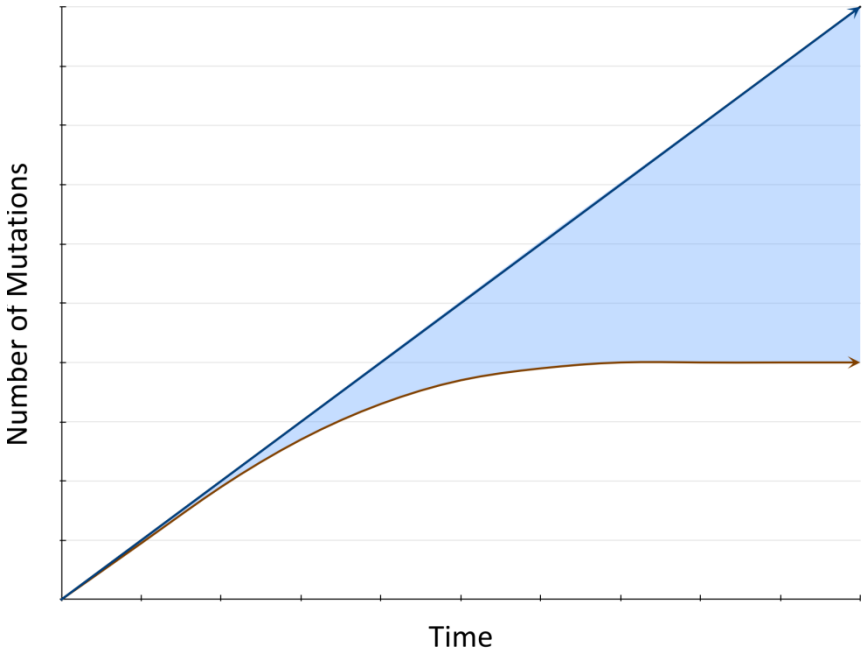


Figure 1:4 - Site change in nucleotide sequences over evolutionary time. The blue line indicates potential or actual changes whereas the brown line indicates observable changes. The shaded area highlights the difference between the two lines (Page & Holmes, 1998).

Site-rate change in phylogenetic analysis is partly controlled for by the use of substitution models, which attempt to correct for the process of change between sites over time. One of the first models of DNA evolution was proposed by Jukes (Jukes & Cantor, 1969) and assumed an equal rate of change for all nucleotide bases. Subsequently, several other models have been proposed, the most commonly used being the Generalised Time Reversible (GTR) (Rodríguez, Oliver, Marín, & Medina, 1990; Tavaré, 1986) model which attempts an analysis without any *a priori* assumptions (Lanave, Preparata, Sacone, & Serio, 1984). Amino acid sequences are also subject to site-rate variation but they do so at different rates to those of nucleotides. This is due in part to the redundancy present in the genetic code (where a change can occur in one base of a codon and not necessarily change the amino acid

produced), frame shifts in codons, and the overall functional constraints on protein evolution. Therefore, models of substitution were designed to account for character changes in amino acid sequences, amongst the first was the Dayhoff matrix (Dayhoff, 1965). The matrix was based on a set of global alignments of closely related sequences and the probability of evolutionary changes observed between them. This was later updated using a larger and more 'modern' set of sequences and called the JTT matrix (Jones, Taylor, & Thornton, 1992). There are now numerous models to choose from, a recent matrix called the LG model (Se Quang Le & Gascuel, 2008; Si Quang Le, Lartillot, & Gascuel, 2008) was based upon nearly 4000 alignments from the PFAM (Sanger Protein Family) database. In order to assess which model best fits your alignments several model testing programs exist, for example in this thesis I will use the program MODELGENERATOR (Keane, Creevey, Pentony, Naughton, & Mclnerney, 2006) to assess alternative matrices, which are discussed in Chapter 2: Methods, the tree building program RAxML (Alexandros Stamatakis, 2006), with the addition of my freely available and open source easyRAx script (http://projects.exeter.ac.uk/ceem/easyRAx.html), can also be used to find the best substitution model for your data.

In addition to the substitution matrices a further method can be applied to the models in the form of an α-parameter, which adjusts a gamma distribution, associated with your alignment. The gamma distribution is simply a probabilistic method of assigning a value to a variable depending on where it falls within a particular interval of the curve, the outcome being, in our case, that the higher the value of the α-parameter the lower the heterogeneity of the sequence alignment (Ziheng Yang, 1994). This method helps

to assist with the accounting of variable substitution rates among invariant sites. However, using a continuous gamma distribution to estimate the α-parameter can be too complex and therefore a discrete gamma model is used, which assigns groups of sites into different categories, usually 4 or 8 (Ziheng Yang, 1994), in order to form an approximation of the continuous gamma distribution.

Sites that remain constant over time are classified as invariant sites, these characters are important because an analysis which assumes that sites are free to vary will 'under-correct' for changes when sites that cannot change are present, therefore representing a violation of the assumptions of the model. A good example of this effect is evident in the corrected placement of the Microsporidia from deep-branches to grouping with the fungi (Hirt et al., 1999). Two elongation factors, EF-1α and EF-2, previously supported an early divergence of the Microsporidia, however, when correcting for invariant sites the support for this hypothesis dramatically reduced, for example the bootstrap for *Glugea plecoglossi* (a microsporidia infecting fish) as the first branch was only 54% which further reduced to 33% when invariant sites were accounted for.

Inappropriate correction of site-rate variation either by invariant sites and/or site rate heterogeneity, directly relates to another problem concerning phylogenetics, specifically that of Long Branch Attraction (LBA) (J Felsenstein, 1978; H. Philippe, 2000; H. Philippe & Germot, 2000). The observed error rate for site-rate changes will be larger for longer branches than it is for shorter branches. LBA describes the artefact where a group of sequences, which have considerably longer branches, making them

more divergent, than another set of sequences increases their likelihood of being grouped together, either by exclusion from other groups or attraction by falsely similar signal. Use of complex parameter-rich substitution models and other related parameters help to reduce this artefact, however, in the case of very long branches their removal from the alignment may be the best and only option. The effects of LBA are particularly misleading for ancient eukaryotic phylogenies, (Dacks, Marinets, Ford Doolittle, Cavalier-Smith, & Logsdon, 2002; Embley & Hirt, 1998; Hervé Philippe, 2000; H. Philippe, 2000); a specific example is that of RNA Polymerase II (RPB1) from *Giardia intestinalis* and *Trichomonas vaginalis*, which are subject to particularly intense LBA effects.

Heterotachy (meaning different speeds) (Peter Lockhart et al., 2006; Lopez, Casane, & Philippe, 2002; Herve Philippe, Zhou, Brinkmann, Rodrigue, & Delsuc, 2005) is another source of potential noise (and LBA) in phylogenetic reconstruction. It relates to the possibility that substitutions may occur at different positions within different phylogenetic groups, as opposed to the general assumption that fast and slow sites are always in the same position across different taxonomic groups. It has been described as analogous with the covarion model (Joseph Felsenstein, 2004; Fitch & Markowitz, 1970), which is a mathematical tool for the detection of repeating patterns specifically with reference to the rates of evolution on different branches and within codon sites of aligned sequences.

Compositional bias describes the amount of AT/U or GC nucleotides that are present in the RNA or DNA sequences of some taxa, whereby taxa with similar biases are

attracted together by false signal. It was first presented by Hasegawa (Hasegawa & Hashimoto, 1993) concerning the tree topologies produced by analyses of rRNA genes. The effects of compositional bias are often thought to be removed by the use of protein coding sequences, as the potential for an amino acid mutation in the functional protein is less, meaning its composition should remain constant compared to the DNA. Nevertheless, Foster et. al. (Peter G. Foster & Hickey, 1999) warns of the potential for phylogenetic reconstruction using protein coding sequences to be affected by the presence of compositional bias. Attempts to control for compositional bias have been made with the LogDet transformation model (PJ Lockhart, Steel, Hendy, & Penny, 1994; M. Steel, 1994) which calculates the 'logarithm of determinant of a matrix' as a measure of the distance among characters such as nucleotides and amino acids (Massingham & Goldman, 2007). These measures of distance are then fitted to a tree using a range of alternative tree calculation methods such as neighbour-joining or a Fitch step-wise addition method (Swofford, 2003).

A further problem occurs at the taxon sampling level; if the sampling is non-congruent between analyses or an analysis is missing key taxa then the resulting tree topology will be skewed in one direction. Therefore, it is the responsibility of the researcher to make sure that they have sampled a wide variety of sequence databases, via homologous BLAST searches, and then to include as many taxa as possible or relevant to the question in hand. Failure in this process could mean that a tree confirms a hypothesis that otherwise would be disproved by the inclusion of more taxa. Inappropriate taxon sampling may also lead to long-branch attraction (LBA) artefacts, as previously discussed, and can also suggest the occurrence of a HGT event which

later reverts with the inclusion of better taxon sampling - for example (Horner & Embley, 2001) who contested the HGT hypothesis for the acquisition of the mitochondrially derived chaperonin 60 (cpn60) gene in *Giardia lamblia* and *Entamoeba histolytica. Giardia* and *Entamoeba* are anaerobic protists, which were thought to have never possessed either a mitochondria or a mitochondrially/α-proteobacterially derived cpn60 gene. Initial analysis of this dataset demonstrated that *Giardia* and *Entamoeba* branched together in contrast to their taxonomic classification and suggested HGT. Re-analysis with improved taxon sampling, specifically the inclusion of the *Spironucleus barkhanus* cpn60 gene into their phylogeny, demonstrated a corrected branching position for both *Giardia* and *Entamoeba* suggesting the HGT hypothesis was not valid, and therefore suggested both taxa possessed mitochondrially/α-proteobacterially derived genes. As the sequence databases and the number of sequenced genomes are constantly being updated, it is important to keep any analyses completely up-to-date.

Gene family evolution may also present complications in the resolution of phylogenies and these present themselves by way of gene loss and gene duplication. Gene loss accounts for the differences in gene collections between genomes (Krylov, Wolf, Rogozin, & Koonin, 2003) and can, with incomplete sampling of paralogues, lead to the building of inaccurate tree topologies, see Figure 1:1. Similarly, when genes are duplicated, creating multiple paralogues, this can cause the wrong paralogue to be selected and so complicate the resolution of a genes ancestry. Gene duplication is where a repetition of a particular ORF on a chromosome within an organism occurs. They generally arise due to errors in replication, for example during meiosis, and are

not limited to single gene duplications. Indeed, whole genomes can be duplicated, much like that in the ancestor of the yeast *Saccharomyces cerevisiae* (Kellis, Birren, & Lander, 2004) and also many plants, with modern wheat being a notable example (Akhunov et al., 2003). Although they can cause problems in phylogenetic reconstruction, if duplications can be polarised accurately they can be used as a shared derived character (a synapomorphy) and mapped on to a phylogenetic topology.

An example of how complex and mosaic gene family evolution can be, is described by Keeling et al. for the EF-1α and the related paralogue family EF-like (EFL) genes (Patrick J. Keeling & Inagaki, 2004). These two paralogue families demonstrate a mutually exclusive taxon distribution across the eukaryotes forming a mosaic-like pattern, where closely related taxa have often retained different anciently derived paralogues. Consequently, any attempts to use the EF-GTPase families to directly infer a species tree would be extremely misleading. These two issues (duplication and loss) in conjunction can represent a mosaic-like pattern of evolution across the tree of life.

Another factor that can confuse phylogenetic signal is horizontal gene transfer (HGT). HGT involves the transfer of genetic material between distinct lineages or across species boundaries see Figure 1:2. The problem occurs when two or more distantly related taxa have exchanged a gene, a phylogenetic tree represents this by showing them to be more closely related than truly reflecting their evolutionary history, this is because the HGT event is the same whereas the other genes of each taxon are dissimilar. A recent example of HGT events, that I was involved in, was reported between the plants and fungi which demonstrated nine putative HGT events between

the distantly related plants and fungi after more than 400 million years of co-evolution (Richards et al., 2009). As we can see progress towards a better understanding of any tree topology, including the wider eukaryotic tree of life, must take into account all of these potential sources of error and provide suitable and effective methods to reduce them. Some of these methods we have mentioned above, such as the use of parameter rich substitution models, and we shall discuss further methods to improve phylogeny below.

1.3 Phylogenomics

The term phylogenomics is used here in a different sense to its original meaning (Eisen, 1998), which focused solely upon the prediction of a function of an unknown gene by its placement within a phylogenetic tree. Here, I will use the term phylogenomics to describe the reconstruction of gene ancestry from multiple genes taken from whole genome datasets.

1.3.1 Recent Interpretations of the Tree of Life

A more sophisticated approach to try to understand the Tree of Life, compared to our earlier understandings (in Section 1.1), is driven by the utility of the phylogenomic approach. As I outlined previously, the grouping of taxa that share a set of morphological characters together has led to some confusion and so more recently a different approach has been taken to define the evolutionary relationships of taxa based on multiple gene phylogenetic analyses or based upon similar genetic/genomic characters. This later innovation has led to several proposals for not just the placement of the root of the eukaryotic tree but also for revisions to the underlying branching

order of the eukaryotic tree and the number of major groupings that are supported by each analysis, reviewed below.

The tree of life is presented as three major groups or domains: the Archaea, Bacteria and Eukarya (Woese et al., 1990). Yet, there exist two competing hypotheses for their branching order. One places the eukaryotes as a separate branch to the Archaea and Bacteria - the three domains tree (Woese and Fox, 1977; Baldauf et al, 1996). The other places the eukaryotes emerging from within a paraphyletic Archaea, often shown as the sister group of the Crenarchaeotes - the Eocyte hypothesis (Lake et al, 1984; Skophammer et al., 2007; Archibald et al., 2008; Cox et al., 2008). Adding further complexity, a recent analysis included a fourth domain with the inclusion of giant viruses (Boyer et al., 2010). They took sequences from metagenomic databases and viruses to create phylogenetic trees based upon several genes involved with DNA processing. This placed the new domain, the giant viruses, between the grouping of the two domains of Eukarya and Archaea and the Bacteria, suggesting a four domain tree topology. Nevertheless, this hypothesis has been very recently contested by Williams et al (2011) and returns us to three domains of life. They suggest that the previous study was subject to bias in the data resulting from non-phylogenetic signals (e.g. compositional heterogeneity and homoplasy). The authors repeated the analysis, albeit with a slightly different gene content, and subsequently tested the dataset with a more rigorous set of phylogenetic reconstruction methods (e.g. ProtTest, PhyloBayes and RaxML) paying closer attention to the modelling of compositional heterogeneity.

In order to help visualise the potential confusion that happens in one domain, a consensus eukaryotic Tree of Life is displayed, in Figure 1:5, which is taken from Baldauf (2008) and is in turn based upon a set of previously published molecular phylogenetic and ultra structural data. Its use here is more a guide than a definitive explanation. Note that the tree shows the two currently proposed positions for the root of the eukaryotic tree, indicated by the black dotted lines, and eight divergent groups (each a represented by a different colour). Note also that the terminal branches are essentially a four-way polytomy with further lack of resolution in the Chromalveolata/Rhizaria grouping. Therefore, the groups within the eukaryotic tree range anywhere from three to eight major clades depending on the datasets that are used to construct them (Adl & Simpson et al., 2005; Burki et al., 2008)

Some groups, for example the Archaeplastida (Plantae), are well established within the eukaryotic tree, which is to say they are considered to be monophyletic. Rodriguez-Ezpeleta et al (2005), which I have mentioned before (Figure 1:3), proposed that the Plantae (now Archaeplastida; Adl et al., 2005) are monophyletic based upon a concatenation of large dataset of genes and the use of robust phylogenetic testing. Now we know they are monophyletic we are better placed to identify a single primary cyanobacterial endosymbiosis leading to all plastids (Archibald et al., 2008).

Another group, the Amoebozoa, were also shown to be monophyletic, even though the group contains highly divergent and fast evolving species, by Bapteste et al., 2002. The authors included a large concatenated alignment of 140 orthologous genes with an archaeal outgroup and the inclusion of three Amoebozoa taxa (the Conosa -

Dictyostelium, *Mastigamoeba* and *Entamoeba*) and subjected it to various tree reconstruction methods (e.g. maximum likelihood and Bayesian inference). In each case, a largely well supported monophyly for the Conosa is recovered, which is part of the Amoebozoa group in Figure 1:5, providing the first consistent dataset supporting this major clade.

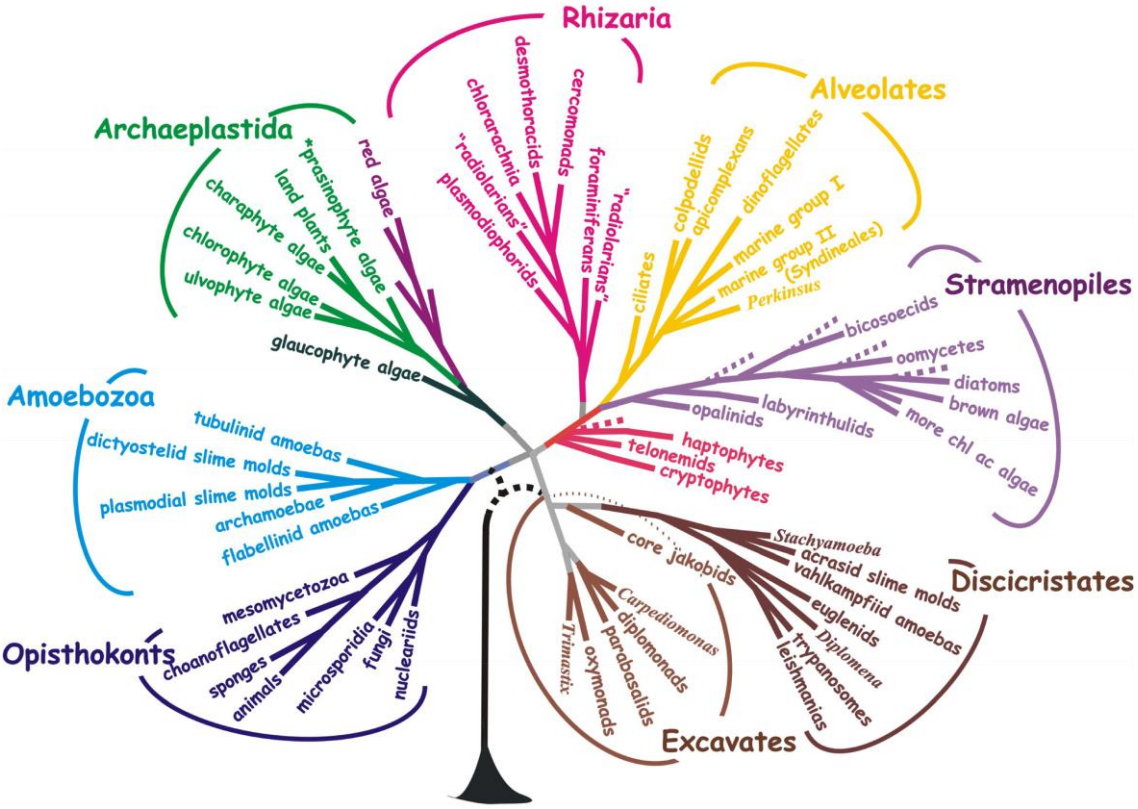


Figure 1:5 - A consensus phylogeny proposed by Baldauf, 2008 based upon a selection of molecular phylogenetic and ultra structural data. It is reproduced here simply as a guide to the accompanying text, note the two proposed positions of the root of the eukaryote tree represented by dotted black lines.

However, other groups are not as easy to confirm for their monophyly; Rodriquez-Ezpeleta et al., (2007) recovered phylogenetic evidence for Excavate monophyly including the jakobids, the Euglenozoa and Heterolobosea (Discicristata) and for the

association of the Cercozoa (Rhizaria) with the Stramenopiles and Alveolates (as one group effectively expanding the Chromalveolata (Cavalier-Smith, 2004)), whilst also confirming the monophyly of the Opisthokonts, Amoebozoa and Archaeplastida (Rodriquez-Ezpeleta et al., 2007). Their groupings are again based upon a large concatenation of genes and the use of fast maximum likelihood tree reconstruction methods (discussed later). This demonstrates the two groups of Alveolates and Stramenopiles form one group called the Chromalveolates, which branches next to the Rhizaria, while moving haptophytes, telonemids and cryptophytes down and potentially out of the Chromalveolata branch - Thus, collapsing the number of groups from eight to six or five (depending on where one draws the Chromalveolate boundary).

Another analysis, Hampl et al., 2009, suggests that Excavata are monophyletic (including the Discicristata) and again groups the Alveolates and Stramenopiles together placing Rhizaria within them. Their analysis follows similar methods to the previously mentioned analyses by way of building a large concatenated dataset, multiple phylogenetic tree construction methods and the gradual removal of fast-evolving sites.

Burki et al., (2008) goes one step further and collapses the eight groups into four, confirming monophyly for the Excavates, Opisthokonts, Amoebozoa and a further clade of all the remaining groups, indicating a 'megagroup' of ancestrally photosynthetic eukaryotes. Again, the authors constructed a multi-gene concatenated dataset and subjected it to multiple phylogenetic reconstruction methods and

performed the removal of fast-evolving sites to test for long branch attraction (Philippe, 2000).

It is apparent that even well-established and highly supported hypotheses surrounding the tree of life can be contested or further confirmed by the inclusion of new evidence (in the form of new genes or new taxa) and by analysis of different methods. It is also noticeable that most of these studies follow a similar methodology and can resolve substantially different branching orders. It is in this area, which I hope to attempt to bring new methods and information to the field with the completion of the aims of thesis (outlined in Section 1.6). The next section discusses the problem of rooting three-branched trees (such as that of the Tree of Life) and how novel and rare additional data can be used to help understand and clarify these concepts where the continual application of traditional methods have been unsuccessful.

1.3.2 Rooting Three Branched Trees

Tree topologies with three main branches (trifurcations, in essence a polytomy) can occur when the root of the three groups of taxa cannot be established to be in any one of the three possible positions based on all the phylogenetic data available. The extreme example is that of all known life on Earth, the root and relative branching order between the Eukarya, Archaea and Bacteria (Forterre & Philippe, 1999; Gribaldo, et al., 2010; Koonin, 2010; Hervé Philippe & Forterre, 1999; Maria C. Rivera & Lake, 2004) which has yet to be firmly established. The eukaryotes are also currently unresolved and according to recent data form a trifurcation with three major groups

i.e. unikonts (Opisthokonts, Amoebozoa and Fungi), Excavates (for which holophyly[1] is unconfirmed and where many key groups are missing from recent analyses, Fabien Burki, et al., 2008, and all other eukaryote groups (e.g. Rhizaria and Chromalveolata for which holophyly[1] is also unconfirmed) including the Archaeplastida (Fabien Burki, et al., 2008) - Figure 1:5. To understand this problem we can look at a more contrived example, if we refer to Figure 1:6 which depicts the case for a root between three groups, A, B and C. We can see by the use of the three coloured arrows that the root could be placed in one of three locations. The insert illustrates three rectangular cladograms, which should help the reader in visualising the arrangement of the branches. If we look at the dark-red coloured root, we can see that a clade is formed between the two groups A and C, which can then be described as monophyletic. In this case, as the position of the root is considered uncertain, based solely on molecular sequence phylogenetic methods, it is reasonable to suggest that further ways of grouping the taxa need to be employed. Indeed, these methods are discussed in depth in Section 1.4 but are described here very briefly by way of an example in Figure 1:6. Consider that group A and C shares an attribute that is not possessed by B; this automatically groups them together to the exclusion of B and so forces the root into a

[1] ***Monophyly*** is the term used to describe a group of taxa that share a common descent, which is to say that a group contains all descendants of and includes the last common ancestor (LCA) (Ashlock, 1971). For example, in Figure 1:6 the red coloured insert depicts a monophyly between A and C and the root of the phylogeny. A term related to this is ***holophyly***, less common in its use; nevertheless it can be viewed as a strict form of monophyly. This is achieved by excluding the LCA (Mats, 2008) from the phylogeny, thus excluding the root. Neither is to be confused with the term ***paraphyly***, which describes a group that contains the LCA but not all of the descending groups or with ***polyphyly*** which is where the LCA is not a member of the group nor are all the descending groups.

position between them. These attributes or synapomorphies, if they can be shown to obey 'Dollo's law'[2] are extremely important as they can be used to identify contentious evolutionary relationships. For an interesting example and an entertaining review of this concept please see Chapter 28 of 'What, if anything, is a Zebra?' (S. J. Gould, 1990) - where the ideas of shared derived characters are elucidated by way of discussion on how the three types of Zebra can be shown to be related by several different shared derived characters; grouping *Equus burchelli* and *Equus grevyi* zebras together to the exclusion of a sister group containing the true horse and *Equus zebra*.

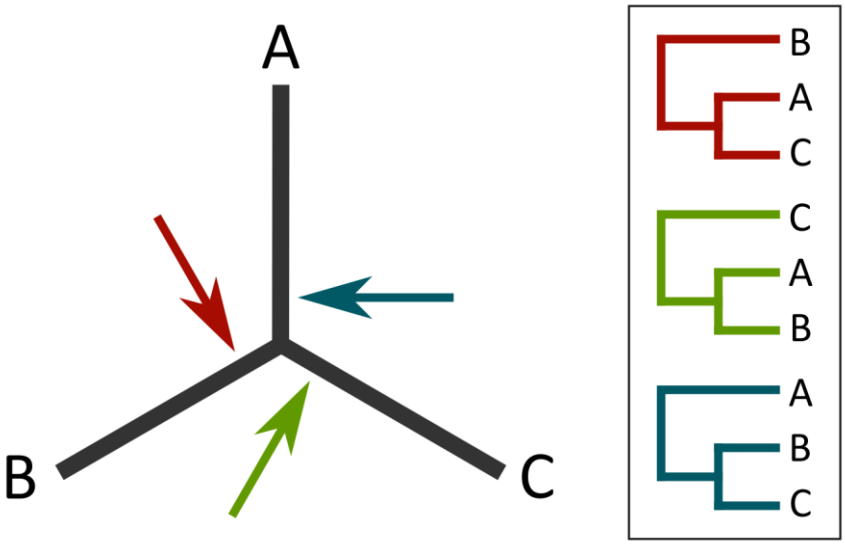


Figure 1:6 - A trifurcated tree topology of groups A, B and C, the arrows denote the three possible locations for a root. The insert depicts the three cladogram topologies for the location of each root; arrow colour corresponds to cladogram colour.

[2] Dollo's law states that evolution is not reversible; meaning that once a broad form of some character has evolved it forecloses any reversal of that form. One of Dollo's examples for this was demonstrating that unrolled ammonoids have not reverted to the ancestral straight nautiloid (Dollo, 1922). For a larger historical discussion of this concept, and on Dollo himself, with a copy of the original inscription in French please see (Stephen Jay Gould, 1970).

1.4 Synapomorphies or Shared Derived Characters

Shared derived characters (SDCs) or synapomorphies are a cladistic device that can be used to group organisms into specific clades. Shared indicates that a particular character is possessed by two or more taxa and derived indicates that the character is present in an organism but absent from the last common ancestor. SDCs are traditionally a morphological device; nevertheless, they have more recently been used to refer to shared derived genetic or genomic traits, such as gene fusions (T. Cavalier-Smith, 2002; Jenkins & Fuerst, 2001; Stechmann & Cavalier-Smith, 2002, 2003), HGTs (Andersson, Sarchfield, & Roger, 2005; Minotto, Edwards, & Bagnara, 2000), insertion/deletions within conserved open reading frames (J. M. Archibald, Rogers, Toop, Ishida, & Keeling, 2003; Bapteste & Philippe, 2002; Bass et al., 2005; P. J. Keeling & Palmer, 2001), location of introns (e.g. Everett, Kahane, Bush, & Friedman, 1999) and gene order within operons (Brinkman et al., 2002), for review please see Rokas & Holland, 2000. These characters enable trait polarization (i.e. rooting between groups of taxa that possess a derived character and groups that do not) which has advantages over sequence phylogeny, which are liable to systematic artefact such as long-branch attraction (Embley & Hirt, 1998; Stiller & Hall, 1999).

The SDC must be present in two or more related taxa and also in their most recent common ancestor (MRCA) yet their ancestor in turn, or the last common ancestor (LCA, the concestor), must not possess that same trait. In Figure 1:7 Group A represents a group (or clade) containing the synapomorphic character symbolized by the green rectangle, note how the root of the tree (the LCA) does not possess the SDC.

The term should not be confused with a symplesiomorphy (shared primitive characters) which is where two or more taxa seemingly possess a SDC however; they have inherited it independently, for example wings in bats and birds. Group A and Group B in Figure 1:7 could, for example, represent a symplesiomorphy, the red and green denoting the same SDC but resulting from different ancient lineages; because they are discrete, they are also separate synapomorphies.

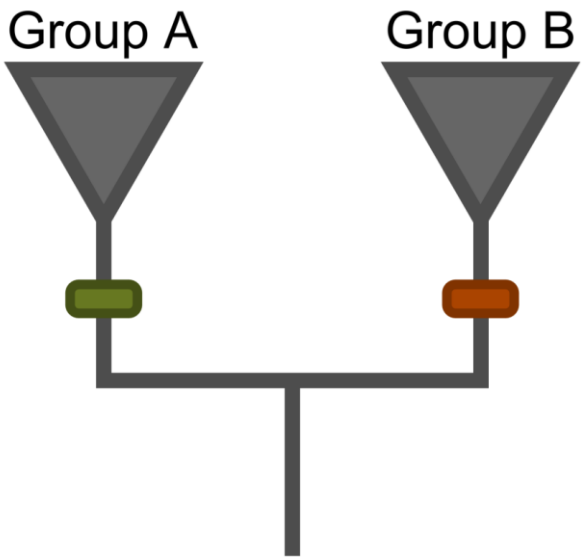


Figure 1:7 - An example of two fictional shared derived characters. Group A and B on their own describe two separate shared derived characters (SDC). However, they could also represent a symplesiomorphy.

An example of a well-studied SDC is present in the support for the monophyly of the bikonts (Stechmann & Cavalier-Smith, 2002, 2003). The gene fusion between the folate biosynthesis genes for dihydrofolate reductase (DHFR) and thymidylate synthase (TS) suggests a position for the root of the eukaryotic tree by excluding the root from the bikont group. The *CPS-DHO-ACT* tri-fusion also appeared to be a synapomorphy for the

unikonts (opisthokonts and Amoebozoa) (T. Cavalier-Smith, 2002). The animals, Amoebozoa and Fungi (collectively, the unikonts) along with the prokaryotes and Archaea, express separate gene coding regions conversely the plants, alveolates and the Euglenozoa (collectively, the bikonts) posses the gene fusion.

The unikont/bikont division which is neatly supported by morphological characters can be contested by the inclusion of an amitochondriate amoeba, *Breviata*, which was previously of an uncertain position within the tree of life (Roger and Simpson, 2009; Walker et al, 2006). This is due to it having a double basal body but where one of them is unflagellated. The problem arises as the unikont/bikont split relies on a distinction between single and double flagellated morphological characters. This highlights nicely the fragility of the grouping of taxa together into large groups or clades based solely on ultra structural characters and missing the phylogenomic information. However, this also indicates that the use of more characters does not necessarily help refine the definition of the major groups and branching order but can further complicate them.

1.4.1 Gene Fusion/Fission Events

Gene fusions are a hybrid of two or more previously separate open reading frames (ORF). They occur as a result of either a chromosomal translocation (the transfer of previously separate genes between chromosomes, Figure 1:8A), an interstitial deletion (leading to the removal of regions, for example a stop codon and promoter region, consequently concatenating two separate genes into a hybrid form, Figure 1:8B) or a chromosomal inversion (the reversal of a segment of the chromosome end to end, Figure 1:8C).

There is hugely conflicting evidence regarding the relative rate of gene fusions and gene fissions. Some estimates suggesting that up to two-thirds of all gene sequences consist of more than one gene fusion event (Teichmann, Chothia, & Gerstein, 1999), with analysis of the *Mycoplasma genitalium* genome supporting this figure with 2/3 of genes a product of a fusion event (Teichmann, Park, & Chothia, 1998). Further analysis of fusion gene families has suggested that specific arrangements of the conserved domains, within a fusion gene, tend to be conserved throughout the evolutionary tree. This suggests that: a) such gene fusions are the product of single events and b) that arrangement of conserved domains is subject to functional selection with alternative domain orders conferring different functional properties (Bashton & Chothia, 2002).

Extensive study of 2869 groups of multi-domain proteins demonstrated that reversions (or gene fissions) occur at an average frequency of one fission to every four fusion events (Kummerfeld & Teichmann, 2005). In contrast, Snel, Bork, & Huynen, 2000, find that fissions are more common across 17 prokaryotic species which they tested using a method adapting the Smith-Waterman algorithm (Smith & Waterman, 1981) to compare ORFs between genomes. Furthermore, gene fusions have been suggested to occur rarely, particularly among eukaryotic species that lack operons (Conant & Wagner, 2005; Kummerfeld & Teichmann, 2005) as it is thought that the proximity of operationally related genes may assist the creation of a fusion event. However, an earlier report suggested that gene fusions may be rather common among the eukaryotes since they already possess many genes with two or more conserved functional domains (Teichmann & Mitchison, 1999). These data also assume that a

gene fusion may occur because there is an advantage for the combination of functions that are linked biologically (Marcotte et al., 1999).

The synapomorphic character of a gene fusion, and therefore its presence or absence, can be used to provide support for the reconstruction of ancient evolutionary relationships where traditional phylogenetic methods have produced uncertain results. The gene fusion events can be used to build phylogenies to understand the evolutionary history of the gene fusion character and can be mapped onto a wider tree of life. This makes for a very useful method to confirm and potentially reorganise known phylogenetic topologies. The use of gene fusions, and other derived genetic characters for deriving evolutionary synapomorphies, depends initially on the usefulness of gene/genome sampling (Arisue, Hasegawa, & Hashimoto, 2004) but also upon the validation of four assumptions: 1) the genetic character has not undergone HGT; 2) patterns of paralogy and gene loss can be accounted for; 3) incidences of reversions can be excluded; 4) similar gene fusion characters have not been produced by convergent evolution. The above four factors can only be investigated using a combination of comparative genomics and phylogenetics (e.g. Brilli & Fani, 2004). However, some studies have only partially accounted for these four factors or assumed that they occur at a low frequency in eukaryotes compared with prokaryotes (Stechmann & Cavalier-Smith, 2002).

Gene fissions occur where a particular gene is cleaved in two producing two new and separate genes that can potentially code for two proteins. For a gene to be split and produce two new functional genes the inclusion of a stop codon followed by a

promoter region and a start codon between the coding sections must take place and the gene must remain 'in frame'. For this reason, it is hypothesized that, fissions should occur much less often than fusion events as it is not at all parsimonious for the inclusion of a promoter region and a start codon during this process (Stechmann & Cavalier-Smith, 2002) as discussed above. This scenario does not exclude the possibility of gene duplication and differential loss of domains, which can drive fission events and is likely to occur in eukaryotic genomes, which have highly duplicated gene families (e.g. Thomas A. Richards & Cavalier-Smith, 2005). Furthermore, it is also important to recognise that a single ancient gene fusion event has numerous opportunities as it is passed down through daughter cells and sister species to revert (fission event), hence one gene fusion can seed numerous fission events. Consequently, even though fission is a difficult evolutionary event in terms of direct gene splitting it is still unclear what the relative rates of these events are across the tree of life.

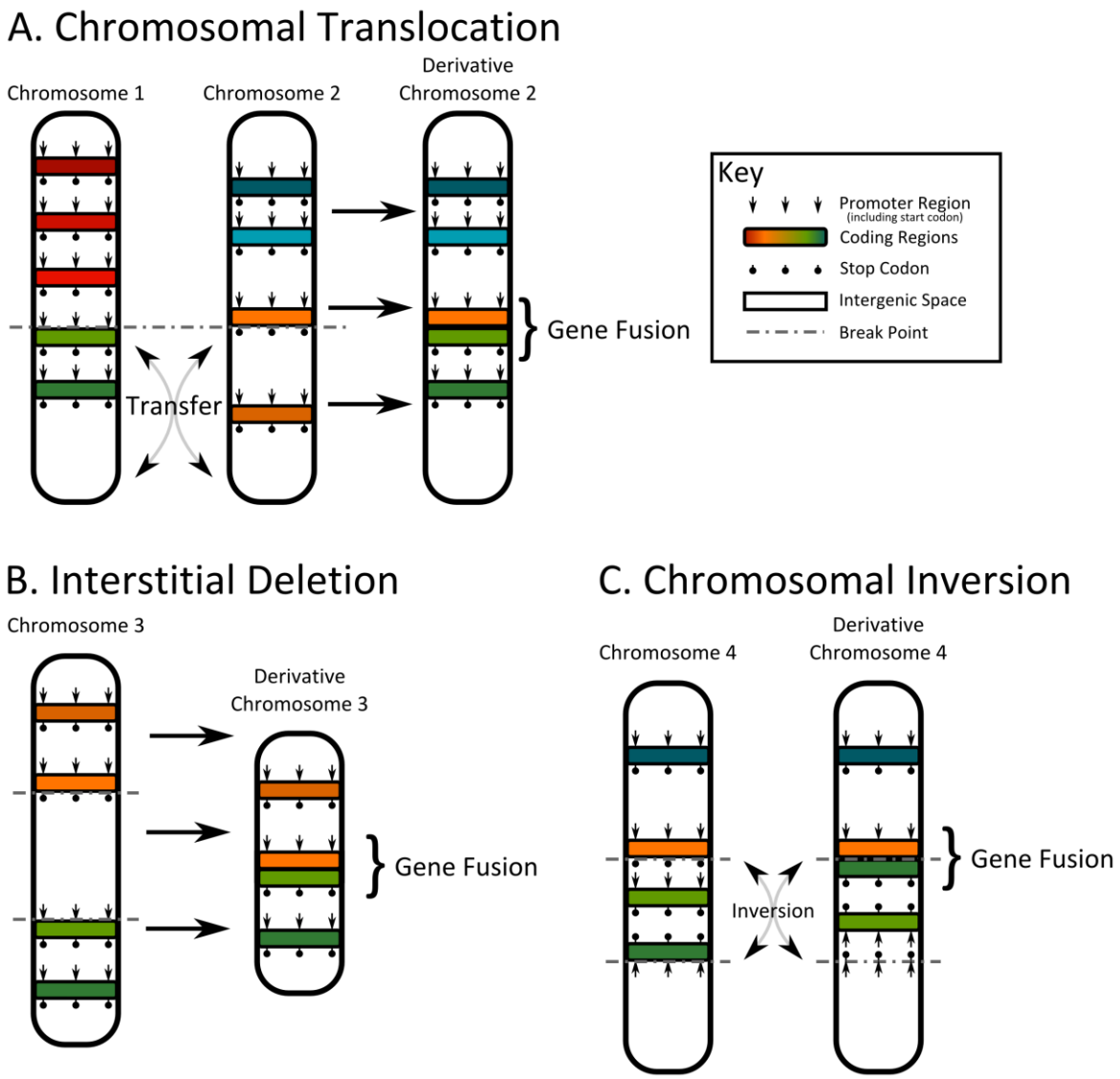


Figure 1:8 - A graphical representation of the three main processes that can result in the formation of a gene fusion event on a chromosome. The individual coloured rectangles represent an open reading frame, with the arrows representing promoter regions and the circles representing a stop codon.

1.4.2 Differential Relative Rates of Fusion and Fission across the Tree of Life

1.4.2.1 Example of Gene Fusion: Viridiplantae

Nakamura et al. (Nakamura, Itoh, & Martin, 2006) compared the genomes of *Oryza sativa* and *Arabidopsis thaliana* where they demonstrated that the rate of fission is generally higher than the rate of fusion between the two species studied; furthermore,

fissions were regarded to be higher in the *Oryza sativa* genome when compared to *Arabidopsis thaliana*.

Nakamura et al. (Nakamura, et al., 2006) achieved this comparison by identifying sixty candidate gene fusions between *Oryza sativa* and *Arabidopsis thaliana*. They performed multiple BLASTp comparisons and recorded all of the one-to-many orthologous pairs produced (Nakamura, et al., 2006). Of these sixty potential fusions and fissions, 39 were found to be cases where *Arabidopsis* presented composite genes with *Oryza* containing the split genes and 21 cases where *Oryza* contained the composite genes and *Arabidopsis* possessed split genes. These results show that fusions and fissions are slowly evolving rare events, which make them ideal in the resolution of ancient evolutionary relationships where standard phylogenies have provided inconsistent results.

1.4.2.2 Example of Gene Fusion: Fungi

Durrens, Nikolski, & Sherman, 2008, discuss detection and the rate of gene fusion and fission events between twelve fungal species by building an algorithm that groups sequence similarities between proteins and then subsequently aligns them in order to convert them into a Hidden Markov Model (HMM)[3]. The HMM was then used to search for composite and two-element groups of sequences.

[3] HMMs are statistical models built from dynamic Bayesian networks and are used to recognise patterns based on training datasets (Durbin, Eddy, Krogh, & Mitchison, 1998; Krogh, Brown, Mian, Sjolander, & Haussler, 1994).

This resulted in the prediction of 1,680, of what they call, elementary fusion and fission rates – these are unconfirmed fusions and fissions that were further sorted using other methods, including the use of PFAM domains. They find that fusions (a total of 376) were more abundant than fission events (a total of 294), in direct contrast to the Nakamura et al. results. This information allowed them to use the relative rates of gene fusion and fission within each genome to be mapped onto a previous fungal topology. They used a previously computed maximum-likelihood tree (Fitzpatrick, Logue, Stajich, & Butler, 2006), based on mutations of orthologous gene families, and plotted the counts of events onto the tree by changing the branch lengths of each taxon. This noticeably changed the dynamic of the tree and they suggested that this was evidence that the three species had undergone "massive genome reshuffling".

The results reviewed here suggest a contrasting pattern between the prokaryotes, which have operons, and the fungi, which appear to have a higher rate of fusion events compared to plants that appear to have a higher rate of fission events. These contrasting results require further investigation and an aim of this thesis is to identify relative rates of fission and fusion across the eukaryotes and how this relates to their validity as evolutionary synapomorphies.

1.4.3 Conserved Functional Domains

Comparative analysis of a wide range of genomes has resulted in the prediction of recurring units of sequence and structure within amino acid sequences. These recurring units are described as domains and they are each associated with discrete functions or 'code' for particular protein folds (three-dimensional structures)

(Marchler-Bauer et al., 2005). Often these domains are shared across multiple species from all branches of the eukaryotic tree of life; where this occurs and where they have very similar amino acid sequences, they are considered conserved. Subsequently, there are many databases that attempt to classify and predict these domains, for example the Conserved Domain Database (CDD) (Marchler-Bauer et al., 2009) at NCBI uses a matrix method to identify protein domains using an altered version of BLAST and draws its data from multiple projects (these methods are discussed in further detail in Chapter 2: Methods). They are a very useful tool when it comes to the identification of gene fusions as most gene fusions result from the joining of two or more of these domains (R. F. Doolittle, 1995). Therefore, if a sequence from an organism contains multiple domains and two or more sequences from another organism contain those domains separately the most parsimonious explanation is that they represent a fusion, potentially with multiple additional fission events, at some point in the evolution of that gene.

1.5 Current Gene Fusion Detection Methods

As discussed above in sections 1.4.2.1 and 1.4.2.2 the current methods for detection of gene fusions are limited to individual and partial datasets with the use of customised and specific bioinformatic pipelines. Both approaches are different and demonstrated opposite conclusions, it is reasonable to suggest that had each method been swapped and used on the other dataset (a swap between the fungal dataset and the plant dataset) a different set of gene fusions would have been identified. This is not to suggest that either method is incorrect, however, it promotes the thought that neither

set of results are directly comparable. Furthermore, one set is based on a very small selection of genomes while the other makes no attempt to validate the observations regarding the authenticity of each fusion gene identified.

Fusions may also be found serendipitously (Morris et al., 2009; Stechmann & Cavalier-Smith, 2002) for example by querying the *nr* BLAST and CDD databases to identify multiple conserved domains between paralogous sequences, though this is no easy task. Nevertheless, all approaches are satisfactory for individual projects, however, when they are applied on a wider scale involving multiple genomes, especially if they are not closely related, or if any comparison is to be drawn between different datasets then they are not effectively comparable. Therefore, this necessitates the fact that a different tool is needed, allowing for a standard set of results in order to make comparisons between groups possible and which was in part the motivation for this thesis. It is also worthy to note that any potential gene fusion that is found, by any of the methods outlined, needs further testing with standard phylogenetic reconstruction which will help to elucidate the nature of the gene fusion, polarise the fusion event and test for evidence of fission events.

1.6 Aims of this Thesis

1. To develop a method for the discovery of differentially distributed putative gene fusions (and fissions) shared between the predicted proteomes of a set of different taxa.

2. To test the fusion finding method developed by a comparison to a previously reported analysis of two plant genomes (Nakamura, et al., 2006).

3. To identify a robust phylogeny for the Discicristata, in order to be used as a foundation dataset for gene fusion comparison.

4. To further test the fusion finder method (fdfBLAST) against five sets of robustly inferred phylogenetic tree topologies (see aim 3). They will include four taxa each from five distinct groups within the eukaryotic tree of life.

5. To use phylogenetic methods to polarise gene fusion events and identify the relative rates of fusion and fission events across the eukaryotic tree of life. This is in order to understand how reliable the use of gene fusion events (synapomorphic characters) are for the polarisation of evolutionary ancestry.
    i. My introduction indicates contradictory results for the prediction of gene fusion and fission events – we will use this data to help understand how often the events occur.
    ii. What is the distribution of fusion and fission events across functional categories?

2 Methods

2.1 Homologous BLAST Searches

To investigate the evolution of all gene families reported in this thesis the program BLAST, basic local alignment search tool, (S. F. Altschul, Gish, Miller, Myers, & Lipman, 1990) was used to search databases containing the predicted proteomes and genomes of sequenced cellular organisms for similar sequences. BLAST is a heuristic[4] algorithm that searches for similarities between sequences by identifying high scoring pairs (HSP). These HSPs, sub-segments that share a high level of similarity, are scored by an algorithm that is based on the Smith-Waterman algorithm (Smith & Waterman, 1981), although it is significantly faster with the heuristics implemented within BLAST. Once HSPs have been generated between the query sequence and the sequences in each genome they are reported and ranked in a BLAST report.

BLAST can be used locally, with your own sequence database, or more often it is accessed online where multiple databases can be queried. For example the *nr* – originally, non-redundant – database includes all the RefSeq Nucleotide, GenBank, EMBL (European nucleotide database), DDBJ (Japanese nucleotide database) and PDB (Protein Data Bank) sequences managed by NCBI, but it does not contain any expressed sequence tags (ESTs) or unfinished high throughput genomic sequences. There are also other individual databases that exist, such as those found at the DOE JGI Genome Portal (containing 189 Eukaryotic taxa and 432 complete Prokaryotic

[4] A heuristic is a search technique that takes advantage of the idea that a problem need not be directly solvable in order to achieve a given result and where the technique relies on feedback to further increase the performance of the search.

microbial taxa), the BROAD Institute and the GeneDB hosted by Sanger which includes the Eukaryotic Pathogen Database Resource. The database TBestDB (TBestDB), or the taxonomically broad EST database, contains over sixty EST databases from a variety of microbial eukaryotes that do not currently have genome projects.

Throughout this thesis, unless otherwise stated, two particular versions of BLAST were used; firstly, tBLASTn in order to search for homologous translated nucleotides (where protein sequences were unavailable) and secondly, BLASTp for amino acid sequences. Throughout, and only when suitable, all the above databases will be sampled, except when an automated bioinformatics pipeline, discussed in Section 2.6.1 is employed as it makes use of its own set of predicted proteomes collated from NCBI GenBank and DOE JGI (The sequences used in this database are listed in Chapter 8).

On occasion a further version of BLAST can be used to achieve an increased sensitivity whilst searching; this version is called position specific iterative BLAST (PSI–BLAST) (Stephen F. Altschul et al., 1997), and is a way to detect highly divergent homologues. The main reason for using PSI-BLAST is to ensure that homologous gene sequences from certain genomes were not absent from any alignments due to limitations using the standard BLAST methods. The process is iterative and the user builds a position specific score, or profile, for an alignment generated from previous rounds of database sampling which is then used to perform further BLAST searches where conserved sequence motifs receive a higher score (Stephen F. Altschul, et al., 1997).

2.2 REFGEN and TREENAMER

Once the results of a BLAST analysis have been returned and where the appropriate top hits have been selected they are collated into a single file in the FASTA format. Each database sequence, as a part of the FASTA format, will have been given a long identification line, also known as the header or definition line. These lines always start with a greater-than or chevron symbol, '>', followed by other information, commonly database dependant and generally will include an accession number and a species name (species names from GENBANK are contained within square brackets, for example [*Homo sapiens*]). This type of labelled format is unfortunately incompatible with most phylogenetic analysis programs, as they only allow short identifiers, and therefore the sequence names require re-labelling in the alignment file.

If a phylogenetic analysis includes many hundreds of sequences and multiple taxon optimisation steps, this process can be extremely time consuming. Therefore, I have developed the program REFGEN (Leonard, Stevens, & Richards, 2009) which takes a FASTA sequence alignment file and converts each long sequence identification line, according to a set of user controlled renaming rules, into a 'REFGEN ID'. The REFGEN ID can be comprised of a combination of either the taxonomic name (genus and species, where available) and the accession number, as can be seen in Figure 2:1. Once the users analyses are complete and a tree topology has been produced the REFGEN IDs within can be re-labelled to their correct taxonomic classifications using the program TREENAMER, this is discussed more in Section 2.6.

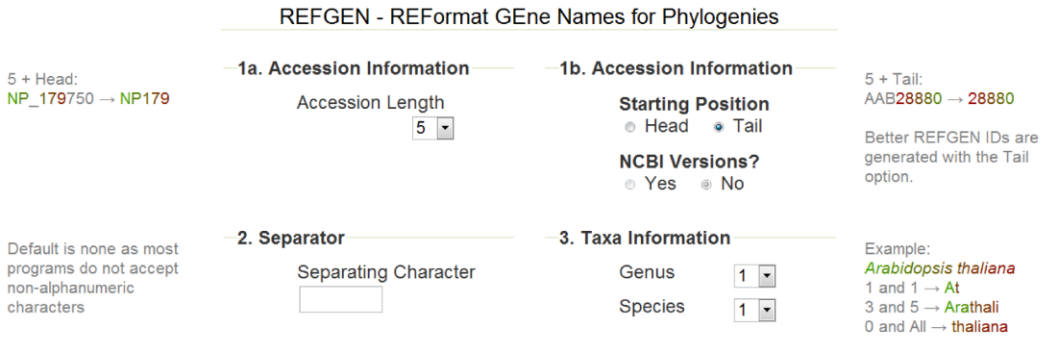
A

```
>gi|13359321|dbj|BAB33386.1| hsp60 [Paramecium caudatum]
KLFSNILQGKTLITTPAFFAGKELSFGQECRQQMLRGCDKLADAVQTTLGPKGRNVVIDQAFGGP
>gi|125542360|gb|EAY88499.1| OsI_009732 [Oryza sativa]
MAAANRGGEEQKTSMLWAPACKFSHRRQAATAATTELNIPDNLECPKSNFPISFGSGSDREGNCE
>gi|83775089|dbj|BAE65212.1| unnamed protein product [Aspergillus oryzae]
MQRALSSRTSVLSAASKRAPFYRSSGFNLQQQRFAHKELKFGVEARAQLLKGVDTLAKAVTSTLG
>gi|70947400|ref|XP_743319.1| hsp60 [Plasmodium chabaudi chabaudi]
MLSRLCGKTIHNGSTDKCVSLLNKIQKRNVAKDIRFGSDARTAMLIGCNKLADAVSVTLGPKGRN
>gi|1217626|emb|CAA65238.1| heat shock protein 60 [Euglena gracilis]
TMNRAGVLARRGYSSKGKDILFGVDARVKMLAGVNRLSQAVSVTLGPKGRNVVIEQPFGAPKITK
>jgi|Phyra1_1|71587|fgenesh1_pm.C_scaffold_29000006
MNPTLAVAVKKAARFSPAGRRLFSSGKDIRFGVEGRAAMLKGADQLANAVQVTLGPKGRNVVIDQ
```

GenBank

DOE JGI

B



C

```
>33861Pc
KLFSNILQGKTLITTPAFFAGKELSFGQECRQQMLRGCDKLADAVQTTLGPKGRNVVIDQAFGGP
>84991Os
MAAANRGGEEQKTSMLWAPACKFSHRRQAATAATTELNIPDNLECPKSNFPISFGSGSDREGNCE
>52121Ao
MQRALSSRTSVLSAASKRAPFYRSSGFNLQQQRFAHKELKFGVEARAQLLKGVDTLAKAVTSTLG
>33191Pc
MLSRLCGKTIHNGSTDKCVSLLNKIQKRNVAKDIRFGSDARTAMLIGCNKLADAVSVTLGPKGRN
>52381Eg
TMNRAGVLARRGYSSKGKDILFGVDARVKMLAGVNRLSQAVSVTLGPKGRNVVIEQPFGAPKITK
>71587P
MNPTLAVAVKKAARFSPAGRRLFSSGKDIRFGVEGRAAMLKGADQLANAVQVTLGPKGRNVVIDQ
```

GenBank

DOE JGI

Figure 2:1 - Input, output, and the interface of the program REFGEN. A, shows an example of a FASTA format file with 'deflines' from NCBI and JGI, sequences have been truncated. B, this is part of the new interface to the program REFGEN showing the different options available to generate a REFGEN ID. C, shows the output from REFGEN for the sequences in A, where they now possess a REFGEN ID based on five characters of the accession and one character each from the genus and species name. Image is adapted from Leonard et al. (Leonard, et al., 2009).

2.3 Alignment and Manual Masking

Before phylogenetic reconstruction can be attempted the set of sequences in your analysis must undergo a process called multiple sequence alignment (MSA). This is because sequence data retrieved from the various nucleotide and amino acid databases for each species will naturally vary in length and vary between individual sites. In order for an analysis to be compared by a statistical measurement it must be performed on the same set of data for each taxon, this is achieved by the introduction of sequence gaps. This process is achieved by aligning each sequence to another in the form of a matrix where each sequence is represented in a row and each site is represented in a column. The process is usually performed using a pair-wise process, comparing two sequences at a time, allowing for conserved regions and sites to be aligned, for a review of this and other methods please see (R. C. Edgar, 2010; R. C. Edgar & Batzoglou, 2006). MSA builds a picture of informative sites and is generally coupled with a process called masking. Masking allows the removal of potentially uninformative sites from the analysis by allowing the user to select only the conserved regions shared between all sequences by removing the sequence gaps (introduced by MSA and naturally occurring as *indels*) and any highly divergent sites which are difficult to align.

The program used in this thesis for all protein alignment purposes was MUSCLE (Robert C. Edgar, 2004) which is a multiple sequence comparison method that uses a log-expectation method for alignment. It calculates a distance measure between sets of unaligned and aligned sequences producing a distance matrix which is used to build

a guide tree. This process is repeated several times in order to refine the alignment until the refinement process can no longer produce a better alignment.

2.4 Substitution Model Prediction

Following the alignment and masking process each set of sequences is subject to an analysis to predict the best substitution model, alpha parameter (adjustment of the gamma distribution) and the rate of invariable sites if they are needed. As discussed in the introduction these variables allow for partial correction of signal-to-noise within the data. The models of substitution were designed to account for character changes in DNA or amino acid sequences and work by building a matrix based on a set of global alignments of closely related sequences and the probability of evolutionary changes observed between them (Dayhoff, 1965; Jones, et al., 1992). In this thesis all analyses are based upon amino acid alignments. There are now numerous models to choose from, for example, a recent matrix called the LG model (Se Quang Le & Gascuel, 2008; Si Quang Le, et al., 2008) was based upon nearly 4000 alignments from the PFAM (Sanger Protein Family) database.

Substitution model prediction was achieved for all my analyses with the program MODELGENERATOR (Keane, et al., 2006) which selects the most optimal amino-acid model based on the alignment. Although the tree building program RAxML (Alexandros Stamatakis, 2006), with the addition of my publicly available and widely used easyRAx script (http://projects.exeter.ac.uk/ceem/easyRAx.html), can also be used to find the best substitution model for protein sequence data we find that MODELGENERATOR

evaluates more models and parameters and includes more criterions to base the outcome upon.

2.4.1 Model Parameters

The program MODELGENERATOR (Keane, et al., 2006) also evaluates the best values for three other variables, the α-parameter (for the gamma distribution - Γ or G), the proportion of invariable sites (I) and the empirical base frequencies (F). The α-parameter applies a gamma distribution rate variation among sites, which changes the shape of the gamma distribution whereby as the α-parameter gets larger the range of variation between sites diminishes (Ziheng Yang, 1996). Invariable or invariant sites describes the amount of static or unchanging sites within a dataset and the parameter attempts to reduce the problems that are caused by long and short branches which can affect estimation of genetic divergences (Mike Steel, Huson, & Lockhart, 2000). Empirical base frequencies describe whether the model assumes an equal or variable frequency for the occurrence of nucleotide or amino acid bases and whether substitutions of these bases are equally likely or undergo variable rates of transition and transversion.

The model generator programme works by generating a phylogeny from the alignment using 96 different combinations of amino acid models and parameters. The trees are generated using a fast approach (the neighbour-joining, NJ, method). The resulting 96 phylogenies are compared using three information criteria (two Akaike - Akaike, 2003 and one Bayesian - Schwarz, 1978) which attempt to measure the fit given the

complexity of the model for an alignment. The models and parameters are subsequently ranked and the user is left to decide the best model.

2.5 Tree Construction Methods

Here I will discuss the various methods used in this thesis for building tree topologies from previously aligned and masked sequence data. There are two main methods used, firstly, maximum likelihood (ML) and secondly, Bayesian analysis. A few other specific methods were also used, when needed, and are discussed in Section 2.5.4. Multiple methods are employed in order to provide a comparison between the support for the topology of each tree, if multiple methods show a consensus topology and support for branching order then there is more likelihood for it being an accurate topology.

2.5.1 Maximum Likelihood Analysis

Maximum likelihood estimation (MLE or simply ML) evaluates a given hypothesis in terms of the probability that a proposed model or tree topology would give rise to the observed dataset being tested, thus the likelihood is the probability of the data given the model (Joseph Felsenstein, 2004). This method necessitates three variables, the observed data, a tree topology and a model of sequence evolution. The model of sequence evolution has already been chosen using the program MODELGENERATOR and the data is contained in the alignment file. The tree topology used is different for each program and the method used by each program will be explained under the headings below. After an initial tree has been built the likelihood of the given topology is calculated under the given model and parameters (P. G. Foster, 2001) and then the

tree topology and model can be refined using a series of heuristic based tree searching procedures. The problem of maximum likelihood (ML) inference of phylogenetic tree topologies is classified computationally as NP-hard[5] (Chor & Tuller, 2005) and is therefore a time-consuming and complex algorithm. This has led to the use of sets of heuristics to help expedite the determination of branching order and likelihood values, the term fast-ML is used in this thesis to describe this class of program.

2.5.1.1 PHYML

As discussed above PHYML (Guindon et al., 2010; Guindon & Gascuel, 2003) uses a fast ML approach to infer the phylogenetic tree branching order of an aligned dataset. Searching tree space in PHYML begins with the construction of a neighbour-joining (NJ) tree using the BIONJ algorithm (Gascuel, 1997). Then a set of optimisations are performed on the tree, originally PHYML only used the nearest-neighbour interchange (NNI) method but the newer version employs the use of subtree pruning and re-grafting (SPR) (Hordijk & Gascuel, 2005) to help minimise the effects of getting caught in local optima. The NNI method tries to improve a starting tree by exchanging two sub-trees connected by a single edge and calculating the likelihood of this rearrangement (Guindon & Gascuel, 2003). SPR creates sub-trees of a starting tree and then reinserts (re-grafts) them into another position within the starting tree.

[5] Part of computation complexity theory, NP hard (non-deterministic polynomial-time hard) states that a problem is at least as hard as the hardest problems in NP (those which have verifiable proofs within polynomial time) (Garey & Johnson, 1979).

2.5.1.2 RAxML

RAxML (Alexandros Stamatakis, 2006; A. Stamatakis, Ludwig, & Meier, 2005) is a program for sequential and parallel fast-ML based inference of large phylogenetic trees. Unlike PHYML it starts with the premise of taking an initial parsimony based tree based on your sequence data, as this is a relatively fast method for building a tree and therefore the process is repeated several times for one analysis resulting in multiple starting trees. These resulting trees are then used to compute a consensus tree and are used to improve the likelihood values of the final result. A further stage, optimisation of the tree topology, is described as a Lazy Subtree Rearrangement (LSR) mechanism. This mechanism is described to be closely related to the program called fastDNAml but differs slightly in its handling of the optimisation of branch lengths after subtree rearrangements have occurred and during the initial optimisation phase where improvements are immediately utilised in further optimisations.

To expedite the process of creating RAxML tree topologies I wrote a, now widely used, Perl script with an interactive menu which is freely available from http://projects.exeter.ac.uk/ceem/easyRAx.html. The standard operating procedure was to run ten best-known-likelihood trees followed by 1000 bootstraps including the predicted amino acid evolution model from MODELGENERATOR with an additional run with the PROTCAT substitution model (this additional run is used because PROTCAT is not evaluated as part of the MODELGENERATOR run but has several advantages over the homogeneous models suggested by MODELGENERATOR. In other words PROTCAT does not use a standard model approach but is a heterogeneous model and allows different models to underscore different parts of the phylogeny). This is option 11 in

easyRAx. It is worthy of note that RAxML insists on calculating its own values for gamma and fixes the number of categories to four, the reasons for this are discussed but not necessarily justified in an author monologue in the RAxML 7.0.4 manual.

2.5.2 Bayesian Analysis

Bayesian analysis (Bayesian inference, BI) differs to ML analysis in that it is based on the concept of evaluating posterior probabilities. These probabilities are based on the outcome of a model or a prior expectation given knowledge of the data (Rannala & Yang, 1996). That is to say it is used to identify the tree that maximises the probability of the tree given the data and model for evolution (J. Archibald, Mort, & Crawford, 2003). This allows for a set of best trees to be calculated and optimised within the tree space (described as a landscape of hills and valleys), this is deemed as a better process than ML as the methods employed can jump the analysis from a suboptimal hill to a better optimum where perhaps ML would get stuck unable to cross the valley (Huelsenbeck, Larget, Miller, & Ronquist, 2002).

The method that helps achieve this is a Markov Chain Monte Carlo (MCMC) simulation (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) which is a set of independent searches that perform information exchange. They work by gathering a random sample of trees together and analysing the branching order probabilities between them. The Markov chain is used to sample from the posterior distribution in order to produce the group being tested by selecting a tree based on the current tree and not the previous state if an improved tree is detected. If, for example, two species are monophyletic in 90% of the sample, then this suggests that the probability for this

grouping is 90% but only if the group sampled is large enough, hence the Monte Carlo reference as it is a gamble. The thought behind the use of MCMC sampling is to jump around the tree space in such a way that trees are sampled according to their posterior probability and therefore a region with a very high posterior probability will be visited more frequently.

2.5.2.1 MrBayes

The program MrBayes (Ronquist & Huelsenbeck, 2003) implements an algorithm based on the MCMCMC processes described above. The extra MC in this process is Metropolis coupling and it is used to increase the number of alternative data points investigated and therefore improve the chance of convergence of the Markov chains.

For all analyses utilising MrBayes3 I will have used the following command blocks to generate a phylogeny.

```
Begin mrbayes;
set autoclose=yes;
log start filename=YOUR_FILE.log replace;
prset aamodelpr=fixed(wag); [as MODELGENERATOR or aamodelpr=mixed]
lset rates=invgamma Ngammacat=8; [as MODELGENERATOR]
mcmc ngen= 1000000 printfreq=100 samplefreq=100 nchains=4 savebrlens=yes
startingtree=random filename=YOUR_FILE.nex;
quit;
end;

Begin mrbayes;
log start filename=YOUR_FILE.log replace;
sumt filename=YOUR_FILE.t burnin=XXX contype=allcompat;
end;
```

These two command blocks tell MrBayes to run an analysis with 1 million MCMC optimisations with the relevant models of evolution as predicted by MODELGENERATOR. The second command block relies on a ‘burnin’ value which can

be calculated from the probability file output from the first run. This essentially discards the first n amounts of trees because the trees before this value in the analysis are untrustworthy as we are unable to say whether they are stuck in a local minimum or not.

2.5.3 Bootstrapping

Bootstrapping (Joseph Felsenstein, 1985) is a test of reliability of an inferred phylogenetic tree topology. It functions on the basis that if there are a set of sequences 'm', each with a known number of sites 'n' then a tree topology can be reconstructed, for example by ML or Bayesian inference. Then from each sequence contained in 'm' a number of alignment sites are randomly replaced within 'n' consequently constituting a new data matrix generated from the original data matrix and then used to infer a tree topology. The two tree topologies are now compared, whereby the internal branching order is examined and those that differ between the original and bootstrap tree are given a score of zero (as they disagree on taxon position), all other branches are given a score of one. This process of re-sampling and ensuing tree inference is then repeated many times (a good analysis should cover at least 1000 bootstraps[6]) and the percentage of times an interior branch is given a 1 is noted. Once the process has finished the final percentage for each internal branch is given as the confidence for that branching order.

[6] This is my suggestion; however the number of bootstraps used depends on a number of conditions, for example: the size of the alignment in characters and the complexity of the data. A fifty character dataset sampled 1000 times would be overkill, as the re-sampling process would be repeated multiple times.

2.5.4 Approximate Likelihood-Ratio Tests (aLRT)

Approximate Likelihood-Ratio Tests (aLRT) are a further method for identifying topology support compared to bootstrapping. They work by calculating the log-ratio between two trees. The most common is the Shimodaira-Hasegawa (SH) test (H Shimodaira & Hasegawa, 1999) which is implemented in a version of PHYML (Hordijk & Gascuel, 2005) as the SH-like test and also forms part of the process in FastTree2 (M. N. Price, Dehal, & Arkin, 2009; Morgan N. Price, Dehal, & Arkin, 2010).

2.6 Drawing Trees

There are several programs that are able to draw graphical representations of the tree topologies generated by the previous methods. For the figures in this thesis I primarily used the program *FigTree* (Rambaut) but on occasion other programs were used to very quickly view a tree, *njplot* (Perrière & Gouy, 1996), or to quickly generate an SVG file of the tree, *TreeVector* (Pethica, Barker, Kovacs, & Gough, 2010), or to plot multiple bootstrap support values onto one topology *TreeGraph2* (Stover & Muller, 2010). The end result, in this thesis, will always be a vector image that is then annotated using a vector image editing program and exported as a high-resolution portable network graphic (PNG). I would also like to draw the reader's attention back to Section 2.2 as at this stage the program TREENAMER should be used to annotate the *Newick* or *Nexus* format tree files with each sequence's correct taxonomic classification.

2.6.1 Automatic Tree Construction Pipeline

A bioinformatic pipeline, referred to as 'Darren's Orchard', and first published in T. A. Richards et. al., 2009 can be used to generate a phylogeny for every predicted protein sequence within a genome. For example, in the case of a set of fusions predicted by my program fdfBLAST each sequence of a predicted protein containing a fusion event is placed into a file. Subsequently, each sequence in the file is automatically subjected to a BLASTp analysis against the whole predicted proteomes of 795 species collated into a MYSQL database. This includes all eukaryotic genomes contained in the GENBANK nr database and a sampling of those from the DOE JGI Genome Portal and the BROAD Institute (listed in Chapter 8). We then used an e-value cut-off of 1e-40, appending every hit from the 795 proteomes to a further file. Each group of sequences is then automatically aligned using the program MUSCLE (Robert C. Edgar, 2004) and then GBLOCKS (Castresana, 2000; Talavera & Castresana, 2007) is used to extract highly conserved regions and remove any alignment gaps. The original pipeline then uses PHYML (Guindon, et al., 2010; Guindon & Gascuel, 2003) to construct phylogenetic trees for each set of aligned and masked sequences using a WAG + Γ + I substitution model (Γ + I as estimated by PHYML) and with a fast approximated likelihood ratio test (aLRT) method for evaluating topological support. However, in order to cope with the large amounts of data predicted by fdfBLAST the program FastTree 2 (M. N. Price, et al., 2009; M. N. Price, et al., 2010), which uses a set of fast heuristics and the SH-like aLRT algorithm to produce a topology, was substituted and run with the 'slow' option. This still produced well supported tree topologies but up to 100 times faster. The resulting trees are then converted into images; this process is described in Section 2.6.2.

2.6.2 Tree Files

The program *Dendroscope* (Huson et al., 2007) can be used in a 'command-line' mode and therefore can be effortlessly utilised in bioinformatic pipelines. For this thesis, it was used to automatically generate rectangular phylogram tree topologies in two image formats[7]: the scalable vector graphic (SVG) and portable document format (PDF). The tree topologies produced were based on the trees generated by Darren's Orchard pipeline (T. A. Richards, et al., 2009) from the sets of putatively predicted gene fusions as predicted by my program fdfBLAST.

The resulting SVG files were then edited with a script (Wickstead, Gull, & Richards, 2010) kindly provided by Bill Wickstead (University of Oxford), and with additions made by myself, which automatically annotates each leaf node of the tree topology with a graphical representation of associated predicted functional domain architectures. The PFAM or CDD parsed domain files from the fdfBLAST analyses can be used with this script, the process for generating these files is described in Chapter 3. Some minor modifications were made to the script. For example, the colour of a specific domain which is shared between multiple datasets was originally randomly assigned and therefore analysis was made difficult, due to the inconsistent colours, during tree comparisons. In order to produce consistent domain colours I altered the code to produce a colour based on the name of the domain, this allowed for the same

[7] The command-line code used to generate these images is shown below, format=SVG is substituted for format=PDF to change the image format.

```
Dendroscope +g false –x "open file=FILENAME;set drawer=RectangularPhylogram;ladderize=left;zoom expand;select labelnodes;set labelcolor=255 0 0;deselect all;select leaves;set labelcolor=0 0 0;deselect all;find searchtext=FILENAME target=Nodes;set labelcolor=255 0 0;deselect all;exportgraphics format=SVG replace=true file=FILENAME.svg;quit;"
```

colour to be used for the same domain between trees and datasets automatically. Subsequently, these SVG files were further edited using *Inkscape* (GPLv2) or Adobe® Illustrator® and then used to create a PDF or portable network graphic (PNG) for inclusion in standard word processing documents.

3 Fusion and Duplication Finder BLAST (fdfBLAST) – a tool to predict differentially distributed putative fusion and duplication events between proteomes.

3.1 Introduction

Gene fusions are a hybrid of two or more previously separate open reading frames (ORFs). They occur as a result of either a chromosomal translocation (the transfer of previously separate genes between chromosomes), an interstitial deletion (leading to the removal of regions, for example a stop codon and promoter region, consequently concatenating two separate genes into a hybrid form) or a chromosomal inversion (the reversal of a segment of the chromosome end to end), please see Chapter 1.4.1 for more details.

Gene fusions are suggested to be rare, particularly among the eukaryotic species that lack operons and have separately transcribed genes with introns and distinct promoter regions (Conant & Wagner, 2005; Kummerfeld & Teichmann, 2005), although in contrast an earlier report suggested that they may be rather common among eukaryotes since they possess numerous genes with two or more conserved functional domains (Bateman et al., 2004; R. F. Doolittle, 1995; Teichmann & Mitchison, 1999). Nevertheless, they can be used as both tools for functional prediction and as evolutionary markers to investigate the relationships between sets of taxa (Rokas & Holland, 2000). In this context they are more commonly described as shared derived characters (SDCs) or synapomorphies; that is they are shared by two or more taxa and

only those taxa. In such a case they can be used to establish phylogenetic topologies by allowing the formation of clades sharing those characters and excluding all other taxa that do not possess those characters. For example, as we know that gene fusions are formed from previously separate genes, it follows that any taxa which possesses a copy of a gene fusion are therefore, under the most parsimonious explanation, monophyletic while taxa that have the unfused forms must branch separately (Rokas & Holland, 2000).

This makes for a very useful method to confirm and potentially reorganise previously computed phylogenetic topologies. Indeed, this is the case with the gene fusion DHFR-TS that supports the monophyly of the bikonts (Stechmann & Cavalier-Smith, 2002, 2003). They allow a distinction to be made between a set of genomes separating them into distinct clades; those with the gene fusion character and those with the unfused versions. Moreover, clades possessing a fusion character are described as holophyletic (strict monophyly under parsimony) and so exclude the root.

Currently there is no standardised and automated method to search for gene fusions across multiple whole genome datasets, the most commonly used method is by manual inspection of open reading frames (ORFs) from each taxon using genomic tools such as BLAST (S. F. Altschul, et al., 1990), RPS-BLAST (Marchler-Bauer, et al., 2009; Marchler-Bauer & Bryant, 2004) and the NCBI COGs database (Tatusov, Koonin, & Lipman, 1997). Consequently, there have been very few actual predictions of their occurrence across the available eukaryotic genomes. Nevertheless, one notable study investigates the two land plants *Arabidopsis thaliana* and *Oryza sativa* (Nakamura, et

al., 2006). This study identified sixty candidate gene fusions between *Oryza sativa* and *Arabidopsis thaliana* in a whole genome analysis by performing multiple BLASTp comparisons.

3.2 Aims

The use of rare genomic changes is an important tool which helps to enhance traditional phylogenetic methods by resolving the branching relationships of taxa (Rokas & Holland, 2000). However, the potential rate of gene fusions has been conservatively predicted to be quite low, mainly because they are difficult to find amongst the relatively large number of genes present in the genomes of available sequenced taxa and so given these obstacles they are therefore a prime candidate for bioinformatic retrieval and prediction.

Consequently, I have designed an automated bioinformatic method to discover differentially distributed gene fusions in whole genome datasets. This was achieved by comparing the predicted proteomes of eukaryotic taxa using sequence similarity comparison tools (e.g. BLAST) to identify and compare differentially distributed gene families in terms of open reading frame (ORF) number followed by a secondary step in the bioinformatic pipeline which maps conserved functional domains (using HMMER with PFAM and/or RPS-BLAST with CDD) to identify differentially distributed gene fusions. We call this program fusion-duplication-finder BLAST (fdfBLAST). Finally to measure the efficacy of this approach an independently identified dataset (Nakamura, et al., 2006) was re-evaluated using BLASTp comparisons of the plant genomes *Oryza*

sativa and *Arabidopsis thaliana* and subsequently compared with the fdfBLAST pipeline results from the same two genomes (see Chapter 4).

3.3 Materials and Methods

The development process of this bioinformatics pipeline utilises a Perl script to compare multiple predicted proteomes with the aim of identifying differentially distributed gene fusions. This pipeline is composed of five interlinked steps which are explained in detail in the following sections. Each step shows a graphical representation, including a short explanation, and a more thorough explanation accompanying it below each figure.

The fdfBLAST program was developed under a free and open-source Ubuntu Linux environment (http://www.ubuntu.com), although it should be capable of running on any UNIX based operating system (including Apple OSX) where the following programs are available for your system. Perl v5.10.0 (http://www.perl.org) was used to write the main portions of the script. The legacy NCBI BLAST executables (version 2.2.21) are used to perform the BLASTp and RPS-BLAST (Marchler-Bauer, et al., 2005) searches and to prepare the proteome databases ready for use.

A new version of the local BLAST executables is now available, called BLAST+ executables, but they were not released at the inception of the program – there is no reason why fdfBLAST could not be adapted to utilise these in a future release – although there is no pressing need to change the script currently since no particular

advantage is conferred. Later portions of the script employ the use of the program HMMER3 (http://hmmer.org) and a copy of the Sanger Institute's PFAM database (Pfam 24.0 - October 2009; Finn, 2010) and the NCBI Conserved Domain Database (CDD version 2.2.2).

3.3.1 Step 1: Automated Serial BLASTp Comparisons

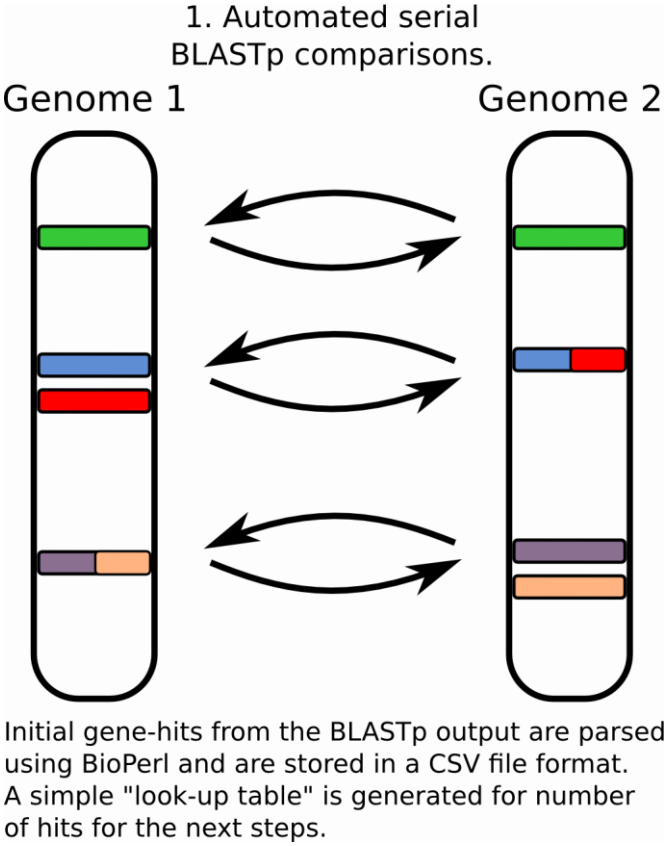


Figure 3:1 – Step one illustrates the process where genomes are cross compared using automated Serial BLASTp comparison to begin to identify differentially distributed gene families.

Predicted proteomes of each of the taxa required for a comparison analysis to identify potential gene fusions events were retrieved from their respective genome projects or the respective NCBI database (ftp://ftp.ncbi.nih.gov/genomes) in the FASTA format. Manual minor alterations were made to the definition lines of each gene contained in each predicted proteome; all information but the accession is removed and a shortened version of the species name is appended, this allows for more efficient parsing of the data. For example, the *Homo sapiens* protein with an accession of *NP_002007.1* in the FASTA format is given as "`>gi|60097902|ref|NP_002007.1|`

filaggrin [Homo sapiens]" and is reformatted to ">NP_0020071Hs". This process can be automatically completed by running a local version (allowing for more than a total of 50 sequences) of the REFGEN tool that I developed (Leonard, et al., 2009) and previously discussed in Chapter 2.

The predicted proteomes are then assembled in a subdirectory of the fdfBLAST program called 'genomes' and subsequently in their own directory with a user given name based on the current analysis. Following this, the program ***formatdb*** from the NCBI local blast package is executed via a command in the fdfBLAST Perl script on each predicted proteome; formatting them ready for running a local BLAST analysis. The program ***blastall***, from the same package, is then executed in the same way performing a set of serial-sequential BLASTp analyses (with standard BLASTp settings) between the proteomes. For example, the three genomes of *Trypanosoma cruzi* (Tc), *Trypanosoma brucei* (Tb), and *Leishmania major* (Lm) are compared in this way; Tc to Tc, Tc to Tb, Tc to Lm and then Tb to Tb, Tb to Tc, Tb to Lm and then Lm to Lm, Lm to Tc and Lm to Tc. Each output file is recorded in a directory called 'g2gc' (genome to genome comparisons) in the style, "Tbrucei_Lmajor.bpo" where they contain a plain text version of the standard BLAST output.

3.3.2 Step 2: Comparative Hit Counts and Identification of Differential Distribution of Hit Numbers

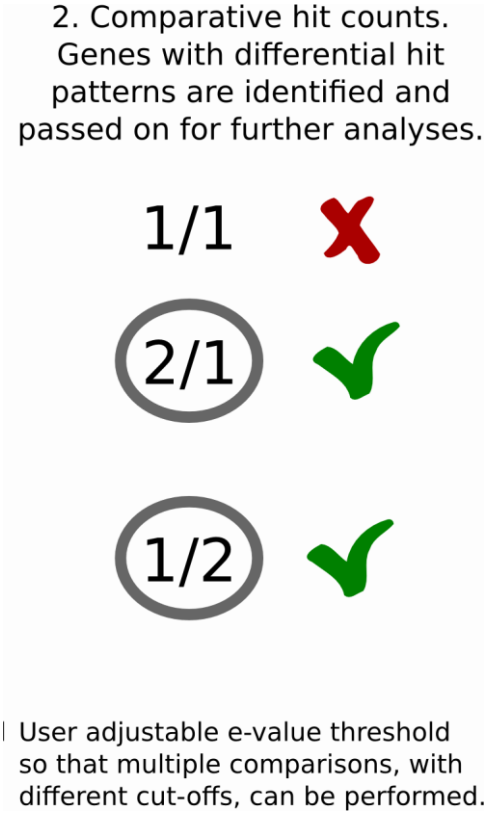


Figure 3:2 – Gene families that have differential comparative hit counts are recorded

Once the BLASTp comparisons are complete they are parsed using the Bio::SearchIO method available in the BioPerl project (Stajich et al., 2002), which contains several methods for parsing BLAST output efficiently. The user is invited to specify an e-value threshold range to scan the BLASTp output at, e.g. between 1e-10 and 0. The user adjustable e-value selection is a threshold adjustment which affects the total number of putative differential fusions/duplications in the output in order to allow the user to manage noise manifested in the form of false positives sampled using a too liberal gathering threshold. This facility allows the user to study how the putative fusion

profile changes at different thresholds. Selecting an e-value closer to zero (e.g. 1e-70) theoretically will produce fewer overall hits than choosing a lower e-value.

Therefore, we consider that a good analysis will include a range of e-value comparison runs of the program fdfBLAST for the previous reasons, although the process is time consuming, especially for larger genomes, and so one e-value range may only be completed. Following the user selection my program fdfBLAST parses the data and collates any 'query gene' to a series of subject/hit comparisons within that range. The data is stored in a comma-separated values (CSV) file in the initial folder in the following format;

```
number of hits, query accession, query length, hit accession, hit length, e-
value, % identity, hit-range start[8], hit-range end
```

For example:

```
1,NP_0753831Hs,463,NP_8487741Mm,427,0.0,74,47,462
```

This indicates that the gene NP_0753831Hs, *Homo sapiens*, with a length of 463 amino acids has one BLAST hit against the gene NP_8487741Mm, *Mus musculus*, with a length of 427, an e-value reported as 0.0 and a percentage identity of 74%. The hit range shows the range where the subject open reading frame matches to the query ORF, they are used in the program for the graphical output and the matching of subject hits to queries in putative gene fusions. As a part of this process a secondary

[8] The hit range represents the starting and ending position of the amino acid characters from the high-scoring pair (HSP) from the BLAST alignment for the gene that is being searched against in a BLAST format database.

file, in the form of a 'look-up table', is generated where the subject hit information is organised for each genome (again stored in a CSV file), this file is utilised in the extraction of differential hits in the next stage of comparison. Each row in the 'look-up table' corresponds to a gene in the order they appear in the predicted proteome FASTA database, columns correspond to each proteome alphabetically as they appear in the file systems' directory.

For example, where the two genomes, *Arabidopsis thaliana* and *Oryza sativa* are being scanned an output for the first three genes may look like this;

```
1,1,
1,0,
1,2,
2,2,
```

The first row, represented by "1,1" shows a self-hit for the first gene in *A. thaliana* and a hit to one gene in *O. sativa*, the identity of the gene can be found by querying the previous initial parsed CSV file (this can be performed manually if the user wishes, although fdfBLAST handles this automatically) and will most likely represent an orthologue (although alternate evolutionary scenarios, for example hidden paralogy, are possible). The second row (the second ORF) has a self-hit in *A. thaliana*, this should always be the case (where self-comparisons are performed) but no BLAST hit within the user-specified e-value range in *O. sativa*. The third example shows a differential hit, our main targets, where the first gene in *A. thaliana* is found to have BLAST hits to

two genes in *O. sativa*. The fourth demonstrates two self-hits and two hits to the *O. sativa.* A number of scenarios are possible here: 1) two sets of orthologous genes (potentially sister orthologues) which would suggest an absence of a differentially distributed gene fusions or duplication. 2) Alternatively within this 2 to 2 hit there could be a differential duplication or gene fusion, hidden in part by gene loss. Currently, for these kinds of results it is very difficult to separate scenario 1 and 2. Therefore, these at present are discounted by fdfBLAST as we focus on easily verifiable differentially distributed gene fusions and duplications. Furthermore, multiple self-hits within the same genome can exist and potentially infer that a gene duplication has occurred or that a gene fusion has occurred. This data is not currently collected as we focus specifically on gene characters that show differential distribution patterns and minimised the chance of sampling highly paralogous gene families.

3.3.3 Step 3: Reciprocal Hits

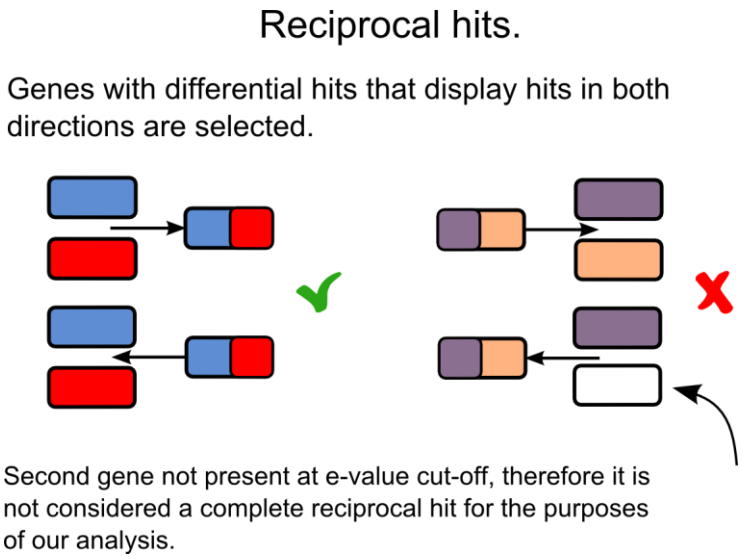


Figure 3:3 – Differential distributed genes identified by pair wise comparisons (step 1) are confirmed by comparison of reciprocal blast hits.

The next stage performs a comparison of reciprocal hits. For each gene that has displayed evidence of differential hit patterns the reciprocal file is queried to see if the differential pattern is preserved when the analysis is performed in the opposite direction. Therefore, if a gene in *O. sativa* is recorded as having two specific hits in *A. thaliana* but at the user specified e-value the reverse is not true it is discarded.

However, if the reciprocal hit does remain the genes are likely to provide further support for a differential distribution that could be the consequence of a gene fusion event. This stage, although potentially computationally exhaustive, reduces the data sampling significantly. In real data many hits are in the tens to hundreds rather than the slightly contrived two in the example given here, this stage removes unwanted or

unlikely hits by attempting to remove sources of noise and false positives. The result of this process is a file listing all the one-to-many hits as a list: for example;

```
0;NP_0091684Hs;867;XP_9062361Mm;402;1e-117;53;442;864;
1;NP_0091684Hs;867;XP_0014757691Mm;322;6e-33;47;10;199;
```

In this example we can see the two hits (in blue), numbered 0 and 1, are both the same query gene but match to two separate hits (one-to-many) that have been confirmed reciprocally (green). The data file also contains other relevant BLAST data such as gene length, e-values and hit-range.

3.3.4 Step 4: Rank and Sort

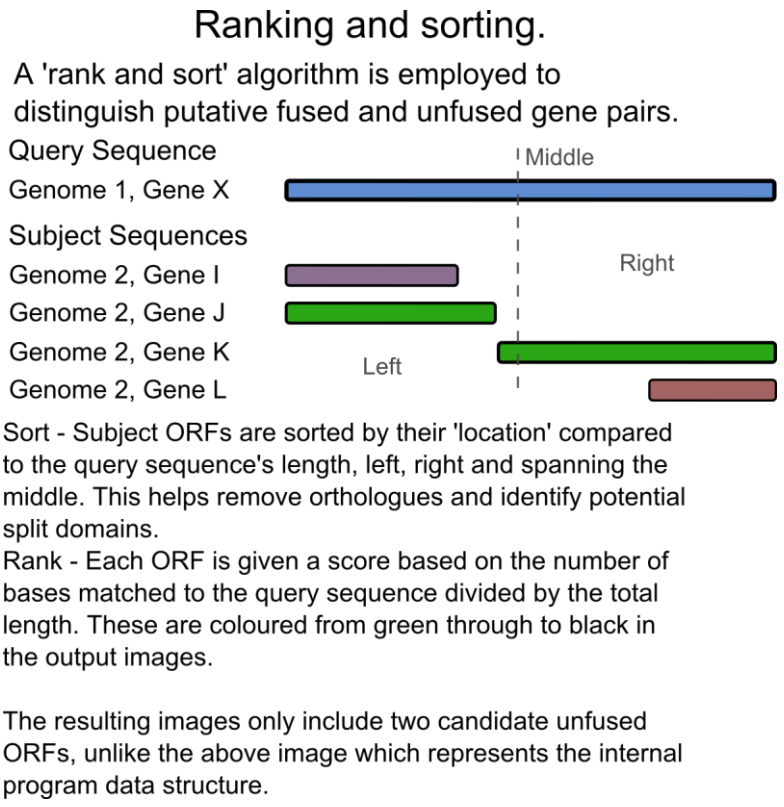


Figure 3:4 - Rank and Sort

The next stage helps to further reduce the dataset to robust examples of differentially distributed fusions dataset by subjecting the data to a rank and sort algorithm. The query sequence in Step 3, NP_0091684Hs (coloured blue), is for the purpose of this analysis given as the composite gene (the fused state) and the two subject hits, both XP_9062361Mm and XP_0014757691Mm (coloured green), represent the split state.

3.3.4.1 Sorting of Data

Using the hit range information from each gene, their 'position' compared to the query sequence's length and ‘middle’ (length divided by two) can be measured. Subject hits

are classified (sorted) as either left of the middle, right of the middle or spanning the middle. Note that hits which have >89% of the region of similarity which is either 'right of middle or left of middle are not classified as middle. This is currently a default value but future iterations of the program will allow this to be user defined. Subject hits that span the middle and that have a user defined percentage similar length (currently fixed to 90% similar) to the full length and longest gene are removed; they almost certainly represent complete homologous genes that share similar patterns of sequence similarity across their ORF arrangements and are therefore unlikely to be differentially distributed gene fusions.

3.3.4.2 Ranking of Data

The ranking step is further split into two ratio score stages to help extract potential fusions from the data. The final split sequences are ranked in two ways; firstly each is given a percentage score based on the number of amino acid bases matched to the query sequence's length. Each potential unfused ORF alignment is then associated by a colour - 80-100% being green, 70-80% as light blue, 60-70% represented by purple, and dark red for 40-60%, and finally grey for less than 40% (short) matches – the colour scheme is reflected in the final alignment images (Figure 3:5).

Secondly, a ratio is calculated based on the remaining ORFs matched to the query sequence; this is achieved by ordering the lengths of the matched ORFs (left and right matches separately) from shortest to longest and calculating a score for each pair of the matched ORFs. The largest end value from the left set is divided by the lowest start value from the right. In Figure 3:5, the largest end value was 224 and the lowest start

value from the right was 370, resulting in a ratio of 0.6. All combinations can be output by fdfBLAST which are organised into folders ranging from 0.1 to 1.0 stepping up in increments of 0.1, when fdfBLAST is run the user is encouraged to select a threshold for this ratio. All fdfBLAST runs undertaken as a part of the analyses enclosed within this thesis covered the full range from 0.1 to 1.0.

These two methods, although seemingly quite complex, are followed in order to make sense of the data produced by fdfBLAST. As gene fusion events can be considered the product of the union of multiple domains, it is advantageous (at least programmatically) to categorise the location of matched split-ORFs to the potentially fused-ORF state, this helps with the identification of candidate split-ORFs. For example, if all the matched ORFs for one fused ORF were to be similar in length (and span the whole putative fused ORF) then they can be identified as potential complete-gene-length homologues and so discarded. Similarly, if all the matched ORFs appeared to be 'one-sided' (only match to one half of the putative fused ORF) then the putative gene fusion is likely to be an artefact (or has suffered from secondary loss of a domain or is missing as a predicted discrete conserved domain in PFAM or CDD) and can also be discarded. It is also advisable to suggest a statistic (as a basis for comparison) in the prediction of a gene fusion event and these two methods attempt to provide this on two levels; the overall match for each of the pair of matched ORFs (reflected by their colour) and a general ratio for their fit compared to the putative fused state.

3.3.4.3 Graphical Representation of Ranked and Sorted Data

Internally, to the fdfBLAST program's data-structure, there are many more matches than the two shown in Figure 3:5. This is somewhat demonstrated in Figure 3:4 where fdfBLAST has generated several matches for the original query sequence indicating there are more than two matched ORFs/domains. If a match has a low ratio, e.g. 0.2, then it is not likely to be a good quality candidate for a gene fusion event. However, these low ratios should not be discounted immediately; due to the algorithms obvious limitations, categorising by left, middle and right, only fusions consisting of two ORFs/domains are detected because fusion genes composed of three or more differentially distributed domains will be excluded. Therefore, lower ratios may indicate gene fusions of more than two ORFs/domains. For example a third ORF may be present between the two predicted split ORFs/domains for a fused query (although it is not the actual case, a third ORF could exist between the two ORFs present in Figure 3:5 from amino acid positions 225 to 369), if this is the case then the user can refer to the differential hits file and extract the extra information manually.

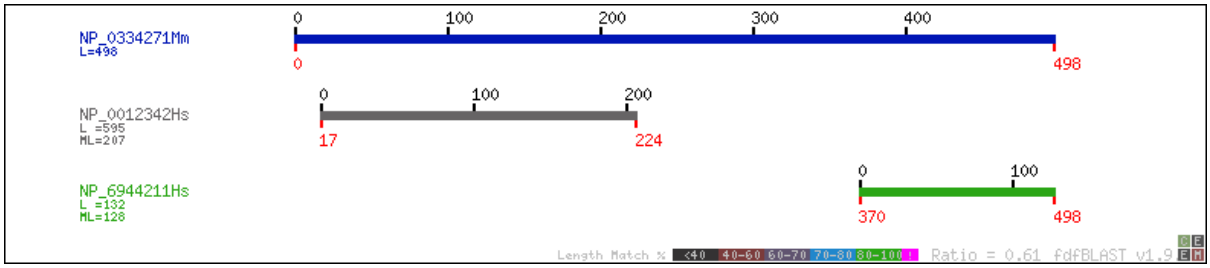


Figure 3:5 - An example of fdfBLAST's output for a comparison between the predicted proteomes of *Mus musculus* and *Homo sapiens*. A putative fusion event has been established in the mouse genome (blue line) and two hits are present in the human (the grey and green line). The image awaits annotation of conserved domains.

3.3.5 Step 5: PFAM and CDD – Identification of Discrete Functional Domains to Confirm Fusions

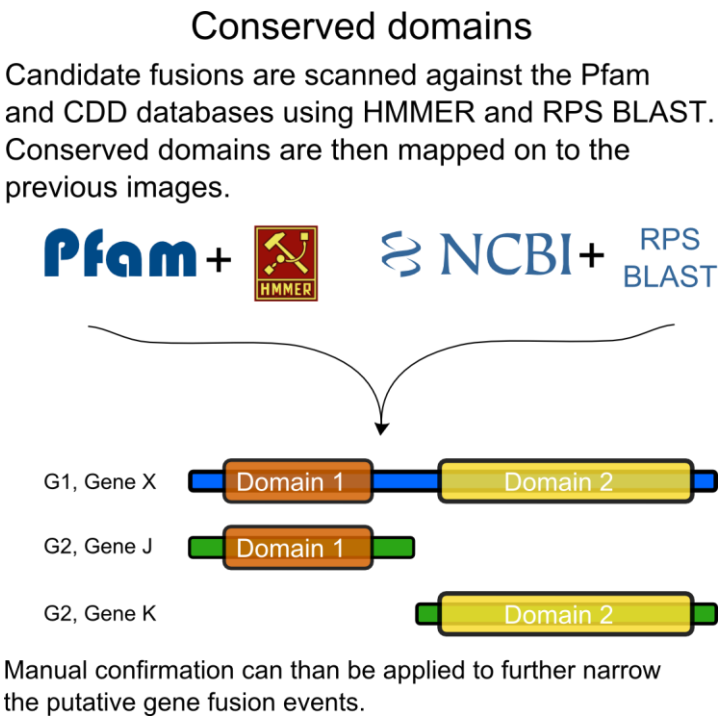


Figure 3:6 - PFAM and CDD Domain Annotation

The sequences, representing the fused ORF and the two best unfused ORFs, from each set of candidate gene fusions are passed to two programs in order to map previously identified conserved functional domains on to the alignment diagrams. The program HMMER (http://hmmer.org) is used to search sequence databases of homologous protein sequences using profile hidden Markov models (profile HMMs) (Durbin, et al., 1998; Krogh, et al., 1994). HMMs are statistical models built from dynamic Bayesian networks and are used to recognise patterns based on training datasets. The PFAM database is a large collection of protein families and is available as a HMM (training dataset) and so it can be used with HMMER to predict conserved functional domains

within the putative list of gene fusions. The data output from HMMER is displayed as an overlay on the alignment diagrams (I will refer to these in short hand as 'domain overlays').

The conserved domain database at NCBI (CDD), a collection of multiple sequence alignments of functional domains, can also be queried to produce domain overlays for inclusion in the images produced by fdfBLAST. HMMs are not available in this case, instead the database is represented as a position-specific score matrix (PSSM) and the program RPS-BLAST (reverse-position-specific BLAST) is used to query the CDD and predict functional domains (Marchler-Bauer, et al., 2005; Marchler-Bauer, et al., 2009; Marchler-Bauer & Bryant, 2004). Both programs can be executed with custom Perl scripts which parse the output into a format ready for fdfBLAST.

The logic behind employing these two databases is to remove putative gene fusions that do not contain any PFAM or CDD identified discrete domains in either of the current version of each database. This allows control of false positive predictions by only allowing functionally discrete domains that are well characterised to be present in the final predictions. Whilst this may potentially remove a number of true positives from fdfBLAST's results we feel that the compromise is effective in order to remove noise created by more frequently occurring false positive hits (i.e. differential matches for low complexity regions).

3.4 fdfBLAST Program Overview

Here follows a brief overview of the operation of the fdfBLAST program using the same two plant genomes as in the Nakamura et al., 2006 analysis as an example. A complete program schematic ends the section with Figure 3:8. The exact process for the two-way Viridiplantae genome analysis can be found in Chapter 4 and an extended 4-way Viridiplantae genome analysis can be found in Chapter 6.

Step 1 – The genomes of *Arabidopsis thaliana* and *Oryza sativa* are subjected to serial genome-to-genome BLASTp comparisons. The results are recorded and passed to Step 2.

Step 2 – Comparative hit lists are generated for all genes in each genome. We shall focus on the known gene fusion with accession AT1G32120.1. The following represents the data stored in the CSV file, the first line indicates the query accession and that it has 12 hits in the query genome. The following twelve lines indicate those hits and their associated data, e.g. amino acid length and e-value.

```
12,AT1G32120.1,1206,
LOC_Os11g40570.3,309,1e-83,55,943,1203,
LOC_Os11g40570.2,309,1e-83,55,943,1203,
LOC_Os11g40570.1,312,2e-82,55,943,1203,
LOC_Os03g36760.2,1030,2e-57,34,66,486,
LOC_Os03g36760.1,1030,2e-57,34,66,486,
LOC_Os03g36830.1,458,2e-49,36,72,407,
LOC_Os12g15450.1,1072,3e-48,34,71,461,
```

LOC_Os09g27250.1,315,2e-33,33,948,1201,

LOC_Os08g36030.1,315,5e-31,30,943,1196,

LOC_Os09g27260.1,318,7e-29,30,943,1180,

LOC_Os08g36040.1,308,6e-22,28,943,1206,

LOC_Os10g11820.1,862,2e-17,27,79,415,

Step 3 – Reciprocal hits are scanned which significantly reduces the data, dismissing ten of the query hits. Just two query hits are left and are stored in a semi-colon separated values file – essentially the same as a comma-separated values file. The data below represents the stored format.

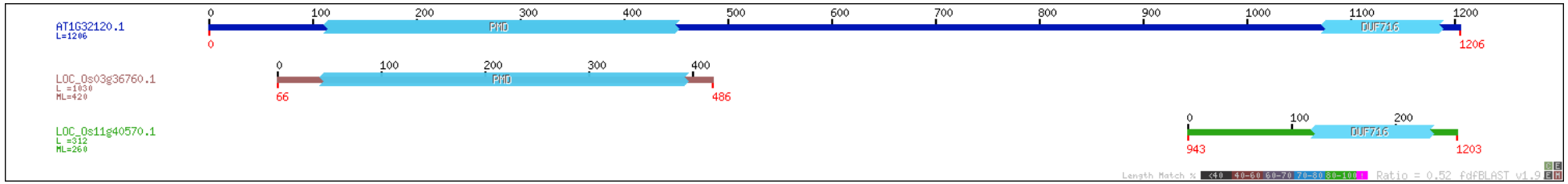
0;AT1G32120.1;1206;LOC_Os03g36760.1;1030;2e-57;34;66;486;

1;AT1G32120.1;1206;LOC_Os11g40570.1;312;2e-82;55;943;1203;

Step 4 – A rank and sort algorithm is applied, as we have only two candidate query hits the sorting is effectively already completed. The ranking is then applied to both query sequences, resulting in a mid-range ratio of 0.52.

Step 5 – The CDD and PFAM databases are queried and the domains are plotted on to the sequences and an image, Figure 3:7, is generated.

Figure 3:7 - An example of fdfBLAST output between the two taxa *Arabidopsis thaliana* and *Oryza sativa*. It indicates a potential fusion event between the two domains PMD and DUF716, fused in *A. thaliana* and split in *O. sativa*. Only PFAM data is shown.



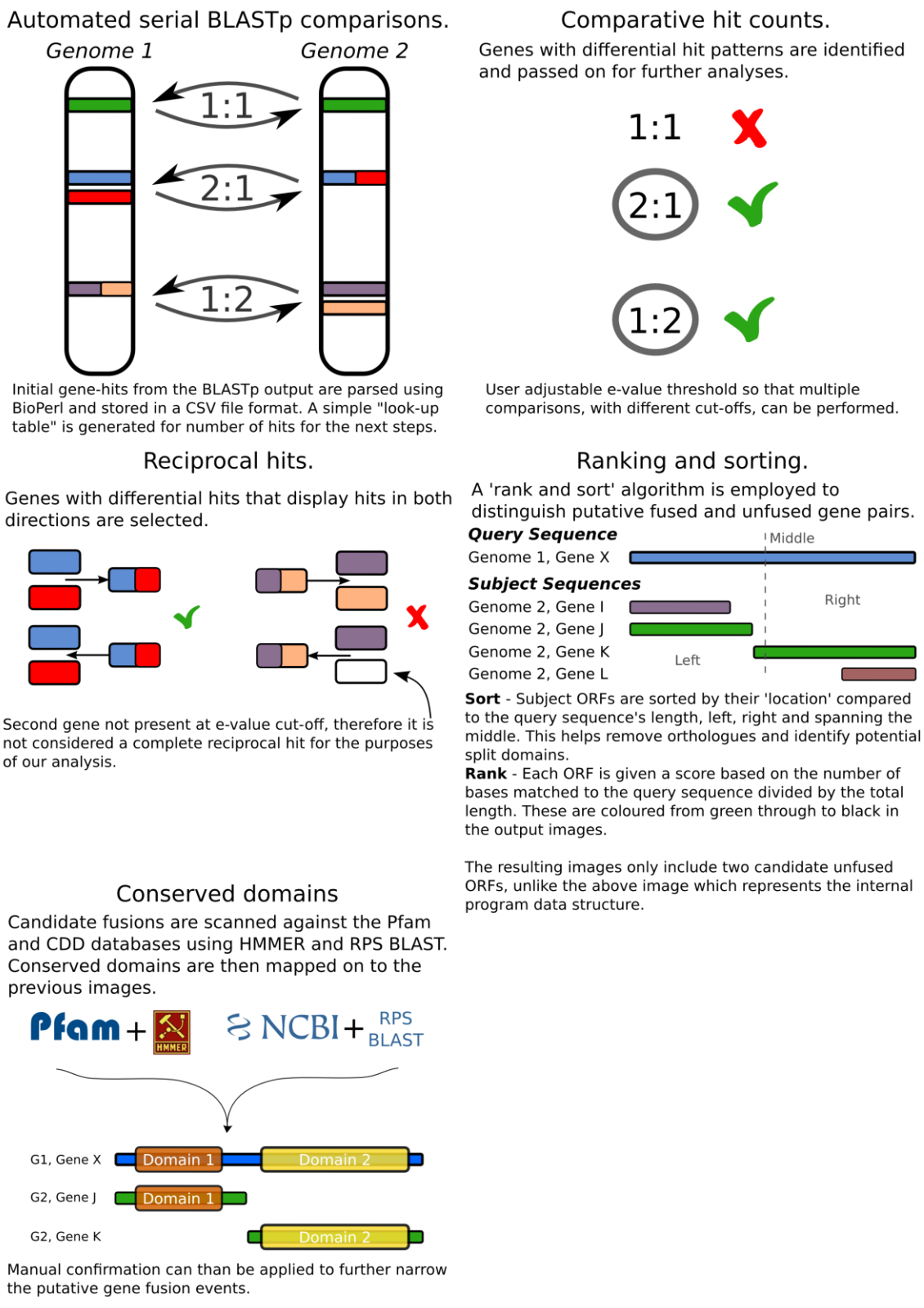


Figure 3:8 - The Complete fdfBLAST Program Schematic

3.5 Discussion

The program fdfBLAST contains a series of methods that aid in the discovery of differentially distributed putative gene fusion and fission events that are distributed between the predicted proteomes of a set of taxa. The process was designed to be automated and repeatable, allowing the user to select many different criteria (e.g. e-value thresholds and similarity thresholds) for use in numerous investigations of a pair or multiple groups of different taxa. It is also extendable and modular, each step can be controlled and run separately from the others as the user desires. For example, the initial BLASTp analysis need only be run once for a group of data and the domain overlay step can utilise either PFAM or CDD domain data independently and has the potential for other database outputs if they are parsed into the correct format.

We feel that the proposed methods described here for the automatic retrieval of differentially distributed gene fusion and/or fission events from whole genome content is a viable and preferable option to the previous attempts where detection was limited to bespoke searches (Durrens, et al., 2008; Nakamura, et al., 2006) or serendipitous discovery (Morris, et al., 2009; Stechmann & Cavalier-Smith, 2002). fdfBLAST allows for the creation of a standard set of results in order to make multiple comparisons between different groups possible. This is achievable with the use of whole predicted proteomes, the inclusion of genomes that are phylogenetically different, dissimilar fusion geometries (how the ORFs/domains are fused or have been reverted) and different similarities between the differentially and reciprocally matched ORFs.

However, we do recognise that, in its current iteration, the program fdfBLAST is subject to several short-comings. Firstly, fdfBLAST is unable to successfully predict gene fusion events of three or more domains due to the identification technique used in the rank and sort method (although it will not miss bi-fusion components of tri-fusions etc). This step of the program will require a new approach to include the detection of multiple domain (equal to or more than three) gene fusion events; however, currently it conveys one advantage which stands to help in the discarding of multiple repeats of small domains which are identified as discrete ORFs.

Secondly, putative gene fusion events that posses ORFs without predicted discrete functional domains in the CDD or PFAM databases are discarded. This is a potential source of artefact, because the PFAM and CDD databases are most certainly biased towards protein motifs that have been well researched. As such the protein catalogue is partial, therefore the results of our fdfBLAST searches are subject to the same bias. Nonetheless we feel it is important at this stage to include this validation step. The inclusion of other available domain databases may confer an advantage in the detection of otherwise false-negative hits (putative gene fusion events discarded due to the lack of an identified discrete conserved domain).

The next chapter will discuss my attempts to use the methods outlined above to re-evaluate the findings of the Nakamura et al., 2006 dataset in order to test the efficacy of fdfBLAST's ability to predict gene fusion and fission events and the conclusions of that paper. Further to the testing stage a set of datasets will also be analysed, in Chapter 6, to test fdfBLAST's ability to detect fusions in larger datasets from across the tree of life.

4 Field-Testing the fdfBLAST Program with the Nakamura et al. (2006) Dataset

4.1 Introduction

Nakamura et al. (Nakamura, et al., 2006) identified sixty candidate gene fusions between *Oryza sativa* and *Arabidopsis thaliana* by performing multiple BLASTp comparisons and recording the entire one-to-many orthologous pairs produced (Nakamura, et al., 2006). Of these sixty potential fusions and fissions, 39 were found to be cases where *Arabidopsis* presented composite genes with *Oryza* containing the split genes and 21 cases where *Oryza* contained the composite genes and *Arabidopsis* possessed split genes (Table 4:1 and Table 4:2).

In order to verify these candidate gene fusions, as the two genome projects have undergone further annotations and updates in the past four years, the sequences for each of the composite genes and split genes were recovered from NCBI GENBANK and their relevant genome project websites. Many composite gene lengths in the original paper appear to have been reported incorrectly; this is most likely due to genes undergoing re-annotation and subsequently a locus name change. I therefore checked and corrected the Nakamura dataset by a) fixing and updating the reported accession numbers and b) updating any incorrect sequences due to changes in the prediction of intron/exon sites; these changes are reflected in Table 4:1 and Table 4:2.

Following this, each set of sequences that represented a Nakamura gene fusion were compared with each other by two methods; firstly a BLAST comparison and secondly

the detection of functional conserved domains using the CDD database from NCBI and PFAM. This analysis, under our strict definition of a gene fusion, dramatically reduced the number of potential gene fusion candidates to twelve including two which correspond to fission events. Interestingly, six potential gene fusions that were reported as figures (Fig.1A, 1B & 1C; Fig.2A 2B & 2C – from their paper) in the original paper (Nakamura, et al., 2006) were discounted due to both the misreporting of composite gene lengths and the absence of domains in the CDD. This does not necessarily mean that they are incorrect but for the purposes of having a testable dataset comparable to the equivalent fdfBLAST analysis with a strict definition of gene fusion they needed to be discounted.

4.1.1 A Re-evaluated Nakamura et al. (2006) Dataset

The putative fusions, displayed in the two tables (Table 4:1 & Table 4:2), are numbered sequentially following the order, left-to-right and top-to-bottom, of the images present in the supplementary data of the Nakamura et al. 2006 article. This happens to correspond to the order of data present in both Table 2 and Table 3 from the same article. Any figures that appear in the Nakamura et al. 2006 article retain their original figure number and are inserted in the order that they appear in Table 2 from the same paper.

In order to help identify the re-evaluated fusion and fission events a '*' precedes the fusion number to indicate a validated fusion and a '!' is used to indicate the presence of a validated fission.

In all cases the composite gene locus name appears as they are printed in the Nakamura et al. 2006 paper, except for genes which have undergone re-annotation or a locus name change, for example K23L20.15 is now denoted as AT5G44800. NCBI GenBank style accessions are also given, in parentheses, to make it easier for subsequent gene lookup and retrieval. Similarly, the split genes locus names are given; however, they do not appear as they were printed in the Nakamura et al. 2006 paper. In their place new locus names, where they exist, are included, and correspondingly NCBI GenBank accessions, in parentheses, have also been given.

4.1.2 Arabidopsis thaliana-composite genes and Oryza sativa-split genes representing candidate gene fusions

#	Composite Gene (*Arabidopsis*)	Gene and Domain Information	Split Genes (*Oryza*)	Gene and Domain Information	Validated Fusion
Fig. 1A	At1g04940 (NP_171986)	Composite gene length incorrectly reported. Tic20 domain present.	LOC_Os07g38110.1 (NP_001060030) LOC_Os01g73150.1 (NP_001045479)	Tic20 domain present. zf-HIT domain present.	Further support needed for zf-HIT domain presence in composite gene. No.
1	At1g11760 (NP_172641.2)	Composite gene length incorrectly reported. No conserved domains present.	LOC_Os10g40070.1 (NP_001065304) LOC_Os03g10080.1 (NP_001049268)	No conserved domains detected in either gene.	No.
2	At1g26760 (NP_173998)	Composite gene length incorrectly reported. TPR 1 & 2, SET, TfoX_N	LOC_Os08g33650.1 (NP_001061873) LOC_Os03g07260.1 (NP_001049093)	DUF377, Glyco_hydro, DUF377 domains present. TPR_2 & TPR_2 and a split SET domain are present.	No.
3*	At1g32120 (NP_174491)	PMD and DUF716 domains present.	LOC_Os03g36760.1 (NP_001050500) LOC_Os11g40570.2 (NP_001068297)	PMD domain present. DUF716 domain present.	Yes.
4	At1g49980 (NP_175420)	Composite gene length incorrectly reported. TMS, TMS_HHH and IMS_C domains present.	LOC_Os03g42010.1 (NP_001050660) Os10g0350800 (NP_001064409)	TMS, TMS_HHH and IMS_C domains present. No domain detected.	No.
5	At1g61000 (NP_176296)	Composite gene length incorrectly reported. Myosin_tail_1 and Noelin-1	LOC_Os03g38010.1 (NP_001050540) LOC_Os03g45770.1 (NP_001050822)	Peptidase_S46 and ERM domains present. Split Membralin domains detected.	No.
6	At2g30100 (NP_180571)	PPR, MA3, MA3, ubiquitin, LRR_RI domains present.	LOC_Os05g28500.1 (NP_001055281)	No domain detected.	Further support needed for PPR and MA3 domain

			LOC_Os10g31790.1 NP_001064753)	Ubiquitin and LRR_RI domains are present.	presence in split gene. No.
7*	At3g02650 (NP_186914)	DUF247, PPR x 5, MRP_S27 domains present.	LOC_Os06g08120.1 (NP_001056969) LOC_Os01g67210.1 (NP_001045088)	DUF247 domain present. Multiple PPR domains present.	Potential.
Fig. 4	At3g23510 (NP_188993)	Amino_oxidase, CMAS and Spermine_synth domains present.	LOC_Os07g29200.1 (NP_001059617) LOC_Os12g16650.1 (NP_001066537)	Amino_oxidase domains present. Amino_oxidase, CMAS and Methyltransf domains present.	No.
8	At3g49140 (NP_190483)	Composite gene length incorrectly reported. Multiple PPR domains, DMAP_binding and TFIIA domains present.	LOC_Os03g13830.1 (NP_001049519) LOC_Os11g34130.1 (NP_001068049)	Multiple PPR repeat domains present. NOA36 and RXT2_N domains present.	No.
9	At3g49640 (NP_201523)	Composite gene length incorrectly reported. Dus and His_biosynth domains present.	LOC_Os10g21660.1 (NP_001064433) LOC_Os04g44890.1 (NP_001053395)	Two PA_decarbox domains present. Dus andTMP-TENI domains present.	No.
10	At4g14310 (NP_193167)	Composite gene length incorrectly reported. No conserved domains detected.	LOC_Os02g56510.1 (NP_001048466) LOC_Os08g45210.1 (NP_001062544)	Two Allexi_40kDa domains present. Mpv17_PMP22 domain present.	No.
11*	At4g19900 (NP_193724)	Gly_transf_sug, Gb3_synth and multiple PPR domains present.	LOC_Os07g37990.1 (NP_001060020) LOC_Os11g39360.1 (NP_001068255)	Mucin, Gly_transf_sug and Gb3_synth domains present. Multiple PPR domains and MRP-S27 domain present.	Potential.
12	At4g22760 (NP_194007)	Multiple PPR repeat domains present.	LOC_Os02g25080.1 (NP_001046756) LOC_Os08g06490.1 (NP_001061056)	No domain detected. PPR repear domains, TPR_4, Glycos_transf_1 and Armet domains present.	No.
13	At4g26450 (NP_194375)	Composite gene length incorrectly reported. No conserved domains detected.	LOC_Os02g34500.1 (NP_001047091) LOC_Os08g38890.1 (NP_001062141)	No domain detected. No domain detected.	No.
14	At4g37920 (NP_195505)	Composite gene length incorrectly reported. Lectin_N domain present.	LOC_Os04g45600.1 (NP_001047091) LOC_Os01g20110.1 (NP_001062141)	No domain detected. No domain detected.	No.
15*	At5g01310 (NP_195751)	HLH, AAA, Macro, Macro, DcpS_C and HIT domains present.	LOC_Os01g61480.1 (NP_001044699) LOC_Os03g18210.1 (NP_001049810)	HLH domain present. DcpS_C and HIT domains present.	Potential. However, there are missing middle domains.
16	AT5G43820 (NP_199195)	Composite gene length incorrectly reported. Multiple PPR repeat domains.	LOC_Os02g20160.1 (NP_001046632) LOC_Os10g41730.1 (NP_001065428)	Multiple ECSIT and PPR repeat domains present. Retrotrans_gag	No.
17	AT5G51540 (NP_199967)	Composite gene length incorrectly reported. Peptidase_M3 domain	LOC_Os06g47210.1 (NP_001058401) LOC_Os02g03250.1	Peptidase_M3 domain present. DUF1183 domain present.	No.

		present.	(NP_001045742)		
18	AT5G55390 (NP_200350)	Composite gene length incorrectly reported. DNMT1-RFD, PHD x3 domains present.	LOC_Os08g24946.1 (NP_001061580) LOC_Os08g39250.1 (NP_001062167)	DNMT1-RFD, PHD x3 domains present. Borrelia_P83 domain present.	No.
19	AT5G58000 (NP_200608)	Composite gene length incorrectly reported. CDC45, Reticulon, Nucleoplasmin and RXT2_N domains present.	LOC_Os08g17870.1 (NP_001061443) LOC_Os05g32430.1 (NP_001055440)	Reticulon domain present. NIF and BRCT domaisn present.	No.
Fig. 1B	At1g33330 (NP_174601)	Composite gene length incorrectly reported. RF-1 and HTH_8 domains present.	LOC_Os01g66379.1 (NP_001045035) Os1g0887200 (NP_001045034)	RF-1, Scaffolding_pro, eIF3g, Cenp-B_dimeris and Cenp-B_dimeris domains present. Sigma70_r4 and HTH_1 domains present.	No.
20	At1g79280 (NP_178048)	Composite gene length incorrectly reported. PR_MLP1_2 and multiple TPR_MLP1_2 domains present.	LOC_Os02g50799.1 (NP_001048082) LOC_Os02g50790.1 (NP_001048081)	2x Filament, IL6, 5_3_exonuc and DUF641 domains present. No domain detected.	No.
21	At2g17930 (NP_179383)	Composite gene length incorrectly reported. Nop14, FAT, PI3_PI4_kinase and FATC domains present.	LOC_Os07g45074.1 (NP_001060453) LOC_Os07g45064.1 (NP_001060452)	No domain detected. FAT, Ndr x2, PI3_PI4_kinase and FATC domains present.	No.
22	At2g19910 (NP_179581)	RdRP and ATP-synt_E_2 domains present.	Os01g0198100 (NP_001042305) LOC_Os01g10140.2 (NP_001042304)	KfrA_N domain present. RdRP and ARL2_Bind_BART domain present.	No.
23	At2g26340 (NP_565620)	Composite gene length incorrectly reported. LEDGF domain present.	Os03g0176700 (NP_001049140) LOC_Os03g07960.1 (NP_001049139)	No domain detected. No domain detected.	No.
24*	At2g46560 (NP_182179)	Rav1p_C, WD40 x2, Rav1p_C, MurB_C and 5x WD40 domains present.	Os01g0552000 (NP_001043310) LOC_Os01g37120.1 (NP_001043309)	Rav1p_C domain present. 5x WD40 domains present.	Potential. Missing middle domains.
25	At3g42670 (NP_189853)	SNF2_N and Helicase_C domains present.	LOC_Os07g49210.1 (NP_001060722) LOC_Os07g49210.1 (NP_001060723)	No domain detected. SNF2_N and Helicase_C domains present.	No.
26!	At3g49410 (NP_190510)	Composite gene length incorrectly reported. Tau95 domain present.	LOC_Os01g34420.1 (NP_001043231) Os01g0528000 (NP_001043232)	Tau95 domain present. Tau95, Rtt106 and CobT domains present.	Fission.
27	At3g49600 (NP_566922)	Composite gene length incorrectly reported. UCH, DUSP x3 and ubiquitin domains present.	LOC_Os03g09260.1 (NP_001049243) LOC_Os03g09270.1 (NP_001049244)	NOB1_Zn_bind, UCH and zf-FPG_IleRS domains present. Ubiquitin domain present.	No.
28!	At3g56330 (NP_191192)	TRM domain present.	LOC_Os05g25880.1 (NP_001055198)	TRM domain present.	Fission.

			LOC_Os05g25870.1 (NP_001055197)	TRM and DUF2067 domain present.	
29	At4g02940 (NP_192203)	Composite gene length incorrectly reported. 2OG-FeII_Oxy domain present.	Os05g0401700 (NP_001055491) Os05g0401500 (NP_001055490)	Mis14 domain present. 2OG-FeII_Oxy domain present.	No.
30	At4g34100 (NP_195136)	Zf-C3HC4 domain present.	Os06g0639100 (NP_001058158) LOC_Os06g43210.1 (NP_001058157)	Zf-C3HC4 domain present. No domain detected.	No.
31	T2L20.8 (Missing?)	Composite gene does not appear to exist!	LOC_Os05g31056.1 (NP_001055373) LOC_Os05g31062.1 (NP_001055374)	TPR_2 and TPR_1 repeated domains present. TPR_1, Adenine_glyco~, yntaxin-6_N, DnaJ and EntA_Immun domains present.	Unknown.
32*	AT5G44800 (NP_199293)	Composite gene length incorrectly reported. PHD, Chromo x2, SNF2_N and Helicase_C domains present.	LOC_Os07g31450.1 (NP_001059706) LOC_Os07g31450.1 (NP_001059705)	PHD and FLO_LFY domains present. SNF2_N and Helicase_C domains present.	Potential.
Fig. 2A	At4g18260 (NP_193560)	Tetraspannin and TSC22 domains present.	LOC_Os01g47630.2 (NP_001043803) LOC_Os01g47620.1 (NP_001043801)	Chs3p and MerC domains present. DUF501 and DUF2992 domains present.	No.
Fig. 2B	At3g49730 (NP_190542)	Composite gene length incorrectly reported. DUF384, PPR, MRP-S27, Coatomer_E, ECSIT and PPR domains present.	LOC_Os03g51840.1 (NP_001051146) LOC_Os03g51790.1 (NP_001051144)	SelB-wing_2 and 9x PPR repeat domains present. GTP_EFTU and FeoB_N domains present.	No.
Fig. 2C	At1g12930 (NP_563920)	Xpo1 domain present.	Os11g0543700 (NP_001068046) LOC_Os11g34120.1 (NP_001068047)	Xpo1 domain present. Alginate_lyase domain present.	No.
Fig. 1C	At1g27750 (NP_174096)	Composite gene length incorrectly reported. FCD, FCD and SPOC domains present.	LOC_Os01g56000.1 (NP_001044348) LOC_Os01g56000.1 (NP_001044349) LOC_Os03g10750.1 (NP_001049312)	DUF729 domain present. RRM_1, RNA_pol_N, RRM_1 and SPOC domains present. CUE domain present.	No.

Table 4:1 - Re-evaluated results of Nakamura et al., 2006. This table represents Table 2 from the same paper, but has updated accessions and indicates if the gene fusion or fission event has been mis-predicted due to genome annotation errors. Of the 39 candidate gene fusion and fission events only nine survived the re-evaluation where the other 30 were rejection due to missing domains or where re-annotation of ORFs had occurred.

4.1.3 Oryza sativa-composite genes and Arabidopsis thaliana-split genes representing candidate gene fusions

#	Composite Gene (*Arabidopsis*)	Gene and Domain Information	Split Genes (*Oryza*)	Gene and Domain Information	Validated Fusion
1*	LOC_Os01g66140.1 (NP_001045018)	SWIB, Plus-3, GYF domains present.	At5g63700 (NP_201175) At2g16470 (Q9SIV5)	SWIB and Plus-3 domains present. GYF~zf-CCCH domains present.	Potential.
2	LOC_Os03g31839.1 (NP_001050432)	No domain detected.	At5g10490 (NP_001078567) At5g41710 (Pseudo gene)	Split gene length incorrectly reported. MS_channel domain present.	No.
3	Os05g0497600	No gene model present in *Oryza sativa* genome.	At5g44280 (NP_199241) At5g53920 (NP_200203)	Split gene length incorrectly reported. Syndecan, zf-Nse and Syndecan domains present. PrmA and Methyltransf_11 domains present.	No.
4	LOC_Os06g12360.1 (NP_001057211)	CARD, CARD, PPR, Clathrin and 9x PPR domains present.	At1g62260 (NP_176416) At1g18790 (NP_173314)	15x PPR repear domains present. Rabaptin and Baculo_LEF-2 domains predicted.	No.
5	LOC_Os06g13030.1 (NP_001057248)	Composite gene length incorrectly reported. LIM, LIM and Rieske domains present.	At1g10200 (NP_172491) At3g05900 (NP_187241)	LIM, PHF5 and Metallothio_Pro domains present. Peptidase_M43 and Peptidase_M43 domains present.	No.
6	LOC_Os10g28640.1 (NP_001064629)	Composite gene length incorrectly reported. APH, PPR x3, ECSIT, PPR x3, RIO1 and Choline_kinase domains present.	AT5G61370.1 (NP_200945) AT5G26110.1 (NP_568482)	Ribosomal_S21e, PPR, RPM2 and 3x PPR repear domains present. Pkinase and DUF227 domains present.	No.
7*	LOC_Os11g33110.1 (NP_001068028)	SNARE_assoc, Cupin_1 and Cupin_2 domains present.	At1g44960 (NP_175116) AT5G61750.1 (NP_200983)	SNARE_assoc domain present. Cupin_1 and Cupin_3 domains present.	Potential.
8	LOC_Os01g29210.1 (NP_001043105)	Composite gene length incorrectly reported. HDAC_interact, DUF3595 and DUF3595 domains present.	At2g48060 (NP_182327) At2g48050 (BAD69280) At2g48040 (AAD13708)	DUF3595 domain present. HDAC_interact, DUF3595 and DUF3595 domains present. DUF3595 domain present.	No.
9	LOC_Os02g17970.2 (NP_001046559)	Composite gene length incorrectly reported. PP2C, cNMP_binding, cNMP_binding, Pkinase and Pkinase_Tyr domains present.	At2g20050 (NP_179595) At2g20040 (AAY57316)	PP2C, cNMP_binding, cNMP_binding, Pkinase and Pkinase_Tyr domains present. Pkinase and Pkinase_Tyr domains present.	No.
10	LOC_Os02g47900.1 (NP_001047885)	Composite gene length incorrectly reported.	At2g23740 (NP_179954)	zf-TRM13_CCCH, zf-TRM13_CCCH, Pre-SET and SET domains present.	No.

			At2g23750 (AAC17088)	Pre-SET and SET domains present.	
11	LOC_Os02g48000.1 (NP_001047892)	Composite gene length incorrectly reported. DUF3548, DUF3548 and TBC domains present.	AT5G52580.1 (NP_200071) AT5G52590.1 (ABH11525)	DUF3548, DUF3548 and TBC domains present. TBC domain present.	No.
12	LOC_Os03g06340.1 (NP_001049028)	Composite gene length incorrectly reported. XS domain present.	At3g22430 (NP_566707) At3g22435 (NP_566708)	DUF605 domains present. XS domain present.	No.
13	LOC_Os03g14020.1 (NP_001049530)	Composite gene length incorrectly reported. DUF1645, Cons_hypoth95 and Methyltransf_16 domains present.	At4g35987 (NP_680769) At4g35990 (ABK32118)	DUF1645, Methyltransf_16 and Methyltransf_12 domains present.	No.
14	LOC_Os04g36580.1 (NP_001052885)	Composite gene length incorrectly reported. Zf-CCHC, rve and RVT_2 domain present.	T15F17.6 () T15F17.4 ()	Unknown split genes!	No.
15	LOC_Os07g49260.1 (NP_001060729)	Composite gene length incorrectly reported. Xpo1 and HEAT domains present.	At3g08960 (NP_187508) At3g08955 (AAU45222)	IBN_N and Xpo1 domains present. No domain detected.	No.
16	LOC_Os08g01130.1 (NP_001060762)	Composite gene length incorrectly reported.	At3g48900 (NP_001118795) At3g48910 (Not Present)	XPG_N, XPG_I and Chromo domains present.	No.
17	LOC_Os08g14770.1 (NP_001061354)	Composite gene length incorrectly reported. Aminotran_3 domain reported.	AT5G57600.1 (ABU50829) AT5G57590.1 (NP_200567)	Aminotran_3 domain reported. Aminotran_3 domain reported.	No. First split gene replaced by second split gene.
18	LOC_Os09g39270.1 (NP_001063953)	Composite gene length incorrectly reported. DUF618 and f-C2H2 domains present.	At2g36485 (NP_565849) At2g36480 (NP_565848)	No domain detected. DUF618 domain present.	No.
19	LOC_Os10g10170.2 (NP_001064251)	Composite gene length incorrectly reported. PPR x8 repeat domains present.	At4g34830 (NP_195209) At4g34820 (AAS76768)	PPR x9 repeat domains present. No domain detected.	No. Second split gene replaced by first split gene.
20	LOC_Os12g10700.1 (NP_001066395)	Composite gene length incorrectly reported. Baculo_IE-1 and PHD domains present.	At4g10940 (NP_567371) At4g10930 (NP_567370)	Baculo_IE-1 and PHD domains present. No domain detected.	No.
21*	LOC_Os11g47944.1 (NP_001068554)	DUF729, Cyclin_N, Cyclin_C and Thaumatin domains present.	AT5G02110.1 (NP_195831) AT5G02140.1 (NP_195834)	Cyclin_N and Cyclin_C domains present. Thaumatin domain present.	Potential.

Table 4:2 - Re-evaluated results of Nakamura et al., 2006. This table represents Table 3 from the same paper, but has updated accessions and indicates if the gene fusion or fission event has been mis-predicted due to genome annotation errors. Of the 21 candidate gene fusion and fission events only

three survived the re-evaluation where the other 18 were rejection due to missing domains or where re-annotation of ORFs had occurred.

4.2 Results

4.2.1 fdfBLAST and a Two Plant Genome Dataset

The genomes of *Arabidopsis thaliana* and *Oryza sativa* were run through fdfBLAST in order to predict potential candidate gene fusions at several e-value cut-offs (between 1e-70, 1e-40, 1e-30, 1e-20 and 1e-10 and 0). At the most strict e-value setting (1e-70) 87 potential gene fusions were predicted – twenty more than the total Nakamura et al. (2006) dataset – and at the lowest e-value setting (1e-10) 266 candidate gene fusions were returned. fdfBLAST, with the help of the CDD and PFAM databases, annotated each alignment diagram with conserved domain topologies. As each database contains different sets of predicted domains there was a degree of difference observed between the final numbers of putative gene fusions (Table 4:3).

	CDD	PFAM	Shared (Overlap)	Total (Unique)
Validated	9	19	7	16
Not a Fusion	225	195	177	243

Table 4:3 - The results from an fdfBLAST search of *Oryza sativa* and *Arabidopsis thaliana* at 1e-10 show a number of validated fusions. The 'shared' column indicates when CDD and PFAM predict the same sets of conserved domains. The 'Not a Fusion' row indicated fdfBLAST putative fusions that are not considered to be positive hits under our strict version of a gene fusion (i.e. fusion of two discrete domains not identified by PFAM or CDD).

4.2.2 fdfBLAST Results Vs the Re-evaluated Nakamura et al. (2006) Dataset

Comparing the candidate gene fusions from fdfBLAST with those of the re-evaluated Nakamura et al. (2006) dataset we can measure the ability of fdfBLAST to predict gene fusions. Table 4:4 indicates the 19 differentially distributed gene fusions that were present and accepted under our strict definition of a gene fusion event from the output of fdFBLAST at an e-value range from 0 to 1e-10.

Fusion #	Domains	*Arabidopsis thaliana*	*Oryza sativa*
1	PMD DUF716	AT1G32120.1	LOC_Os03g36760.1 LOC_Os11g40570.1
2	SYS1 WD40 Coatamer_WD40	AT1G79990.1	LOC_Os01g41040.1 LOC_Os06g05180.2
3	PTEN_C2 FH2	AT2G25050.1	LOC_Os07g40520.1 LOC_Os02g55150.1
4	DUF3535 DEXDc HepA HELICc	AT3G54280.1	LOC_Os02g06588.1 LOC_Os02g06592.1
5	Cu_amine_oxideN2 TynA (Cu_amin_oxid)	AT4G12290.1	LOC_Os04g20164.2 LOC_Os04g20164.1
6	MYSc_type_XVIII COG5022 DIL	AT4G33200.1	LOC_Os10g25565.1 LOC_Os02g53740.2
7	Mut-7_like_exo COG1656	AT5G24340.1	LOC_Os07g26920.1 LOC_Os07g26930.2
8	COG1936 MaoC	AT5G60340.1	LOC_Os07g22950.3 LOC_Os05g09370.3
9	SMC_prok_B C2_1 C2 GRAM	AT1G03370.1	LOC_Os05g04950.1 LOC_Os06g19400.1
10	MIF4G MA3	AT1G62410.1 AT4G30680.1	LOC_Os02g39840.1

11	Porin3_VDAC (Tom40) PRK07912/6772	AT5G15090.1 AT1G74710.1	LOC_Os09G19734.1
12	tif TRX_PICOT	AT1G48500.1 AT3G17880.1	LOC_Os09g23650.1
13	DUF3453 Symplekin_c	AT1G27570.1 AT1G27595.1	LOC_Os01g49940.1
14	PP2C PTZ00224 STKc_PCTAIRE_like	AT3G63340.1 AT3G63330.1	LOC_Os11g37540.1
15	ANF_receptor Lig_chan	AT5G11210.1	LOC_Os06g08880.1 LOC_Os06g09090.1
16	Neurobeachin Beach WD40	AT2G45540.1	LOC_Os04g46894.1 LOC_Os04g46892.1
17	DUSP UBP12 Peptidase_C19	AT2G40930.1	LOC_Os11g28360.1 LOC_Os11g28365.1
18	Tau95	AT3G49410.1	LOC_Os01g34420.1 LOC_Os01g52800.1
19	TRM	AT3G56330.1	LOC_Os05g25880.1 LOC_Os05g25870.1

Table 4:4 - The 19 putative differentially distributed gene fusions accepted under our strict definition of a gene fusion event predicted by fdfBLAST between the genomes of *Arabidopsis thaliana* and *Oryza sativa*. 18 and 19 are candidate fission events, hence having only one domain present.

In comparison to the Nakamura et al. (2006) dataset, fdfBLAST predicted a significantly larger number of potential fusions; however, this number is dramatically reduced (16 of 243 = 7%) when data from either CDD or PFAM is used to predict conserved domain structures in the candidate gene fusion (for the purposes of this analysis we preferred the data from PFAM as it yielded more results overall than CDD in our comparisons). Observing Table 4:5 we can see that in most cases fdfBLAST predicts a higher number of potential fusions than was reported in the revised Nakamura et al. dataset. Please

also note that the Nakamura dataset was not run at different e-value thresholds because the analysis was manually completed. Of the twelve verified Nakamura et al. (2006) putative gene fusion events only three (25%) were predicted by the fdfBLAST program at the 1e-10 e-value setting, although in each instance fdfBLAST predicted nearly double the amount of putative gene fusion events, suggesting its efficacy as a fusion finding tool.

	1e-10	1e-20	1e-30	1e-40	1e-70
Revised Nakamura et al. (2006) Dataset	12	12	12	12	12
Total fdfBLAST Putative Fusion Events	266	174	145	126	87
fdfBLAST Validated by PFAM	19	18	16	13	8
Shared	3	3	2	1	0

Table 4:5 - The total figures for fdfBLAST's prediction for fusion events between *Arabidopsis thaliana* and *Oryza sativa*. Validated fdfBLAST fusion predictions are presented along with the overlap between the verified Nakamura dataset.

fdfBLAST is further confirmed as a better approach when you compare the number of predictions (Nakamura's orthologous gene pairs) with the confirmed events. The original Nakamura et al. dataset thus has a roughly 0.5% success rate which is further reduced to 0.11% when we use the validated dataset, fdfBLAST records a much better value at 7%.

To help visualise the difference between predictions and validations Figure 4:1 illustrates the number of fusions events as circles (in the style of a Euler diagram) with a diameter based on the number of predictions. As we can see the Nakamura et al. dataset, based on an extrapolation by the authors (Nakamura, et al., 2006), to the

whole genome predicts a vast number of orthologous pairs (fusion events) in blue and is so large that is runs outside the image. These numbers are dramatically reduced when they were both confirmed in the original paper (orange) and subsequently reconfirmed in this thesis (green). Contrastingly, fdfBLAST predicts far fewer differentially distributed fusion events (red) yet manages to retain a higher number of validated fusions after comparison with PFAM. The tiny circle in the middle represents the total overlap (3 differentially distributed fusions) of validated fusions between the two processed datasets.

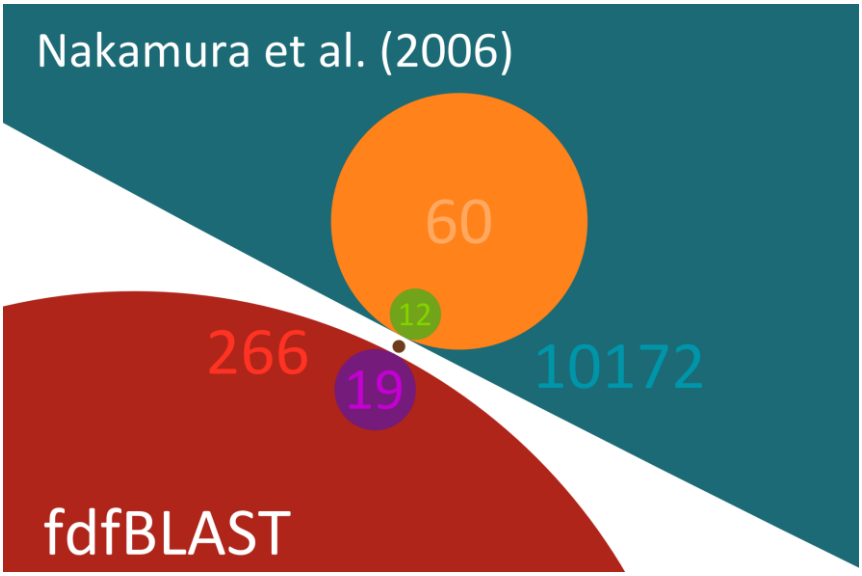


Figure 4:1 - A representation of putative orthologous gene pairs (blue, 10,172), confirmed pairs (orange, 60) and subsequently , in this study re-evaluated, pairs (green, 12) from the Nakamura et al. (2006) dataset compared to the differentially distributed putative fusions predicted by fdfBLAST (red) and the fdfBLAST putative fusions validated by PFAM (purple). The overlap represented by the brown circle is the three shared fusions between the datasets. Circle diameters are based on the number of predictions and are represented as pixels in the construction of the figure.

Further to Figure 4:1 it is interesting to look at the overlap directly for the 1e-10 e-value range output from fdFBLAST, this is represented as a Venn diagram in Figure 4:2

where we can quite clearly see fdfBLAST has predicted more validated fusion events than that of the previous study.

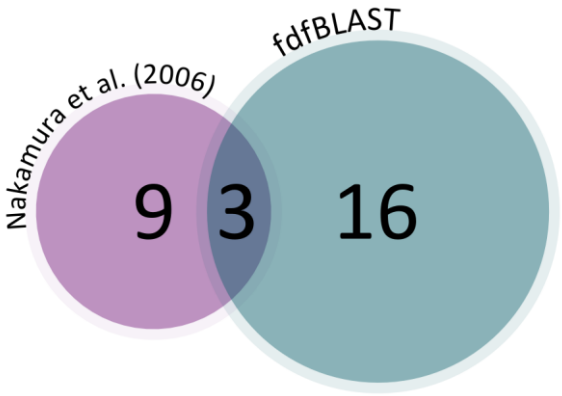


Figure 4:2 - A Venn diagram demonstrating the different number of gene fusion events validated under a strict definition of a gene fusion for both datasets. Three gene fusion events were predicted and are shared by both methods. fdfBLAST is noticeably better than the manual approach.

4.3 Discussion

When compared the two approaches produced rather different sets and numbers of gene fusion and fission events. Nevertheless, there was a clear overlap of the data (as we can see in Figure 4:2 and Table 4:5) which is encouraging for both fdfBLAST and also for the Nakamura dataset, in that it would be unlikely for there to be an exact match of all fusion events due to the different nature of each search technique. Moreover, it would be equally unlikely for there to be no overlap between the two datasets, especially when a very stringent definition of a gene fusion was used for the comparison here. It is clear, however, that fdfBLAST's automatic approach yields far more positive hits than the manual approach achieved, conversely it appears to predict far more candidate false positives (this result is of course amplified by lack of representation of unknown protein domains in PFAM/CDD) but these are much easier

to remove as a final manual process post the automatic prediction. We believe that this is an acceptable time cost in comparison to the process of manually assessing each gene within each genome, especially so if the study was escalated to include more genomes (the more genomes, the more genes, and so the more comparisons to complete).

The principle finding of the Nakamura et al. (2006) that we are interested in, was that by the process of out-group comparison (not completed for out dataset in this instance, however chapter 6 does attempt this) on their 60 candidate gene fusions they established that gene fission in *O. sativa* occurred at a higher rate than gene fusion did (6 fissions and 3 fusions) and that gene fusion and fission events were almost equally common in *A. thaliana* (2 fissions and 3 fusions). This data therefore represented an overall higher rate of fission events than expected when compared to the previous fusion and fission rates. Interestingly, once I had completed the re-evaluation of the Nakamura et al. (2006) dataset this distribution was no longer present, with a total of 7 fusions and 2 fissions present in *Arabidopsis thaliana* and only 3 fusions present in *Oryza sativa*.

In contrast to the initial Nakamura at al. findings and with similarity to the re-evaluated dataset the predictions from fdfBLAST also identify a higher rate of fusion events compared to that of fission events, as we would expect under the most parsimonious explanation (discussed in Section 1.4.1). Twelve fusion events were identified in *Arabidopsis thaliana* and 5 fusion events and 2 fission events were identified in *Oryza sativa* which equates to over 8 times more fusion events within both genomes, please see Figure 4:3.

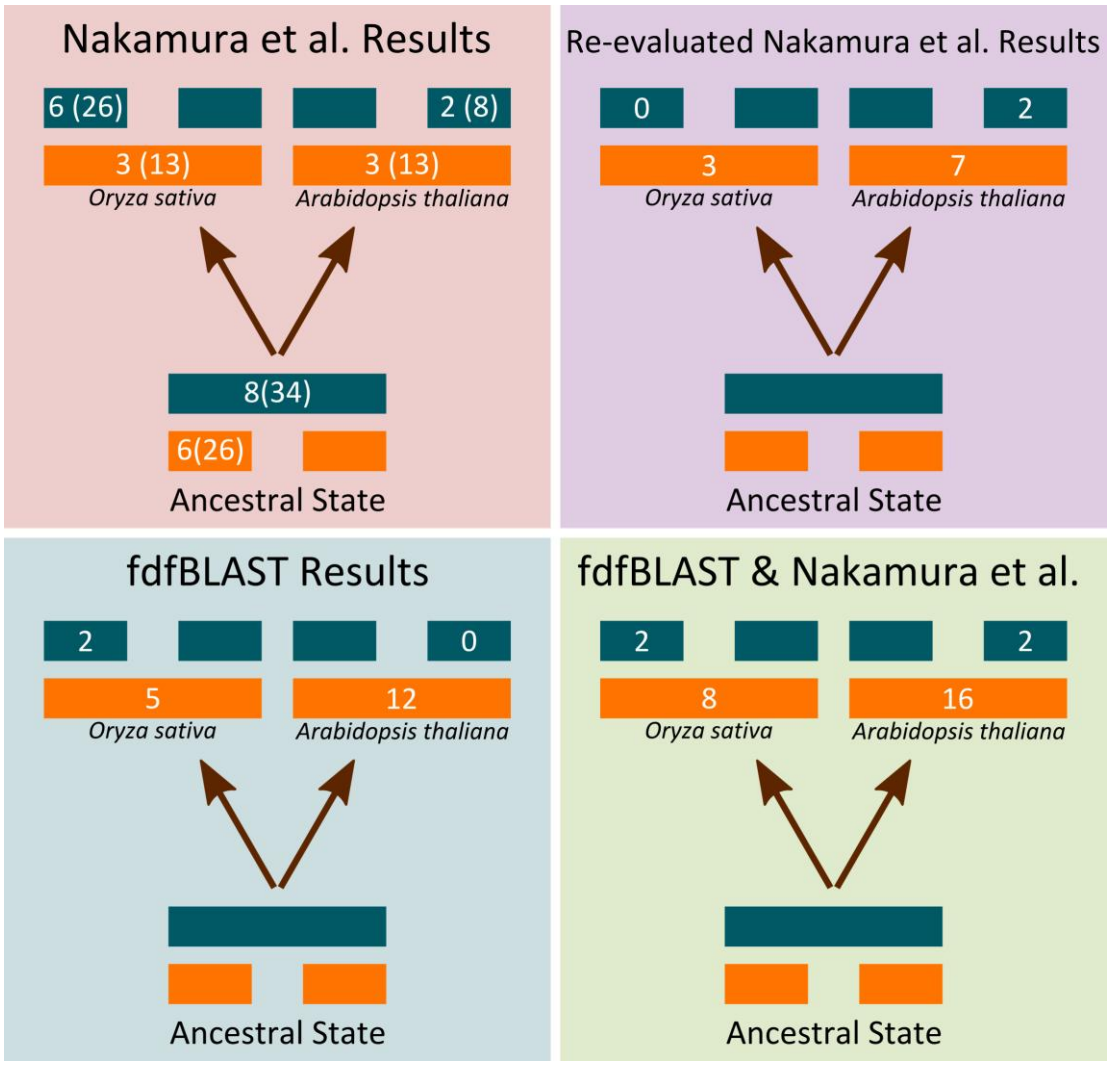


Figure 4:3 - These four boxes indicate a summary of polarised gene fusion and fission events for the two different approaches outlined previously. Top left (red), depicts the original Nakamura et al. (2006) results - the numbers indicate the quantity of observations for gene fusion and fission events and numbers in parentheses indicate the extrapolated figure. Top right (purple) depicts the re-evaluated Nakamura et al. (2006) data. Bottom left portrays the results for fdfBLAST and bottom right (green) indicates the cross-over between the re-evaluated and fdfBLAST analyses. Please note that ancestral states are not extrapolated for the re-evaluated dataset or for the fdfBLAST results as phylogenies were not computed for those datasets.

The rates of fusion versus fission are as we would expect under the most parsimonious explanation, that is to say there are fewer fission events as it is evolutionary costly to presuppose that the inclusion of a stop codon followed by a promoter region and a

start codon between the two coding sections whilst remaining 'in frame' will produce gene fission events more frequently than gene fusion events (discussed more in Section 1.4.1). However, these analyses only look at the comparison between two plant genomes, which is not particularly useful for the polarisation of synapomorphies or for commenting on the rates of fusion and fissions events across different kingdoms. Understandably, the manual search only looked at two genomes, especially as they identified over 10,000 extrapolated gene fusions, which is a lot of data to manually validate. Therefore, it is most advantageous for an automatic system that can perform both tasks, especially for the inclusion of more genomes.

The analyses in this chapter only attempt a two-way comparison (between two genomes) and no phylogenies were calculated for the putative gene fusion and fission events predicted by fdfBLAST. Therefore, considering both of these factors, there may have been a direct impact on the number of polarised reversion events that we were able to observe. Consequently, construction of automatically produced phylogenetic topologies, using 'Darren's Orchard' (T. A. Richards, et al., 2009), of the unfused domains was added to the aims of any additional analyses. Indeed, Chapter 6 looks at what happens when we include more genomes (a total of four in a dataset), create phylogenies based on the split ORFs/domains and also attempts to report the relative occurrence and abundance of gene fusion and fission events across the eukaryotic tree of life.

5 Inferring the phylogeny of the kinetoplastids: a comparative genomics approach using whole genome datasets with low taxon sampling

5.1 Introduction

The group of pathogenic flagellate protozoa known as Kinetoplastida, so called due to their possession of a modified mitochondrion (a kinetoplast) with a disc-shaped mass of circular DNA, includes the genera *Trypanosoma* and *Leishmania* (both vertebrate parasites), in addition to others such as *Crithidia* and *Leptomonas* (both parasites of arthropods) (Lake, de la Cruz, Ferreira, Morel, & Simpson, 1988; O. W. Olsen, 1986). The genus *Trypanosoma* includes: *Trypanosoma brucei*, an extracellular parasitic protist (Berriman et al., 2005), which causes sleeping sickness in humans and a similar wasting disease known as Nagana in mammals; *Trypanosoma cruzi*, an intracellular parasitic protist that causes Chagas disease in humans and also infects a range of mammals which can act as reservoirs of the human form of the disease; and *Leishmania major*, an intracellular parasitic protist which causes the disease known as Leishmaniasis in humans and animals; Leishmaniasis can take a variety of forms, cutaneous, subcutaneous and visceral, and pathogenicity can vary widely between hosts.

Since the first attempts to classify their evolutionary history there have been inconsistent and conflicting reports about their true phylogenetic relationship, which have varied considerably depending on the gene sequences analysed, the number of taxa included, choice of out-group and phylogenetic methodology employed (Alvarez,

Cortinas, & Musto, 1996; Hamilton, Stevens, Gaunt, Gidley, & Gibson, 2004; A. Hughes & H. Piontkivska, 2003; A. L. Hughes & H. Piontkivska, 2003; Lukeš et al., 1997; Piontkivska & Hughes, 2005; A. G. B. Simpson, Gill, Callahan, Litaker, & Roger, 2004; Stevens & Gibson, 1999; Stevens, Noyes, Schofield, & Gibson, 2001; Wright, Li, Feng, Martin, & Lynn, 1999). The issue that has provided most debate is that concerning the monophyly of the trypanosomes (see A. G. B. Simpson, et al., 2006 for an overview of this topic). As discussed earlier in this thesis (Chapter 1), the problem can be defined within the framework of a trifurcated tree - a topology with three branches - with three hypothetical positions for the root to be placed. This is demonstrated in Figure 5:1, which shows a topology and the locations for the three possible positions for the root; the insert depicts the three resulting cladograms. Topology X describes the monophyly of trypanosomes, whereas topologies Y and Z each display a paraphyletic distribution.

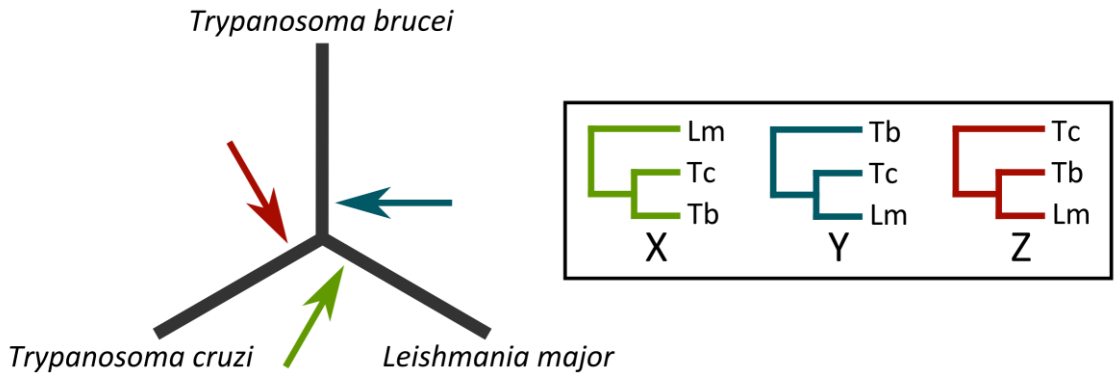


Figure 5:1 - A representation of a trifurcated tree topology for the kinetoplastids. The insert depicts three topologies, X, Y and Z, which show the branching order of three kinetoplastids. Note that topologies Y and Z indicate paraphyly of the trypanosomes, whereas topology X depicts their monophyly.

One early single-gene, with low taxon sampling (Maslov, Lukes, Jirku, & Simpson, 1996) suggests that *Trypanosoma cruzi* branches with *Leishmania major* (topology Y from Figure 5:1). Additionally, many other early single-gene phylogenies also inferred paraphyly of the genus *Trypanosoma*, recovering the same grouping of *T. cruzi* with *L. major*; similarly, evidence from the genome content of *L. major* and *T. cruzi* (Berriman, et al., 2005; El-Sayed, Myler, Bartholomeu, et al., 2005; Ivens et al., 2005) (specifically protein families which are expanded in *T. brucei* compared to *L. major* and *T. cruzi*, and a large set of orthologues shared between *L. major* and *T. cruzi*) supports their grouping to the exclusion of *T. brucei*. Nevertheless, although gene content may support such a hypothesis, a number of other factors do not agree.

More recently, improved techniques and available datasets have allowed for multiple (2-9 genes) DNA and protein alignments to be made (Hamilton, et al., 2004; A. G. Simpson, Lukes, & Roger, 2002; A. G. B. Simpson, et al., 2004). In contrast to results from some previous studies, these analyses have suggested that *T. brucei* and *T. cruzi* do demonstrate monophyly with the exclusion of *L. major* (topology X from Figure 5:1). However, analyses based on single protein datasets, inappropriate taxon sampling and the associated problems of compositional bias, hidden paralogy and lateral gene transfer (LGT), as discussed in Chapter 1 and 2, continue to diminish support for relationships defined in phylogenetic analyses and have lead to uncertainty in the reconstruction of the kinetoplastids' evolutionary history.

Therefore, a new approach, which could attempt to overcome or reduce the effects of these problems, was needed. Using whole genome datasets, we have undertaken an analysis of protein genes in order to resolve the branching relationships of *T. brucei*, *T.*

cruzi and *L. major*. Using whole genome gene-by-gene phylogeny (T. A. Richards, et al., 2009) we first identified 'reliable' single gene families that demonstrated monophyly of the kinetoplastids within a broad eukaryotic phylogeny. We then used this dataset in combination with a precise out-group choice of *Naegleria gruberi* and/or *Euglena gracilis*, coupled with large-scale gene concatenation, signal stripping (Pisani, et al., 2007) and diverse phylogenetic techniques to investigate the relative branching order of *T. brucei*, *T. cruzi* and *L. major*.

5.2 Methods

The main methods employed in this chapter are similar to the processes outlined in Section 2.6.1: Automatic Tree Construction Pipeline. Briefly, the entire predicted proteome of *Trypanosoma brucei* was subjected to a sequential, genome-to-genome, BLASTp analysis against 795 other eukaryotic genomes, Section 8, using the automatic tree building pipeline called 'Darren's Orchard' (T. A. Richards, et al., 2009).

Out-group choice was based on the most closely related sequenced genome available to the kinetoplastids (Hampl, et al., 2009); the genome of *Naegleria gruberi* which is available from the DOE JGI (http://genome.jgi-psf.org). *N. gruberi* was deemed most suitable for this set of analyses as it is contained within the super-phylum of the Discicristata (T. Cavalier-Smith, 1998) which are comprised of a set of unicellular protists and so called as they contain mitochondria which posses disc-shaped cristae. *Naegleria* is found within the class Heterolobosea, the sister-group to Euglenozoa, which contains the Kinetoplastida.

The output of the BLASTp analyses showed the presence of 599 gene-markers putative homologues that were shared between the genomes of *T. bruci*, *T. cruzi*, *L. major* and *Naegleria gruberi.* These 599 maximum likelihood (ML) trees were assessed visually for the presence of kinetoplastid monophyly, which identified 75 reliable genes for analysis, which in turn showed support for three different possible tree topologies. These 75 genes were split into three datasets; dataset X included all genes which recovered the topology of (Lm, (Tb, Tc)), dataset Y included all genes which recovered the topology of (Tb, (Tc, Lm)) and dataset Z included all genes which recovered the topology of (Tc, (Tb, Lm)). These topologies follow the three possible branching orders outlined in the insert of Figure 5:1. Once the datasets were assembled, the protein sequences from each tree were collated and concatenated together. A further dataset was assembled and represented a concatenated dataset alignment, containing all of the genes present in datasets X, Y and Z, which contained 36,278 individual sites.

Each of the four concatenated alignments were then subjected to six different phylogenetic methodologies and alternative topology tests; two fast-ML topology tests (RAxML and PhyML) with both the LG model (at the time of analysis the LG model was new and was not contained within MODELGENERATOR) and the best model optimised by MODELGENERATOR, two approximate likelihood ratio tests (SH and X^2), a log-det approach with LDDist and finally a Bayesian analysis in the program MrBayes. The use of all these programs is described earlier in this thesis; see Chapter 2: Methods.

A secondary analysis was conducted in order to increase the number of taxa present in the analysis. We were able to obtain data from expressed sequence tags (ESTs) from the unicellular protist *Euglena gracilis* (grouped within the Euglenozoa) from the

taxonomically broad EST database (TBestDB) (TBestDB) which was used as a second species for the out-group. *Crithidia deanei* (Kinetoplastida), *Leptomonas seymouri* (Kinetoplastida), *Diplonema papilatum* (Euglenozoa) and *Bodo saltans* (Euglenozoa) were also assessed for their suitability as out-groups. The 58 gene families identified that represent dataset X (and so topology X, Figure 5:1 & Figure 5:2) were subjected to BLAST searches against each of the taxa listed above. However, the majority of the gene families did not return sufficient reliable BLAST hits from the available EST databases or the ESTs were simply unavailable. This reduced the dataset somewhat. Further sequences were recovered from two other kinetoplastids: *Trypanosoma vivax* and *Leishmania braziliensis*, both from GeneDB {Hertz-Fowler, 2004 #32}. The genomes of *Leishmania mexicana* and *Trypanosoma congolense* were not included as they were considered too closely related to *Leishmania major* and *Trypanosoma brucei*.

The four concatenated datasets that were recovered were also subjected to a process of serial stripping of groups of sites with incrementally increasing rates of variation; followed by a calculation of the probability of selecting either one of three proposed tree topologies (Figure 5:1) using a combination of topology comparison tests. The probability for this analysis was estimated using the approximately unbiased test (AU), the Kishino-Hasegawa test (KH) and the Shimodaira-Hasegawa test (SH) in the program CONSEL (Hidetoshi Shimodaira & Hasegawa, 2001) and was completed with help and support from Peter Foster at the Natural History Museum, London.

5.3 Results

5.3.1 Multi-gene Phylogenetic Analysis of the Kinetoplastids

Of the 599 trees resulting from the automated BLASTp analysis of all the predicted functional proteins from *T. brucei*, seventy-five presented monophyly of the kinetoplastids. These 75 were sorted into three datasets, based upon which topology they presented; see Figure 5:1. Fifty-eight of the tree topologies recovered a grouping of *T. brucei* and *T. cruzi* to the exclusion of *L. major*. The two other possible topologies were largely unrepresented, both being similar in number, as shown in Figure 5:2. The pattern is interesting as it not the expected outcome if the *Trypanosoma* are to be considered paraphyletic instead this seems to support monophyly. The 17 datasets that did not support monophyly of the *Trypanosoma* are therefore theoretically the product of horizontal gene transfer, hidden paralogy, or phylogenetic reconstruction artefact occurring amongst these datasets.

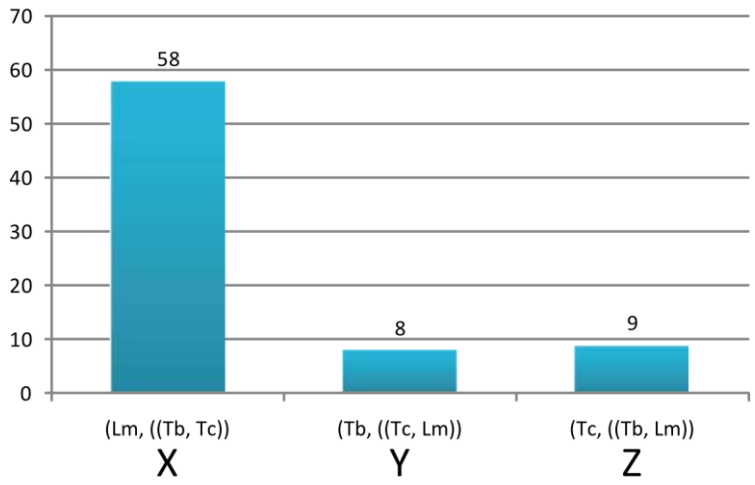


Figure 5:2 - The bar chart represents the number of trees found supporting each topology. Note the considerably smaller support for topologies Y and Z. This pattern is not to be expected if there was support for a paraphyletic grouping of the trypanosomes. Note also the presence of some support for

datasets Y and Z, suggesting the presence of differential topologies and possible hidden paralogy, HGT and phylogenetic reconstruction artefact amongst the datasets.

Datasets X, Y, Z and the full concatenation were subjected to a range of phylogenetic topology tests and methodologies. The results of these analyses can be viewed in Figure 5:3; each dataset is represented by a different colour, which changes intensity depending on the level of support for each topology under a specific methodology. For example, dataset X is represented in green and has full support for all tests for the topology of (Lm, (Tb, Tc)); consequently, there is no support within this dataset for any other topology. Dataset Y, conversely, shows differential support values across multiple topologies with the majority of support represented in that of topology X. Dataset Z recovers near full support for the grouping of *T. brucei* and *T. cruzi*, with an insignificant posterior probability value under the RAxML maximum likelihood test for its own topology. The complete concatenated dataset (purple) recovers full support across all tests and methods for the grouping of *T. brucei* and *T. cruzi* together to the exclusion of *L. major*.

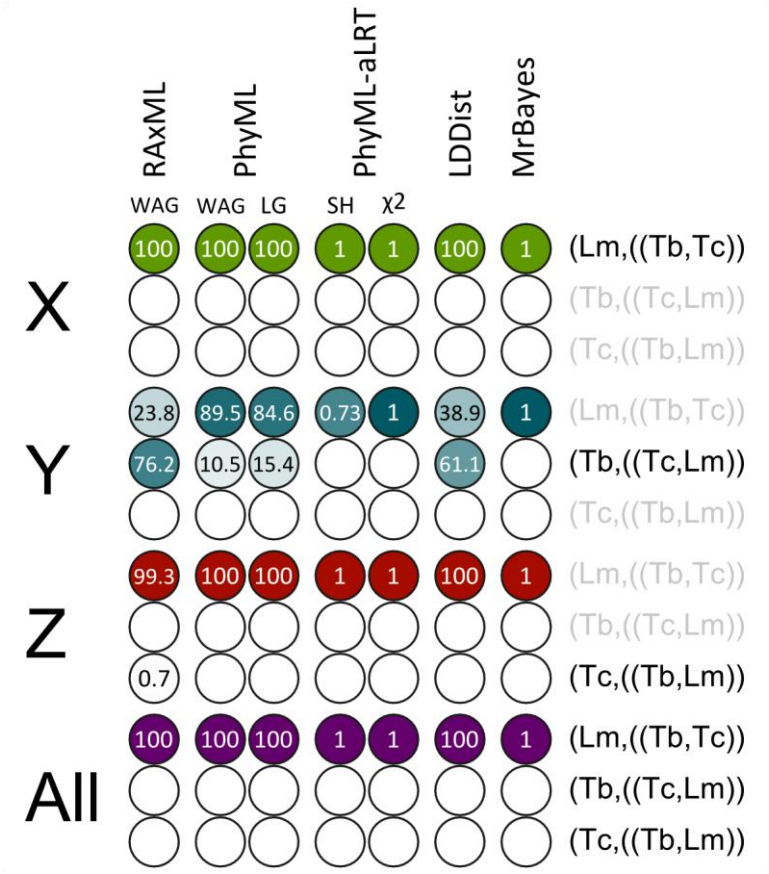


Figure 5:3 - This figure shows the support values returned for each topology for the three datasets (X, Y and Z) and the total concatenated dataset. Main topologies for each dataset are shown in black, those that are not the predominant topology are shown in grey. Note that the concatenated datasets for Y and Z, although derived from single cell analyses that supports an Y and Z topology (Fig. 5.1) once concatenated together strongly support topology X. This therefore suggests the source of error here is phylogenetic artefact and not HGT or hidden paralogy.

The resulting topology for the full concatenation (of datasets X, Y and Z) is shown in Figure 5:4. The topology is based on the results from the MrBayes comparison and support values from each test are shown in the order they appear in Figure 5:3.

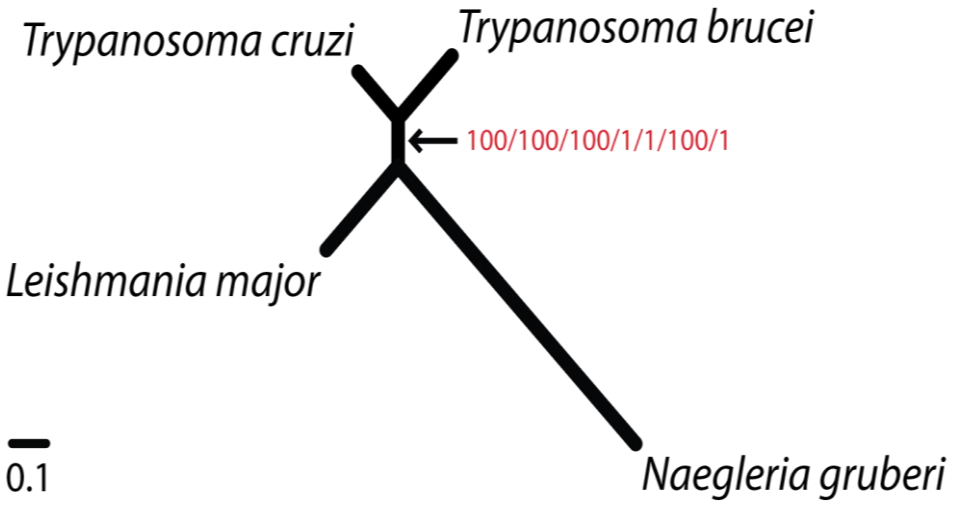


Figure 5:4 - A MrBayes topology generated from the concatenation of all three datasets. The arrow indicates full support for the monophyly of the Trypanosoma based on all models used, shown in Figure 5:3.

The conflicting support present in dataset Y is a result of the phylogenetic analyses performed on the eight conflicting sets of gene markers constituting this dataset and the associated concatenated alignment. In order to better understand the varied phylogenetic signal present in this dataset, separate analyses were performed for each of the eight gene-markers. The trees shown in Figure 5:5 are the topologies constructed with MrBayes; all relevant phylogenetic test results are indicated. The trees in green (A – C) represent topologies that support the monophyly of the Trypanosoma, whereas blue trees (D – G) recover a topology where *L. major* and *T. cruzi* are grouped, and the final red tree groups *L. major* and *T. brucei*. Note the weak support values present in the majority of trees, especially those that initially showed support for topology Y.

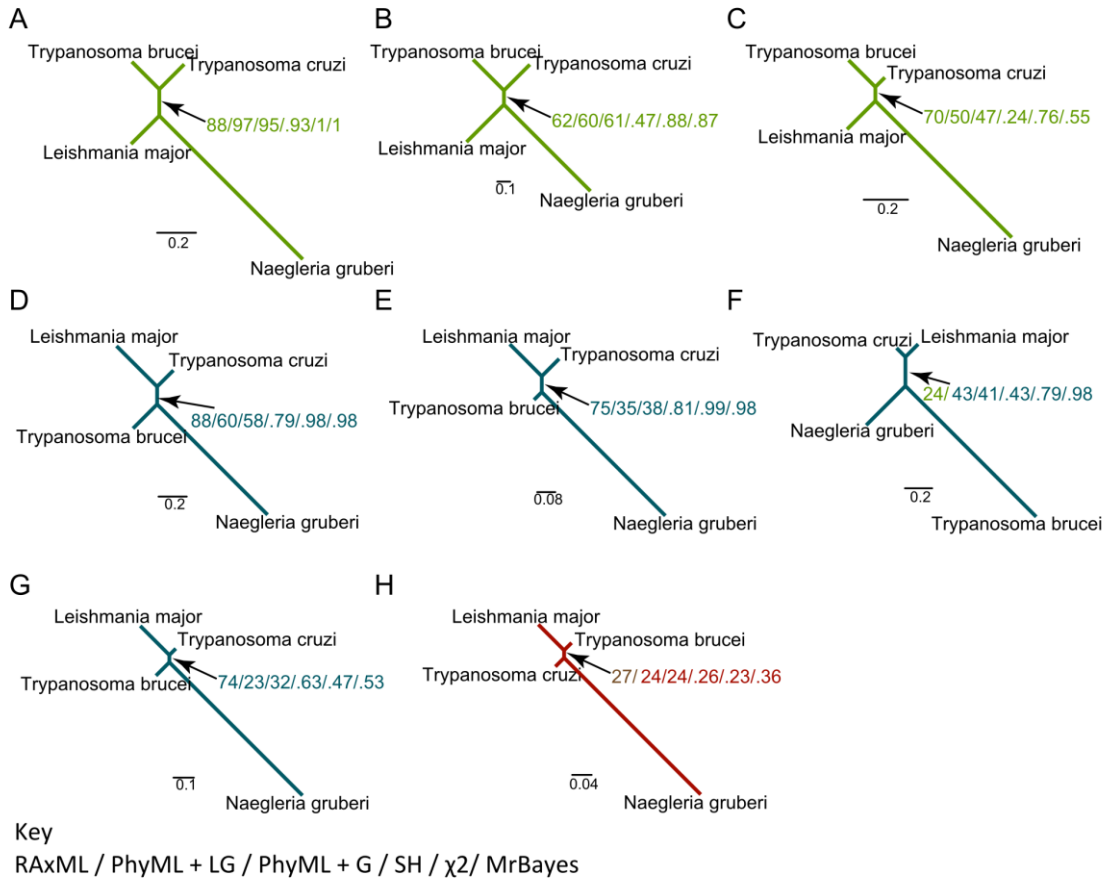


Figure 5:5 - A set of eight individual phylogenies representing the gene-markers within concatenated dataset Y. Note the low support values for the branching order of the taxa especially in the blue (D-G) trees which would be expected to have higher support given that they represent the topology of the dataset. Thus, the mixed support for the different topologies is indicative of the conflicting signal within dataset Y.

5.3.2 Paralogue Mirror-Tree Analysis

Amongst the 75 datasets, eight datasets showed the presence of reciprocal rooting of paralogous genes within the kinetoplastids. These eight datasets were concatenated and subjected to the same methods and analysis as before, in order to produce a 'mirror-tree', so called as the tree shows a particular topology and its 'mirror image' in two distinct clades when the tree is rooted between them. The analysis was conducted twice, firstly with only the two *Trypanosoma* genomes with *Leishmania major* and,

secondly, the former three genomes with the addition of *Naegleria gruberi* outgroup for the eight gene datasets. In both cases the same topology was recovered with full support across all tests and methods for the grouping *T. brucei* and *T. cruzi* together to the exclusion of *L. major* (Figure 5:6).

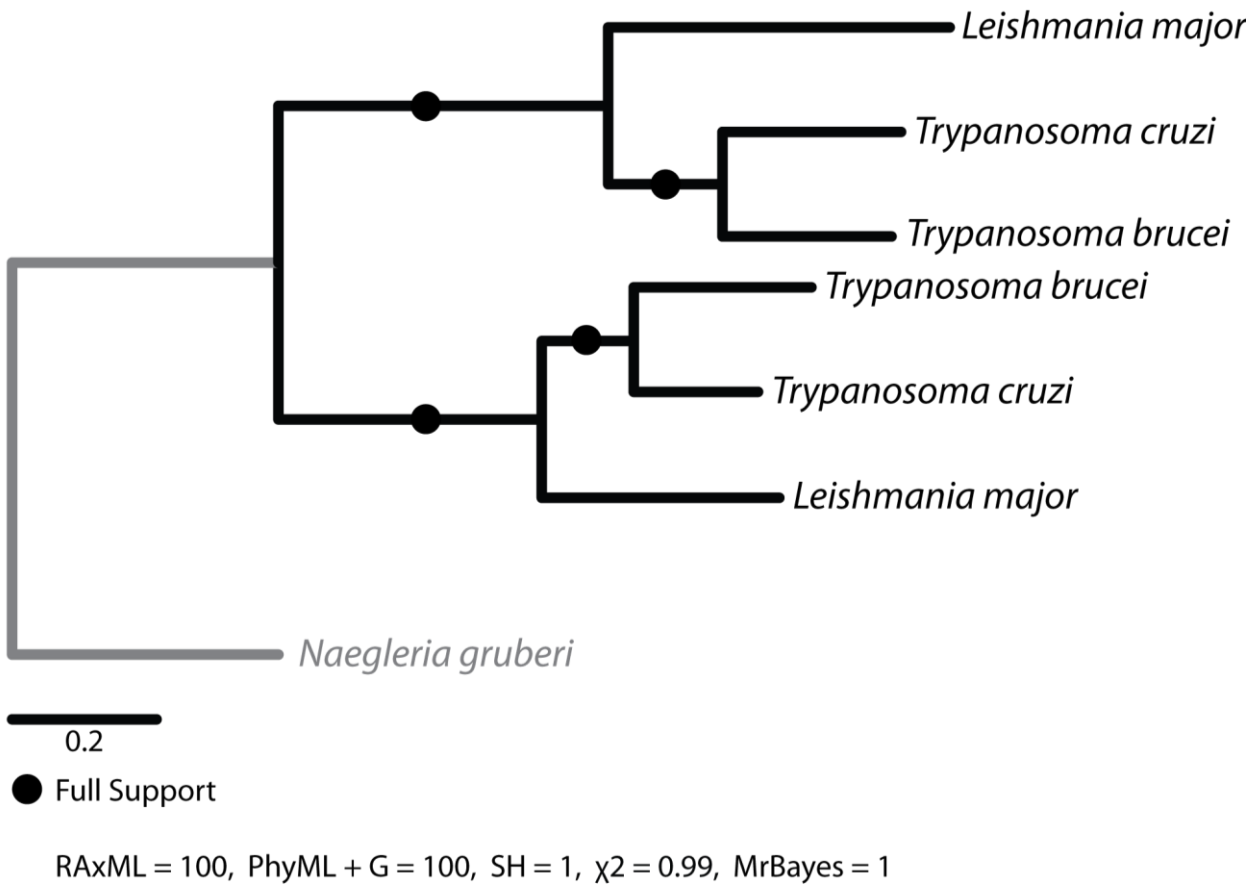


Figure 5:6 - A MrBayes topology generated from eight reciprocally rooted paralogous datasets indicating full support for the monophyletic grouping of *T. brucei* and *T. cruzi*. *N. gruberi* is shown in grey as it was included in a secondary analysis; the same topologies and almost identical support values were recovered

5.3.3 Phylogeny with Increased Taxa and Reduced Gene Sampling

To account for any effects due to artefacts created by long-branch attraction (LBA) (Herve Philippe, et al., 2005), three other closely related species were added to the

phylogenetic analysis. These were *Leishmania braziliensis*, *Trypanosoma vivax* and *Eulgena gracilis*. The topology support values are colour coded on the tree displayed in Figure 5:7. The green branch represents the monophyly of trypanosomes to the exclusion of both *Leishmania* species and shows very strong support; all other branch support values within the tree are generally very high. This finding adds some further support to the hypothesis that the genus of *Trypanosoma* is monophyletic.

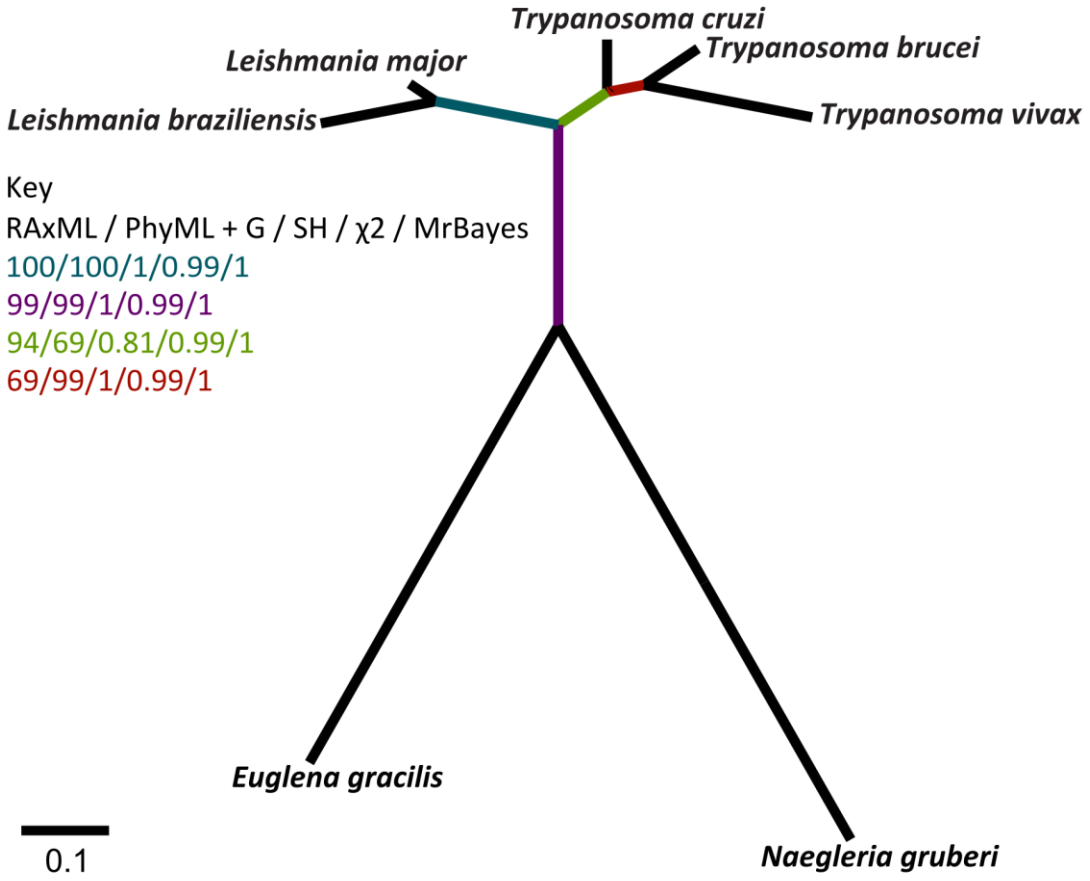


Figure 5:7 - The inclusion of further taxa, at the expense of gene family number and sites available for phylogenetic reconstruction also to confirm a largely well supported monophyly for the Trypanosoma.

5.3.4 Serial-Stripping of Fast Evolving Sites

With the help of Peter Foster at the Natural History Museum, London, the three topologies for the kinetoplastids, see insert of Figure 5:1, were tested using the final concatenated alignment (see Figure 5:3, the dataset represented by 'all' which contains 36,278 sites) using a series of site-stripping approaches combined with alternative topology tests.

Firstly, the alignment was tested for the best substitution model under the Akaike Information Criterion (AIC) in the program PROTTEST (Abascal, Zardoya, & Posada), which is somewhat similar to the program MODELGENERATOR (Keane, et al., 2006) which was used previously in this chapter. It suggested that the model with the parameters WAG+I+G+F was the best, and so these values were used with the program CONSEL (Hidetoshi Shimodaira & Hasegawa, 2001). CONSEL calculates a p-value (probability) using several testing procedures, such as the Shimodaira-Hasegawa test, the Kishino-Hasegawa test and the approximately unbiased test.

The results for these tests were a set of 1000 sites were removed at each test, up until the removal of 7000 sites (as under the model WAG+I+G+F it became difficult to accurately optimise the tree and model parameters after this number of removed sites) probably because of the removal of useful sites, are shown in Table 5:1.

	AU/KH/SH Tests							
Dataset	Full Sites	-1000 Fastest Sites	-2000 Fastest Sites	-3000 Fastest Sites	-4000 Fastest Sites	-5000 Fastest Sites	-6000 Fastest Sites	-7000 Fastest Sites
X	1/1/1	0.999/1/1	1/1/1	1/1/1	1/1/1	0.913/1/1	0.951/1/1	1/1/1
Z	6e-46 /0/0	0.001/0/0	4e-37/0/0	4e-37/0/0	5e-74/0/0	0.087/0/0	0.049/0/0	2e-05/0/0
Y	4e-12/0/0	3e-45/0/0	9e-05/0/0	1e-05/0/0	5e-99/0/0	5e-65/0/0	5e-38/0/0	8e-80/0/0

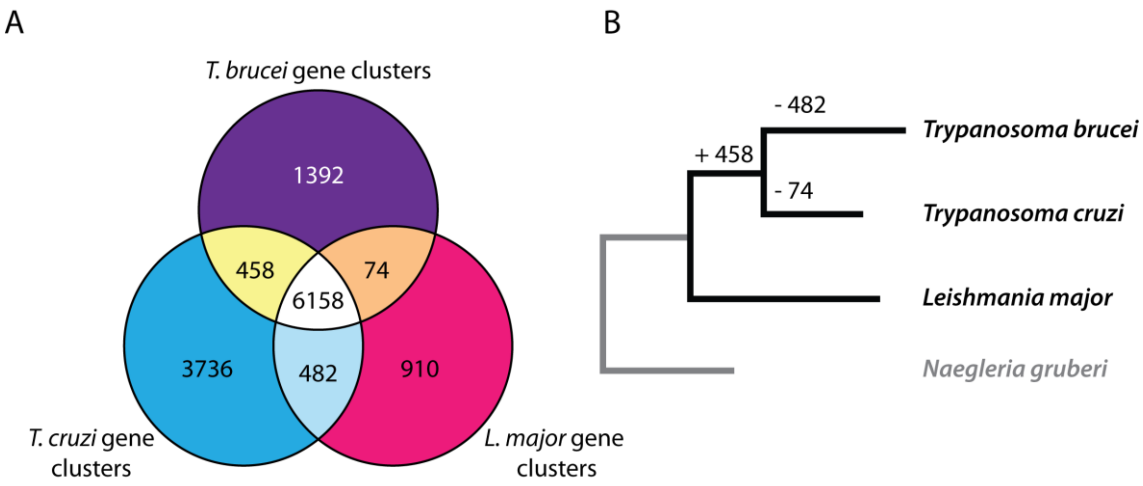
Table 5:1 - Results from the program CONSEL for the three tree topologies of the trypanosomes and *N. gruberi* showing full support under all conditions for the topology (Ng, (Lm, (Tb, Tc))) and confirmed as fast sites are serially removed. AU = Approximately Unbiased test, KH - Kishino-Hasegawa test, SH - Shimodaira-Hasegawa test.

In all cases, as we can see in Table 5:1, the best tree, based on the three statistical methods used, was topology X - (Ng, (Lm, (Tb, Tc))), which indicates the monophyly of the Trypanosoma. Moreover, it was often the case that all of the tree comparison methods in CONSEL calculated little to no support for the other proposed phylogenetic relationships.

5.3.5 Polarised Kinetoplastid Phylogeny and Gene Gain and Loss

By understanding the relationship of shared genes between the genomes of the three trypanosomatid taxa (Tb, Tc, Lm), as presented in (El-Sayed, Myler, Blandin, et al., 2005), patterns of gene loss and acquisition across the kinetoplastid tree may be polarised (Figure 5:8). This analysis confirms a large-scale gene loss event specific to the *T. brucei* branch (El-Sayed, Myler, Blandin, et al., 2005). As *T. cruzi* and *L. major* are both intracellular parasites and *T. brucei* is extracellular we posit that the pattern of loss could relate to the evolution of extracellular parasitic life cycles. This is in direct

contrast to many other conclusions based on genome sequencing of parasites, where gene loss seems to specifically correlate with adaptation to intracellular life (Katinka et al., 2001). It is not clear at present how these genes losses relate to specific cellular and parasitic mechanisms.



Recreated from El-Sayed et al. (2005). Science 309(5733): 405.

Figure 5:8 - (A) A Venn diagram depicting the genes shared amongst the three trypanosome genomes. (B) The figures from the Venn diagram can be mapped onto a topology supporting the *T. brucei* and *T. cruzi* monophyly. The position of the figures suggests possible gene acquisition and loss events in their evolutionary past.

5.4 Discussion

The data presented in this chapter attempts to resolve the contentious nature of the monophyly (Hamilton, et al., 2004; Stevens, 2008; Stevens & Gibson, 1999; Stevens, Noyes, Dover, & Gibson, 1999; Stevens, et al., 2001) or paraphyly (A. Hughes & H. Piontkivska, 2003; A. L. Hughes & H. Piontkivska, 2003; Piontkivska & Hughes, 2005) hypotheses for the phylogeny of the *Trypanosoma*. We used several robust phylogenetic techniques (including fast-ML and Bayesian) adapted for analyses of a very few taxa but encompassing 'whole genome' based analyses. The analyses used,

multi-gene concatenated alignments (super-matrices) and selected the most appropriate out-group taxa that were currently available. This strategy recovered a near-total support for the monophyly of *Trypanosoma brucei* and *Trypanosoma cruzi*. We therefore have presented a newly resolved phylogenetic analysis of the major branching order of the kinetoplastids and a strategy for resolving phylogenies of whole genome datasets with low taxon sampling (3-4 genomes). The rejection of the monophyletic relationship of the *Trypanosoma* appears to be very weak; indeed of the initial 599 gene families only seventeen (see Figure 5:2 - dataset Y and Z) appeared to suggest a different topology. However, on further analysis all of the genes in dataset Z returned a phylogeny supporting the monophyly of the trypanosomes along with four of the eight genes in dataset Y Figure 5:3. This suggests that only five genes (from dataset Y), or <1%, reject the hypothesis that the trypanosomes form a monophyly. In conclusion we present very strong evidence to support the monophyly of the trypanosomes. These data enable us to polarise gene gain and loss events within the *Trypanosoma.*

The method of serially stripping fast evolving sites from our protein alignment datasets was used to help control for the problems associated with using a relatively distantly related out-group, in this case *Naegleria gruberi*, which can cause problems with LBA (H. Philippe, 2000; Herve Philippe, et al., 2005). Similarly, the paralogue mirror-tree analyses were also completed to account for the selection of a relatively distantly related out-group because this approach allowed us to perform an analysis with and without the out-group, confirming this had no effect on our results.

The processes outlined here, i.e. using serial site-stripping methods and mirror-trees (reciprocally rooted paralogous gene trees) after the automatic tree topology building process (with 'Darren's Orchard') has been completed for all the gene families within an entire genome and is proposed as a new systematic method for investigating phylogenetic conflicts. Specifically, these methods can be adapted for addressing phylogenetic conflict hypotheses among trifurcated branching relationships, where taxon sampling is limited to a few whole genomes and where selection of distantly related taxa as an out-group is the only current viable option.

A further outcome of the analysis in this chapter suggests that the prediction and identification of a set of synapomorphic characters, such as gene fusion and fission events, could be of use to help further resolve any contentious branching orders and topologies that exist for the kinetoplastids. It was therefore decided that this phylogenetic group of taxa should be included as a dataset for use with my program fdfBLAST. Indeed, the results of the analysis of the fdfBLAST comparison of the *Trypanosoma brucei*, *Trypanosoma cruzi*, *Leishmania major* and *Naegleria gruberi* genomes can be found in Chapter 6.

6 Four-way Genome Analyses using fdfBLAST on Five Calibration Datasets from across the Tree of Life

6.1 Introduction

The approach I took for the identification of datasets to test with fdfBLAST was to, firstly select a testing dataset (Chapter 3) and secondly to select a series of calibration datasets (where the phylogenetic topologies have been previously strongly supported but could also benefit from the addition of shared derived characters, SDCs). Thus, the aims of this chapter were to test the performance of the fdfBLAST program across a set of known phylogenetic relationships, which will consist of four taxa each. This will allow for the establishment of the rate of fusion and fission events and also their distribution within the different groups being tested, potentially allowing us to draw comparisons between the different biologies (the life processes of an organism) of the selected taxa and to potentially observe how fusion and fission events vary across metabolic and cellular pathways and processes. We can then compare our conclusions to previous analyses identified on manual genome comparisons (Durrens, et al., 2008; Nakamura, et al., 2006), group specific analyses and *ad hoc* discoveries (Stechmann & Cavalier-Smith, 2002, 2003).

6.1.1 Four-way Genome Dataset Selection for Calibration Purposes

The use of only two genomes in an fdfBLAST analysis was deemed to be too few to make any substantial impact on contentious or previously confirmed topologies due to the lack of polarisation of synapomorphic characters and also making it difficult to formulate predictions of fusion rates across distinct phylogenetic relationships

(Chapter 4). Nevertheless, the previous analysis was essential as a test dataset to make sure fdfBLAST was producing viable results. Furthermore, two-genome analyses are of limited use in the case of trifurcated tree topologies as they do not allow for a comparison between the three or more branches (Nakamura, et al., 2006). This is due to two-way analyses essentially placing the root between any excluded taxa and the two taxa being analysed, directly because the third taxon or any other out group taxa are not a part of the comparison, although this depends on where the gene fusion event is found to be.

Therefore, to tackle this problem it was decided that the number of taxa per dataset analysed would be increased to four, hopefully, to allow for fusions to be established between the branches of a trifurcated tree topology and potentially the out-group. The predicted fusions would then be subjected to more bioinformatic analysis using the gene-by-gene phylogeny pipeline 'Darren's Orchard' (T. A. Richards, et al., 2009) to produce a set of reference tree topologies to help polarise and confirm the candidate fusion and fission events across the underlying species tree.

Five representative groups were chosen: the Discicristata (T. Cavalier-Smith, 1998; Hamilton, et al., 2004; Hampl, et al., 2009; Stevens & Gibson, 1999; Stevens, et al., 1999), the Vertebrata (Cotton & Wilkinson, 2009; Frederic Delsuc, et al., 2005; Gillis, St John, Bowerman, & Schneider, 2009; Stuart, Moffett, & Leader, 2002; Townsend, López-Giráldez, & Friedman, 2008), the Viridiplantae (Gao, Su, & Wang, 2010; Hedges, 2002; Medina, 2005; Parfrey et al., 2010; Pryer, Schneider, Zimmer, & Ann Banks, 2002), the Fungi (Galtier, 2001; Kovalchuk & Driessen, 2010; Y. Liu et al., 2009; McLaughlin, Hibbett, Lutzoni, Spatafora, & Vilgalys, 2009; Schoch et al., 2009), and the

Deuterostomia (Frederic Delsuc, et al., 2005), all representing groups from across the eukaryotic tree of life. From each group a subset of four taxa were chosen where a known or proposed phylogenetic relationship had been previously established, with good support, by phylogenomic analysis. These taxa were often chosen based on their status as a model organism and so the genome sequence is often finished to a high standard and well annotated (in order to minimise the prevalence of annotation errors; on the understanding that the genome projects of model organisms should be better annotated) or for their position within the tree of life (to help infer ancient evolutionary relationships - for example, the inclusion of derived ancient out groups). The taxa selected for the five 4-by-4 analyses are reflected in Figure 6:1 and are explained in the following five subsections.

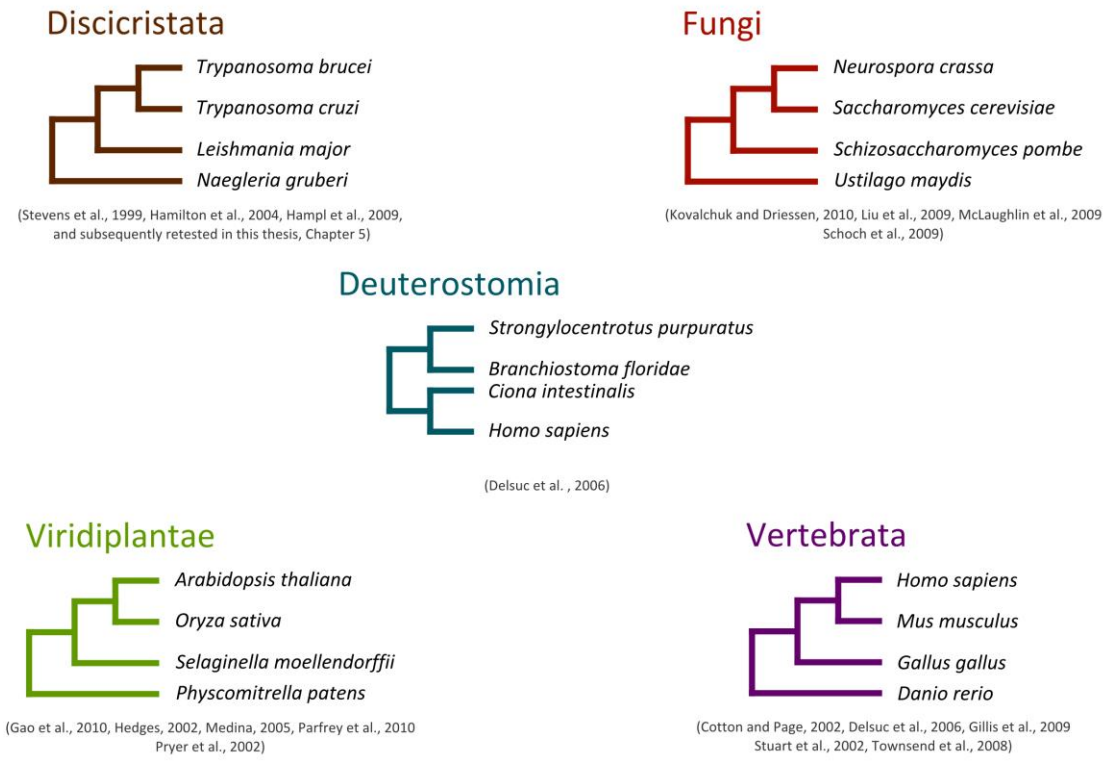


Figure 6:1 - A set of five reduced-taxon phylogenetic topologies based on a consensus of the topologies from a selection of the latest analyses from each group. Each cladogram shows the consensus inferred branching order for each taxon considered in the four-way analyses.

6.1.1.1 Extended Viridiplantae Dataset

The two-way plant analysis, described in Chapter 4, was extended to include *Physcomitrella patens* (Rensing et al., 2008) and *Selaginella moellendorffii* (http://genome.jgi-psf.org/Selmo1) as out-groups to the two previously analysed land plants. *Physcomitrella patens* is a moss (Bryophyta) and represents the most ancient derived taxa within our group, most interestingly it is not a vascular plant (as it does not have roots or possess xylem or phloem) therefore taking the role of the out-group in this case (Willis & McElwain, 2002). Whereas *Selaginella moellendorffii* is a lycophyte, and is distinguished by the possession of microphylls (leaves with a singular vein) it is therefore suggested to be an anciently derived branch of all extant vascular plants (Willis & McElwain, 2002). Additionally, as the previous analysis, described in Chapter 4, was undertaken with only two plant genomes this 4-way analysis may also allow for a further comparison to understand how the addition of more taxa affects the results of the fdfBLAST program.

6.1.1.2 Fungi

The genomes of *Neurospora crassa* (Galagan et al., 2003), *Saccharomyces cerevisiae* (Giaever et al., 2002) (the classic baker's yeast), *Schizosaccharomyces pombe* (Wood et al., 2002) (fission yeast), and *Ustilago maydis* (Kamper et al., 2006) (the causative agent of corn smut) were chosen as examples of relatively well annotated fungi with the first three belonging to the phylum Ascomycota and the last representing the phylum Basidiomycota. Both phyla are described as monophyletic (Hibbett et al., 2007; James, et al., 2006) making the Ascomycota the sister group to Basidiomycota where together they make up the subkingdom named Dikarya (Hibbett, et al., 2007).

6.1.1.3 Vertebrata

The genomes of *Danio rerio* (http://www.sanger.ac.uk/Projects/D_rerio) (zebra fish), *Homo sapiens* (Venter et al., 2001), *Gallus gallus* (ICGSC, 2004) (red jungle fowl), and *Mus musculus* (Waterston et al., 2002) were chosen as they represent four distinct vertebrata that are used as well established model organisms, including two mammals, one bird and one fish and have been the subject of numerous genomic and transcriptomic studies and so have comparatively well assembled and annotated genomes. The vertebrata is a subphylum of the chordates and includes, amongst others, the groups of fish, amphibians, reptiles, mammals and birds. The representative species, above, include two mammals (*Homo sapiens* and *Mus musculus*) and one bird (*Gallus gallus*) which belong to the superclass Tetrapoda (four-limbed) and the fish (*Danio rerio*) which belongs to the superclass Osteichthyes (bony fish) (Cotton & Wilkinson, 2009; Frederic Delsuc, et al., 2005; Gillis, et al., 2009; Stuart, et al., 2002; Townsend, et al., 2008).

6.1.1.4 Discicristata

The monophyly of the Trypanosoma (kinetoplastids) has been demonstrated (e.g. Hamilton et al (Hamilton, et al., 2004)) and subsequently re-evaluated in this thesis, Chapter 5, using a phylogenomics approach, although it remains a contentious relationship when environmental kinetoplastid-like 18S rRNA sequences are used (Piontkivska & Hughes, 2005). Consequently, it is a prime candidate dataset to test with my program fdfBLAST because additional genomic synapomorphies such as gene fusions would be very important for confirming the kinetoplastid branching relationships. The genomes of *Trypanosoma brucei* (Berriman, et al., 2005), *Trypanosoma cruzi* (El-Sayed, Myler, Bartholomeu, et al., 2005), *Leishmania major*

(Ivens, et al., 2005) and *Naegleria gruberi* (Fritz-Laylin et al., 2010) were chosen due to the three kinetoplastids previously demonstrated topology and *Naegleria gruberi* as it is currently the most closely related sequenced genome available to use as an out-group according to the Discicristata (Percolozoa and Euglenozoa) hypothesis (T. Cavalier-Smith, 1993, 2003). These four datasets will potentially allow us to build a set of fusion and fission events that can be mapped onto the topology, allowing us to retest this branching relationship.

6.1.1.5 Deuterostomia

The deuterostomia are a putative super-phylum and includes *Branchiostoma floridae* (Putnam et al., 2008) (commonly the lancelet or amphioxus), *Homo sapiens*, *Ciona intestinalis* (Dehal et al., 2002; Satou & Satoh, 2003) (the sea squirt) and *Strongylocentrotus purpuratus* (Sodergren, Shen, et al., 2006; SodergrenWeinstock, et al., 2006) (the purple sea urchin). The latter is interesting as it appears to possess a contingent of vertebrate-only genes, and genes that are usually found outside the group deuterostomia (SodergrenWeinstock, et al., 2006). The topology presented here in Figure 6:1 demonstrates that the tunicates are the closest living relatives of the vertebrates, as demonstrated by Delsuc et al. (2006) (F. Delsuc, et al., 2006); however, this is a contentious relationship as the cephalochordates (lancelets) have been depicted as the closest living relatives of the vertebrates based on a large concatenated protein datasets (Blair & Hedges, 2005; H. Philippe, Lartillot, & Brinkmann, 2005).

6.2 Methods

6.2.1 fdfBLAST Comparisons

fdfBLAST was run at a variety of e-value cut-off settings for each dataset, see Table 6:1, the lowest e-value used, 1e-10, was chosen for all datasets and then a sampling of higher, and so, more constrictive e-values were also used on some of the datasets. Where the extra e-values were run the results will show rates of fusion and fission events across the e-value range for the confirmed fusion events.

Datasets / E-values	1e-70	1e-60	1e-50	1e-40	1e-30	1e-20	1e-10
Viridiplantae							✓
Fungi	✓			✓			✓
Discicristata	✓		✓		✓		✓
Vertebrata							✓
Deuterostomia							✓

Table 6:1 - Ticks indicate that a dataset was subject to a fdfBLAST analysis at the relevant e-value. It should be noted that only select datasets were repeated at further e-value settings, in these cases the datasets were generally much smaller and so they took less time to run. Larger datasets, e.g. Viridiplantae would have taken much longer than 1 month to complete at further e-values.

6.2.2 Phylogenetically Informative Putative Shared Derived Characters

Where fdfBLAST recovered gene fusions or reversions that appear to be phylogenetically informative (where they are differentially distributed in two or more genomes), their alignments are recovered from the gene-by-gene phylogenetic analysis pipeline 'Darren's Orchard' and subsequently reanalysed by passing them through the pipeline once more. This second run was completed on the 'split domains' while the first run was conducted on the composite amino acid sequence. The primary ORF, identified by fdfBLAST, for the fusion or fission event, is taken from the alignment

and subjected to a more restrictive BLASTp analysis (by increasing the e-value cut-off) in the Darren's Orchard pipeline. This is to try and reduce the taxa set present in the resulting phylogeny when there are multiple hits for over sampled taxa in the phylogeny. Subsequently, the methods present in Chapter 2: Methods for phylogenetic reconstruction are also used in order to ensure the phylogeny has been performed under the best conditions possible. Whereby, multiple reconstruction techniques are employed, e.g. Bayesian as well as fast-ML, along with manual alignment masking for the sequence alignment instead of the automatic masking carried out by the GBLOCKS stage of the pipeline. Each phylogenetically informative SDC will be described under the relevant section and more generally in the final comparative results section.

6.3 Results

The results of the various fdfBLAST runs for the five 4-way analyses were manually assessed for the presence of a likely predicted fusion event by looking at each of the images output by fdfBLAST containing PFAM domains. As described previously the resulting set of sequences that represent the putative gene fusion events were subjected to the bioinformatic gene-by-gene phylogeny pipeline, Darren's Orchard, which produced a set of reference phylogenies. These phylogenies were then visually assessed for the support of the putative fusion and fission events by mapping discrete functional domains from the PFAM database onto the trees, as described in Chapter 2. This enabled us to view the locations of the gene fusion and fission events directly and so accept those that showed clear indications of these events. In order to be thorough in the acceptance of a fusion event the sequences representing the split-domains were checked and verified against their relevant genome project using the genome browser to view the genome scaffold. This allowed us to check for the mis-annotation of

predicted proteins, where two or more ORFs are identified as separate genes instead of one fused ORF, which can cause fdfBLAST to predict a fusion event as a false positive.

The five individual datasets are discussed below in the following sub-sections and are followed by a comparison between the datasets; all the accessions for the gene fusion and reversion events and other relevant information can be found in Section 10.

6.3.1 Extended Viridiplantae Dataset fdfBLAST Analysis

The Viridiplantae analysis predicted 24 candidate gene fusion events; of these only six showed the presence of genuine gene fusion events (once confirmation via phylogenetic analysis was completed) along with two reversion events (fissions) shown in Figure 6:2. Unfortunately, however, none of these were potentially phylogenetically informative because the putative synapomorphy was restricted to a single genome of the data available. As more genomes become available we predict that these fusion characters may become increasingly useful to investigate interrelationships among the tips of the plant phylogeny shown in Figure 6:1.

Of the remaining candidates three gene fusions were excluded due to paralogue and resolution problems in the subsequent phylogenies, so that we could not resolve the nature of the fusion or fission event (mostly due to the effects of LBA artefacts and/or lack of resolution within the tree topology). Four candidate gene fusions did not produce a tree topology when their sequences were run through 'Darren's Orchard' because the taxon and site sampling methods failed to accurately recover enough data

to generate a phylogeny, suggesting the genes analysed are too divergent to be informative. In total, these seven candidate gene fusions were excluded.

fdfBLAST made six prediction errors (false-positives) where, when analysed with the help of tree topologies, the fusion or fission event could not be identified, i.e. the resulting tree did not demonstrate a resolved differentially distributed protein domain architecture.

Each genome project has its own genome browser which allows you to look at the annotation of genes based on the scaffold data assembled from the sequenced organism (e.g. Aslett et al., 2010). We used these genome browsers to check all cases of apparent gene fission. This was achieved by accessing the genome projects of the taxa containing the predicted split ORFs/domains. If the two split ORFs/domains were to appear next to each other and in the same direction on the scaffold then it is very likely that they represent a mis-prediction for two ORFs instead of just one and so were excluded as putative annotation errors. No predictions were excluded due to the absence of any genome annotation errors within the predicted SDCs.

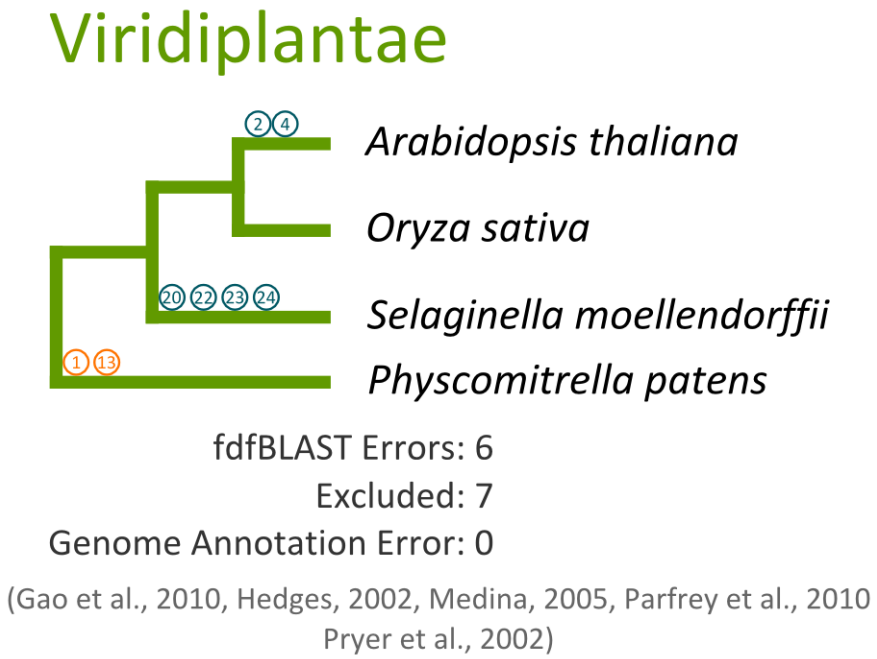


Figure 6:2 - A consensus topology for the Viridiplantae showing the branching order for the four taxa included in this analysis. The blue circles represent predicted gene fusion events and the orange reversion events. The numbers within the circles correspond to the accessions and tree topologies in Section 10.

6.3.2 Fungi

The fungi produced more hits than the Viridiplantae with 36 differentially distributed putative gene fusion and fission events between the four fungal genomes analysed. As shown in Figure 6:3 one genome annotation error existed within this dataset which was identified by accessing the genome viewer to check that the two split ORFs/domains were not next to each other on the scaffold, indeed in this case they were and they also shared consecutive accession numbers. Nine candidate gene fusion events produced no tree after analysis with 'Darren's Orchard' tree building pipeline (due to the identification of too few homologous genes or at the masking stage where there existed too few conserved regions of the alignment to build a phylogeny) and

three tree topologies were excluded due to lack of phylogenetic resolution between paralogues and apparent LBA (H. Philippe, 2000; Herve Philippe, et al., 2005) artefacts, which prevented resolution of the gene fusion or fission event. fdfBLAST also made three prediction errors, where subsequent trees did not identify a putative gene fusion or fission event.

Two of the predicted fusions showed some indication that they could be phylogenetically informative for the branching relationship between Ascomycota and Basidiomycota. This analysis also identified a gene fusion with a complex distribution across the fungi, present in *S. pombe* and *S. cerevisiae* but not present in *N. crassa* in contradiction to the established fungal phylogeny (James, et al., 2006; Y. Liu, et al., 2009) and indicating a potential HGT event between *S. cerevisiae* and *S. pombe*. The HGT event has recently been independently published by Slot et al. (2010) (Slot & Rokas, 2010). The recovery of this particular gene fusion and accurate identification of its taxon distribution is further confirmation of the efficacy of the fdfBLAST pipeline.

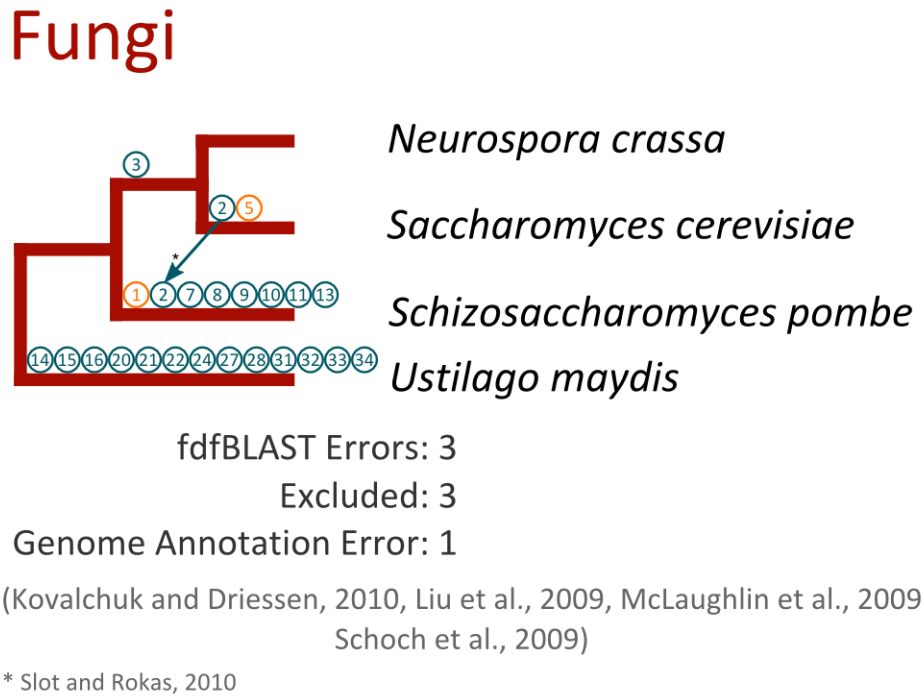


Figure 6:3 - A consensus topology for the Fungi detailing the branching order for the four taxa included in this analysis. The blue circles represent predicted gene fusion events and the orange reversion events. The numbers within the circles correspond to the accessions and tree topologies in Section 10. The HGT of a gene fusion reported by Slot et al. (2010) is also indicated.

6.3.2.1 Fungi: Phylogenetically Informative Datasets

Fusion 3 - Figure 6:4 and Figure 6:5 depict the topologies for a fusion event between the two ORFs/domains of SurE (Figure 6:4) and TTL (Figure 6:5), where each tree topology is based on a masked alignment phylogenetic analysis of the sequence respectively representing one of the domains. In Figure 6:4, note the presence of the fusion, represented by a large blue circle containing an 'F', at the base of the Ascomycota (Saccharomycotina and Pezizomycotina). This suggests that it may be a potential synapomorphy for them, thus it excludes the basidiomycetes and other groups. However, there is also a revision event present within the ascomycetes (specifically Pezizomycotina), represented by an orange circle containing an 'R', which does not allow us to resolve the monophyly of the fusion completely. However, it

follows that it is most parsimonious to suggest that one fusion event occurred and was followed by a later fission event of the second domain. Figure 6:5, the TTL domain, however does not show such clear support for this fusion event, with very low support. The tree topology also hints at several other cases of gene reversion (as marked on Figure 6:5) in the TTL domain phylogeny, however bootstrap support in this tree is too weak to conclude if this was the product of one or more fission events. Nonetheless the most parsimonious interpretation of this dataset is that it is a gene fusion in the last common ancestor of the Pezizomycotina and Saccharomycotina with one or more cases of gene fission. Please also be aware of the duplicate presence of *Ustilago maydis* in both trees, one written only with an accession, this is an artefact produced by 'Darren's Orchard' tree building pipeline and they were left in place as a marker to help identify seed sequences and trees, they both share the exact same amino acid sequence.

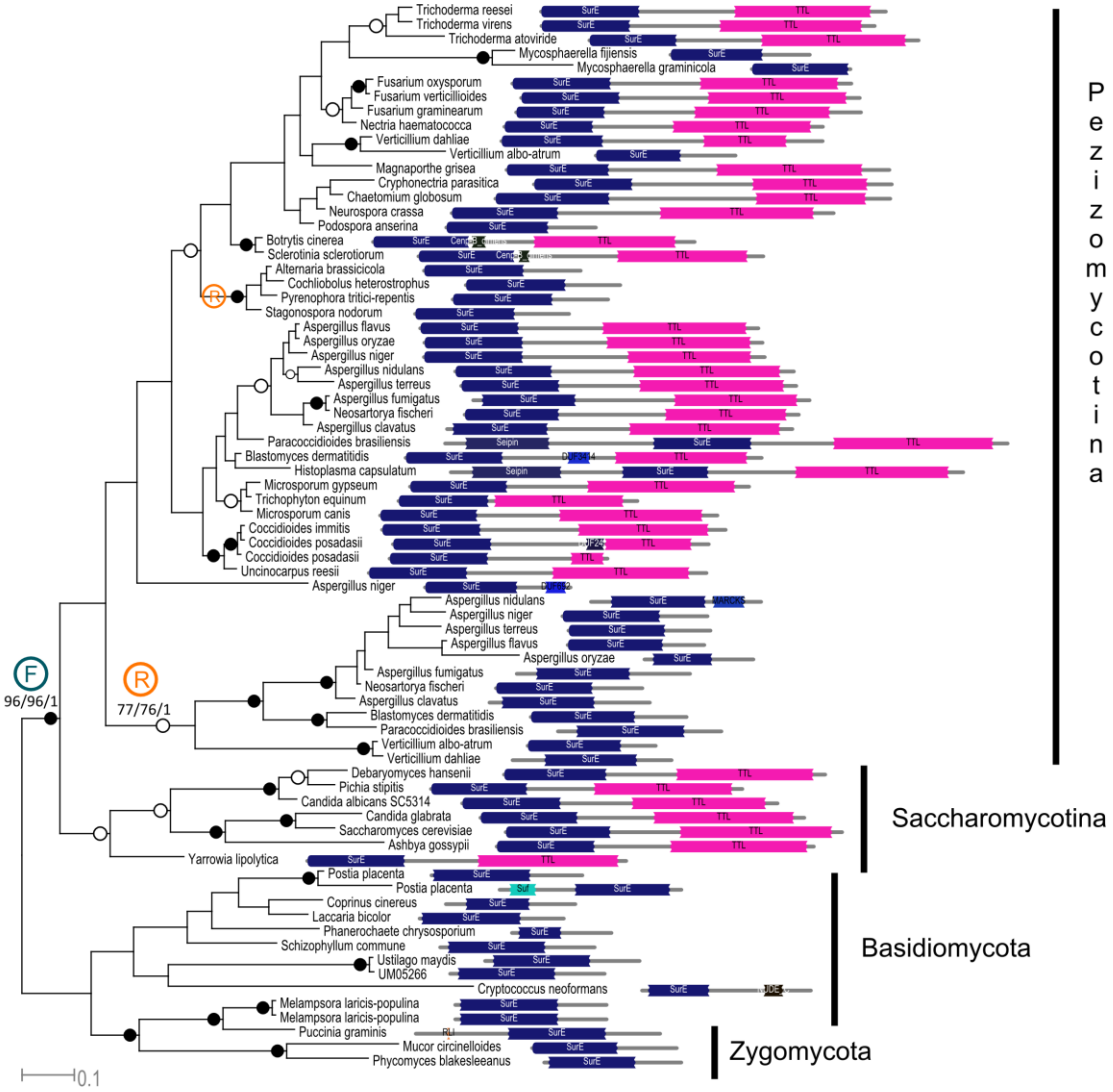


Figure 6:4 - A tree topology representing the SurE domain present in Fungi Fusion 3. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 are represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML. Additional cases of gene fission (reversion) are also illustrated, but we note these are not polarised by strong bootstrap support, so these additional cases are tentative and are not included in our summary statistics.

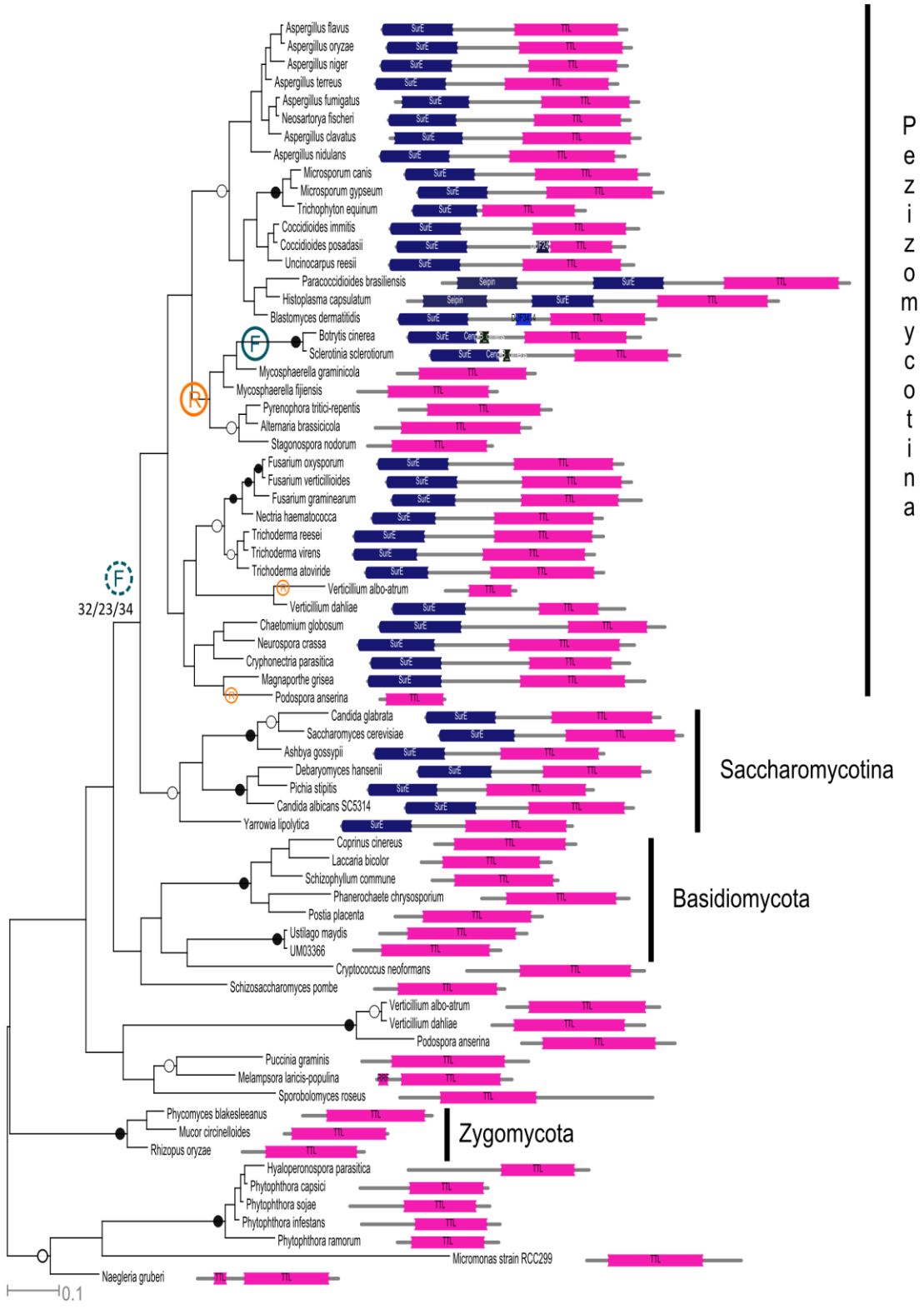


Figure 6:5 - **A tree topology representing the TTL domain present in Fungi Fusion 3. The tree topology is based on the results of a MrBayes analysis. Where Bayesian inference values and both the fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 are represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML. Additional cases of gene fission (reversion) are also illustrated, but we note these are not polarised by strong bootstrap support, so these additional cases are tentative and are not included in our summary statistics.**

Fusion 21 - Figure 6:6 and Figure 6:7 depict the topologies for a fusion event between the two domains of a tandem duplicated Allantoicase domain and a Ureidogly_hydro domain, where each tree topology is based on a masked alignment and phylogenetic analysis of a sequence, respectively, representing one of the domains. Both phylogenies suggest that the fusion event occurred within or prior to the basidiomycete radiation although support is weak and the phylogenies are inconsistent and several basidiomycete taxa do not possess the gene fusion character (e.g. *Cryptococcus*). The placement of the Zygomycota close to the Ascomycota (Saccharomycotina and Pezizomycotina) in Figure 6:8, with weak support, is in contrast to the accepted relationships within the Fungi (Adl, et al., 2005; Hibbett, et al., 2007; James, et al., 2006; Y. Liu, et al., 2009) suggests that the phylogenies are unresolved or hints at evidence of hidden paralogy. Figure 6:7 also shows inconsistent support for monophyly of fusion genes, especially as the fusion event appears in two separate clades across the tree, with prokaryotic taxa branching in-between. Because of this lack of support and inconsistent branching relationship it is difficult to definitively identify the ancestry of the gene fusion or relative fission events in the Basidiomycota. However, with further data and improved phylogenetic resolution this fusion gene may prove useful for polarising evolutionary relationships between and within the basidiomycetes.

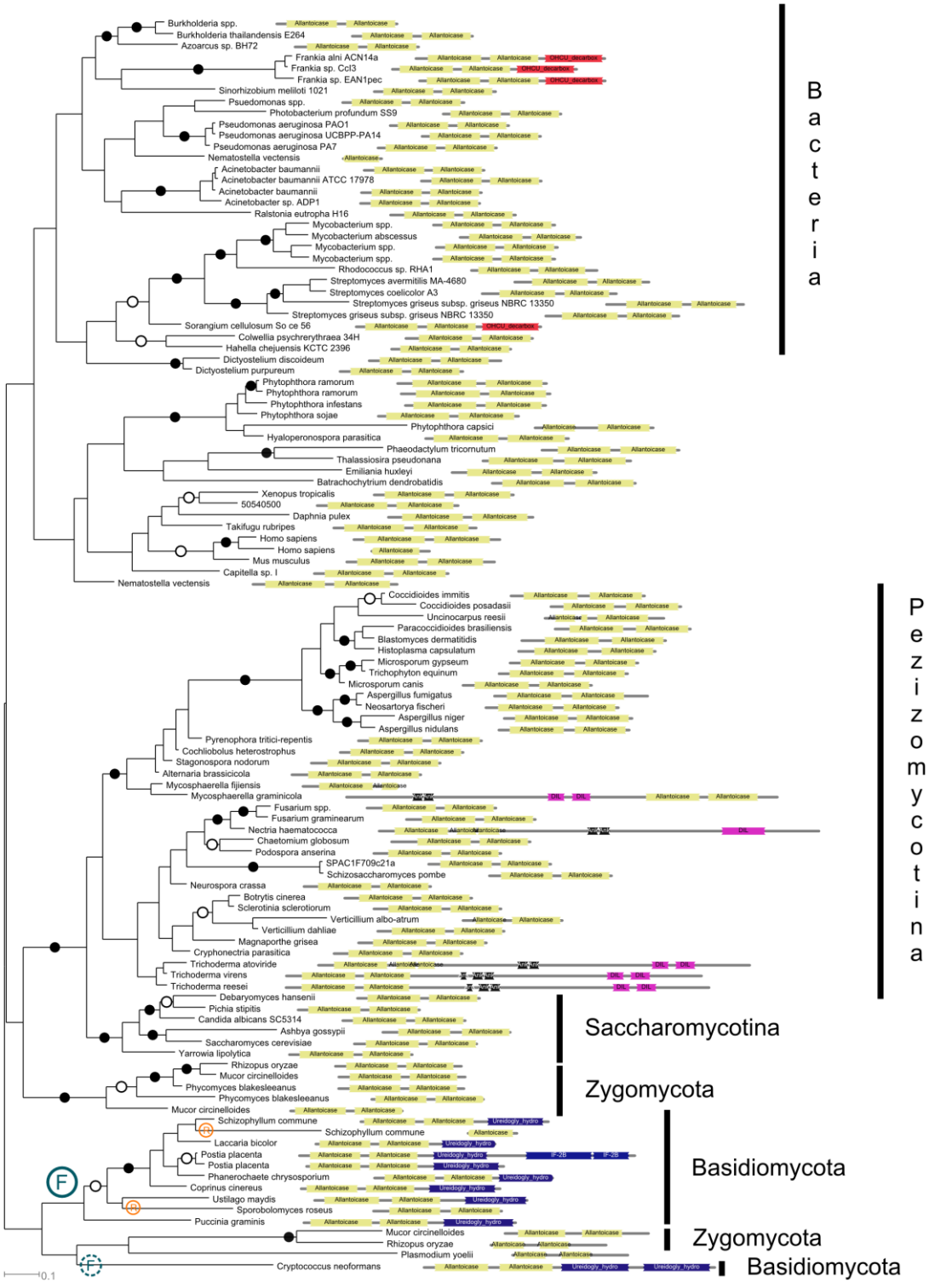


Figure 6:6 - A tree topology representing the Allantoicase Allantoicase domain present in Fungi fusion 21. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 are represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML. Note that placement of some basidiomycete gene fusions (e.g. *Cryptococcus*) within the tree are weakly supported and therefore it is currently not possible to polarize additional fusion/fission events within this clade.

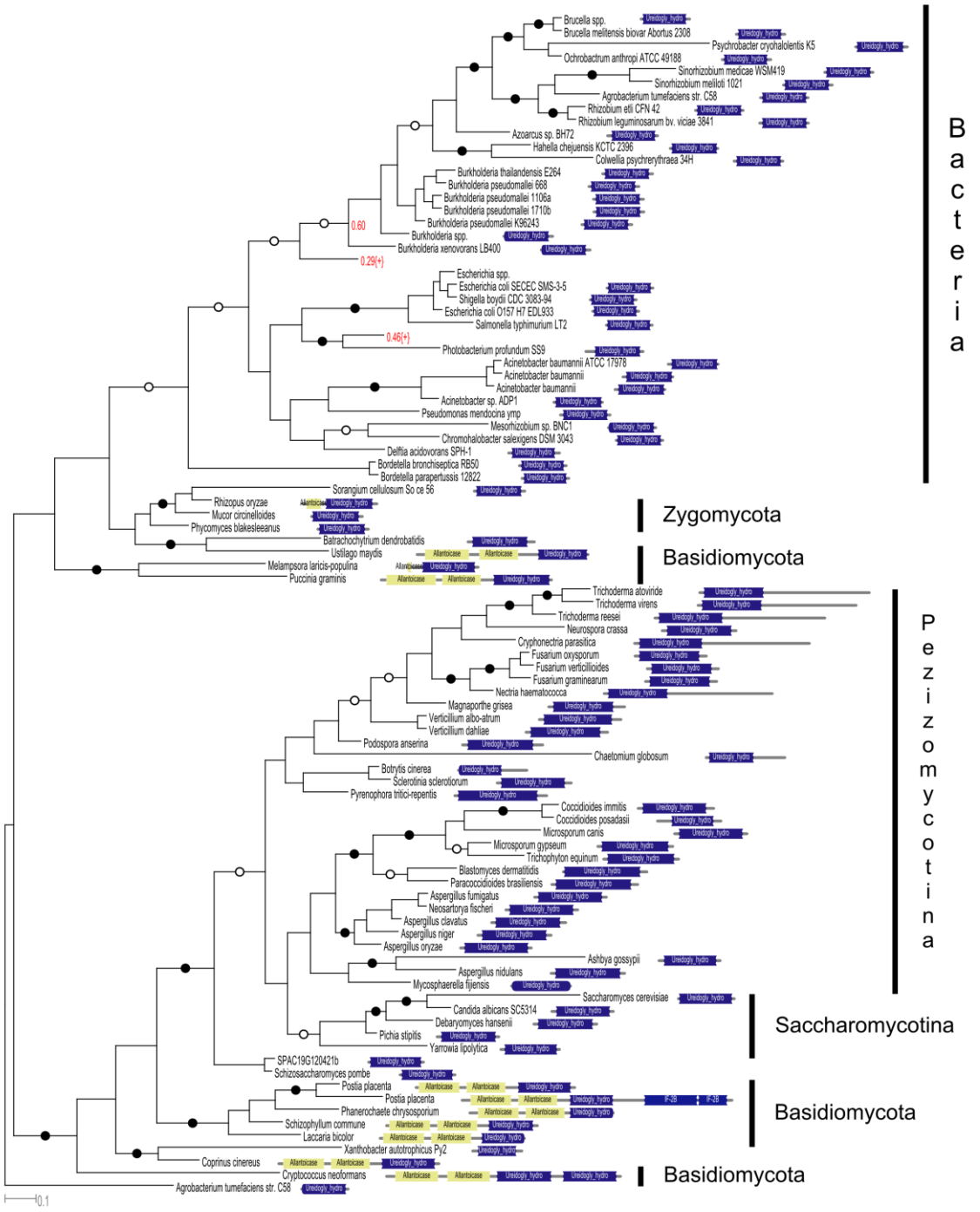


Figure 6:7 - A tree topology representing the Ureidogly_hydro domain present in fungi Fusion 21. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 is represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML. Note that placement of the *Agrobacterium* sequence among the some basidiomycete gene fusions means that it is currently not possible to polarize additional fusion/fission events within the Fungi.

Fusion 34 - Figure 6:8 and Section 11 (not shown in the text here as it is a large phylogeny spanning multiple pages) depict the topologies of a gene fusion event between the two ORFs/domains of Cys_Met_Meta_PP and GHMP_kinase_N with GHMP_kinase_C (Figure 6:8) where each tree topology is based on a masked alignment and phylogenetic analysis of a sequence representing one of the domains, respectively. As the gene fusion event only occurs in the basidiomycetes it can be potentially used as a synapomorphic marker for this clade; however, the support for the branching order and overall topology is very weak, but as the domain of Cys_Met_Meta_PP does not appear elsewhere on the tree it is reasonable to suggest the gene fusion event is valid.

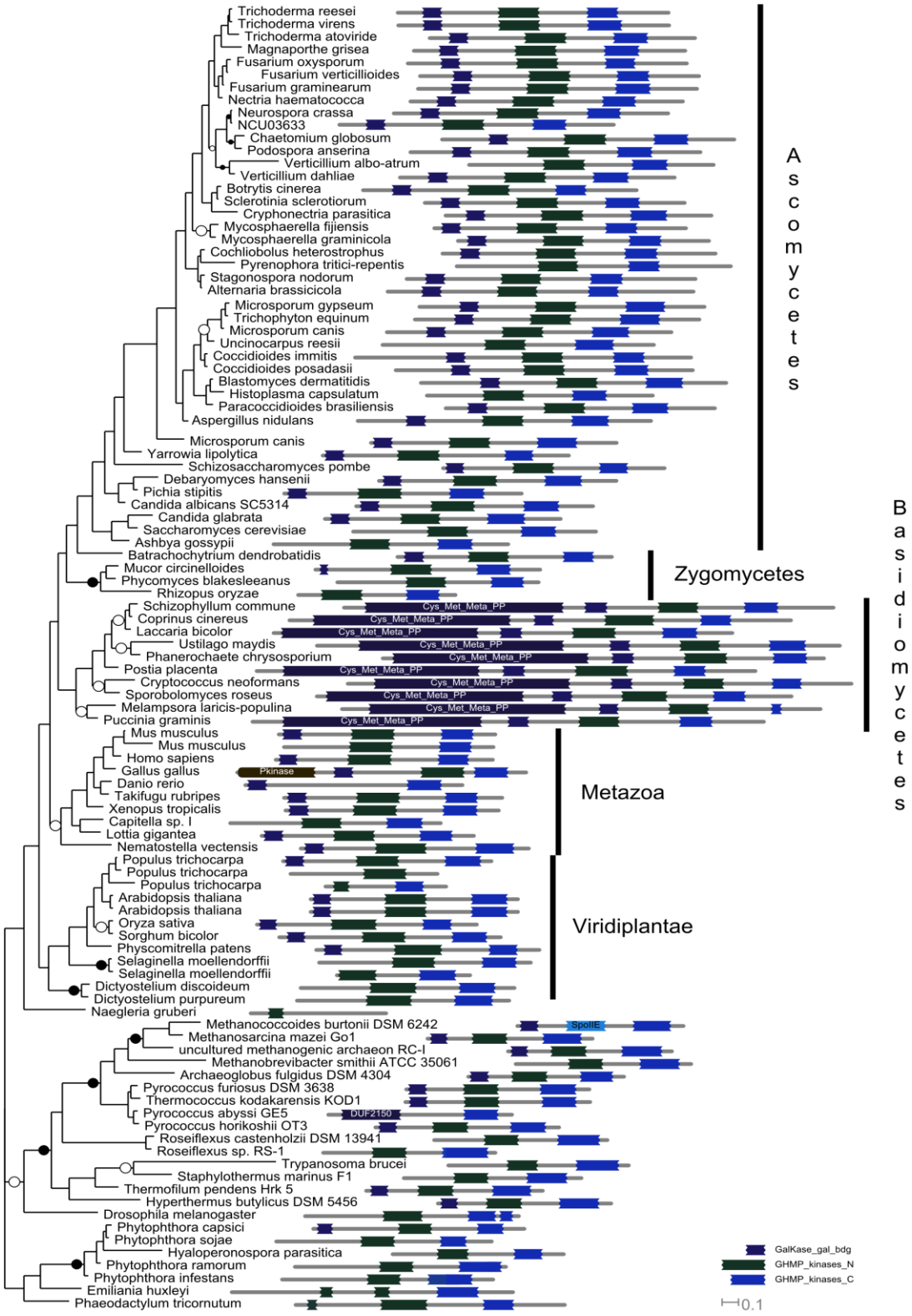


Figure 6:8 - A tree topology representing the GHMP_kinase_C domain present in fungi fusion 34. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 is represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML.

6.3.3 Vertebrata

The vertebrata analyses also produced a relatively large number of gene fusion events, 29 in total, shown in Figure 6:9. The same validation methods used previously were used here in order to determine whether or not the split domains/ORFs followed each other on the respective genome scaffolds. There were five genome annotation errors identified by comparison of genome scaffolds that resulted in fdFBLAST predicting false-positive results; this is much larger than the previous two datasets and perhaps suggests that the genomes in this group need further annotation work. Only two putative gene fusion events were mis-predicted by fdfBLAST because the phylogenetic analysis did not demonstrate differentially distributed protein domain architecture.

After phylogenetic analysis six gene fusion datasets showed no resolution and therefore these six candidate SDCs were discarded due to unresolved paralogous genes and/or the effects of LBA (H. Philippe, 2000; Herve Philippe, et al., 2005) artefacts which prevented the polarisation of the gene fusion or fission event. Three gene fusion candidates did not produce a tree topology from the gene-by-gene phylogeny pipeline 'Darren's Orchard' because taxon and/or character sampling was so low that we failed to construct a meaningful tree and so were also discarded.

Eight gene fusion characters were confirmed by phylogenetic analysis and were present only in *Gallus gallus* and one was confirmed and present in *Homo sapiens* of the genomes sampled. Four reversion events were also predicted and subsequently confirmed. None of these differentially distributed characters were able to be used as SDCs because given the current genome sampling these putative SDCs were only

present in a single genome and so they were not considered phylogenetically informative.

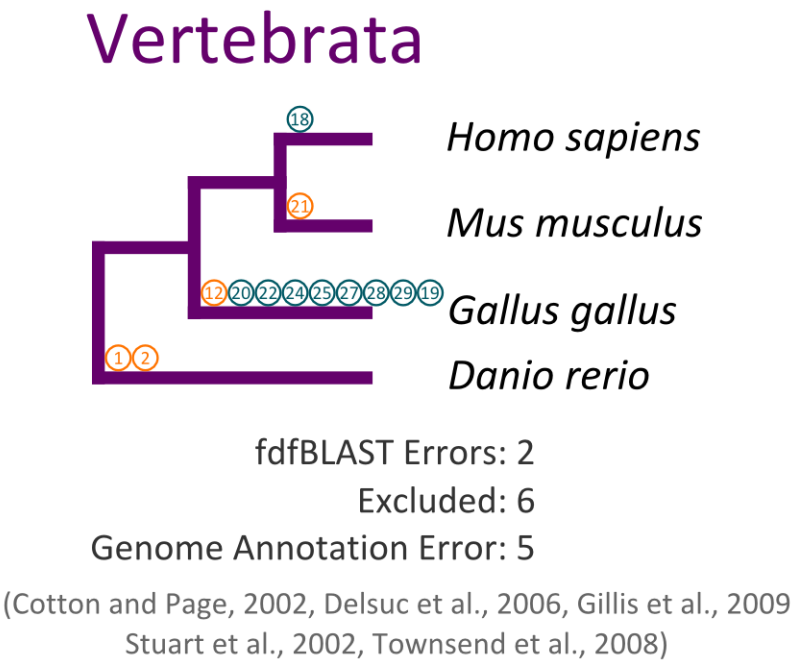


Figure 6:9 - A consensus topology for the Vertebrata indicating the branching order for the four taxa included in this analysis. The blue circles represent predicted gene fusion events and the orange reversion events. The numbers within the circles correspond to the accessions and tree topologies in Section 10.

6.3.4 Discicristata

The Discicristata dataset returned twelve putative differentially distributed fusion and fission events (Figure 6:10), there were no fdfBLAST mis-predictions, however one tree was excluded due to a very long branch which caused the tree to have low resolution preventing the polarisation of the gene fusion or fission event. Five putative fusions/fissions were removed from the predicted list due to genome annotation errors present in the *T. cruzi* genome database where the genes were mis-predicted as

two separate domains. These were checked by going to the respective genome database website and using the genome browser to look at the annotation of the scaffold sequences. If the two split domains/ORFs of a candidate gene fusion were next to each other on the scaffold and appeared in the same direction then they were discarded as potentially miss-annotated separate genes when they were potentially a composite fusion gene, otherwise they were accepted and used in the next stage of analysis. Nevertheless, three gene fusion events and four reversions were accepted following phylogenetic inference. Interestingly, three of these showed evidence of being SDCs for the monophyly of the kinetoplastids, including *Leishmania major*.

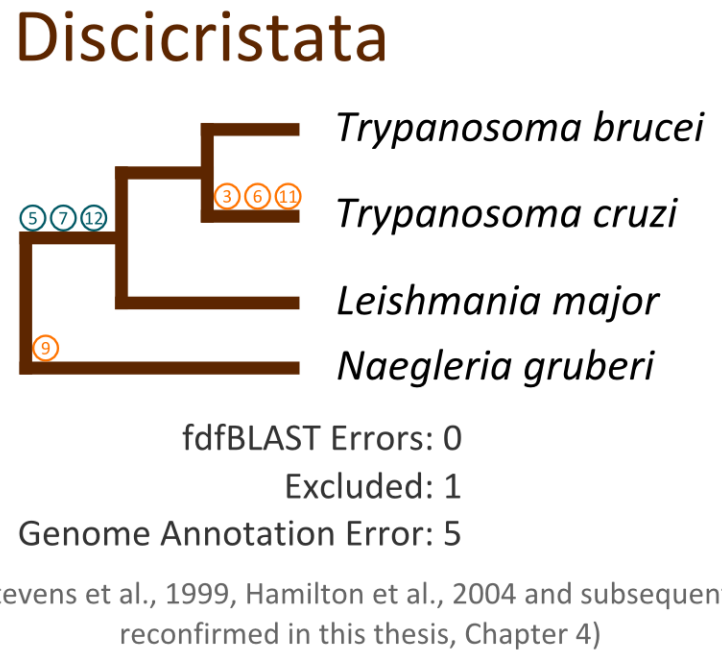


Figure 6:10 - A consensus topology for the Discicristata indicating the branching order for the four taxa included in this analysis. The blue circles represent predicted gene fusion events and the orange reversion events. The numbers within the circles correspond to the accessions and tree topologies in Section 10.

6.3.4.1 Phylogenetically Informative Datasets

Fusion 5 - Figure 6:11 and Section 11 (not shown in the text here as it is a large phylogeny spanning multiple pages) represent a gene fusion event between the two ORFs/domains of GAF (Figure 6:11) and TIP41 Section 11 where each tree topology is based on a masked alignment and phylogenetic analysis of a sequence representing one of the domains, respectively. In Figure 6:11 the gene fusion event appears to be unfused in *Naegleria gruberi* suggesting that it may be a Kinetoplastida specific synapomorphy; however, *Naegleria* branches elsewhere in the tree. Nonetheless the monophyly of the kinetoplastid fusion genes is strongly supported suggesting this is a reliable SDC for the kinetoplastids.

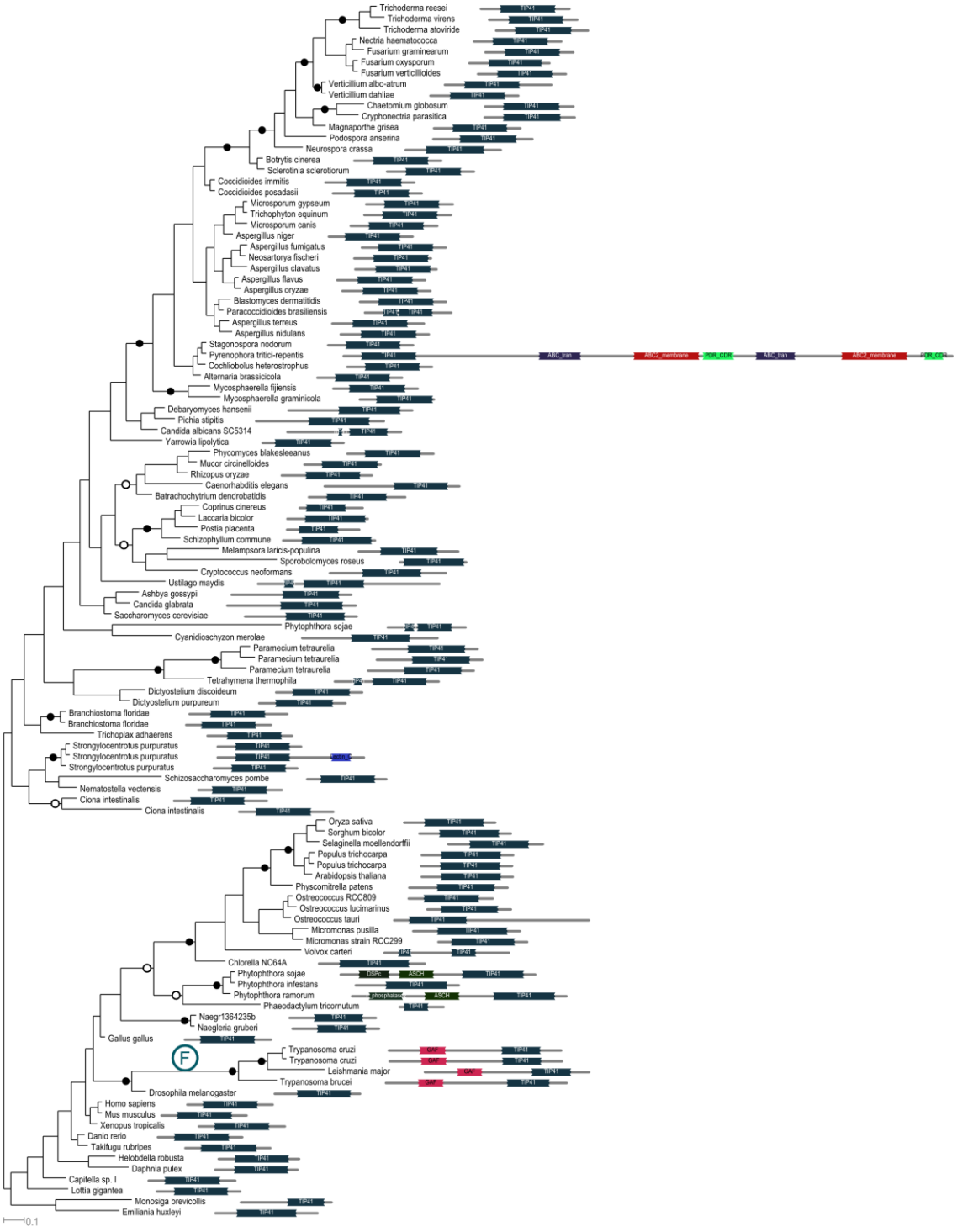


Figure 6:11 - A tree topology representing the TIP41 domain present in Discicristata Fusion 5. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 are represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML.

Fusion 7 - Figure 6:12 and Figure 6:13 depict a gene fusion event that occurs between the two ORFs/domains of Alg14 (Figure 6:12) and Glyco_tran_28_C (Figure 6:13), where each tree topology is based on a masked alignment and phylogenetic analysis of a sequence representing one of the domains, respectively. In both phylogenies the kinetoplastid gene fusions are monophyletic with strong bootstrap support, suggesting that this is a SDC for the holophyly of the Kinetoplastida (given current genome sampling). I note that both trees are weakly supported elsewhere; nonetheless the gene phylogenies demonstrate two additional gene fusion events of Alg14 and Glyco_tran_28_C in the Amoebozoa *Entamoeba* and *Dictyostelium.* Amoebozoa are distant relatives of the kinetoplastids (Bapteste & Philippe, 2002; Hampl, et al., 2009). These gene fusions represent domain architectures in the opposite orientation. Taken together, this suggests that a gene fusion between Alg14 and Glyco_tran_28_C has occurred twice, suggesting that the combination of these two protein domains may be the product of convergent evolution. However, it is currently impossible to rule out other explanations, such as hidden paralogy, especially as the resolution across the tree is weak.

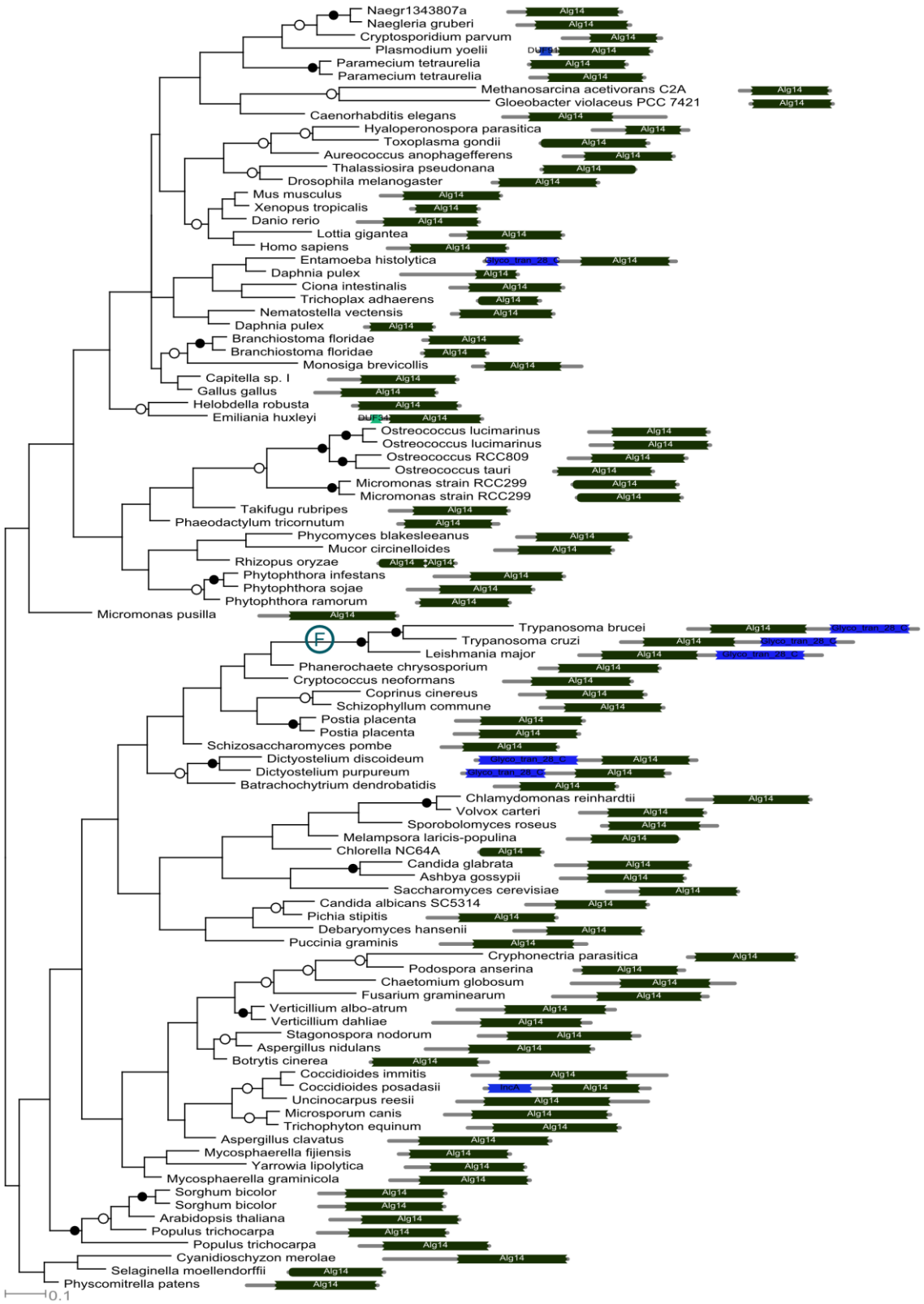


Figure 6:12 - A tree topology representing the Alg_14 domain present in Discicristata Fusion 7. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 are represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML.

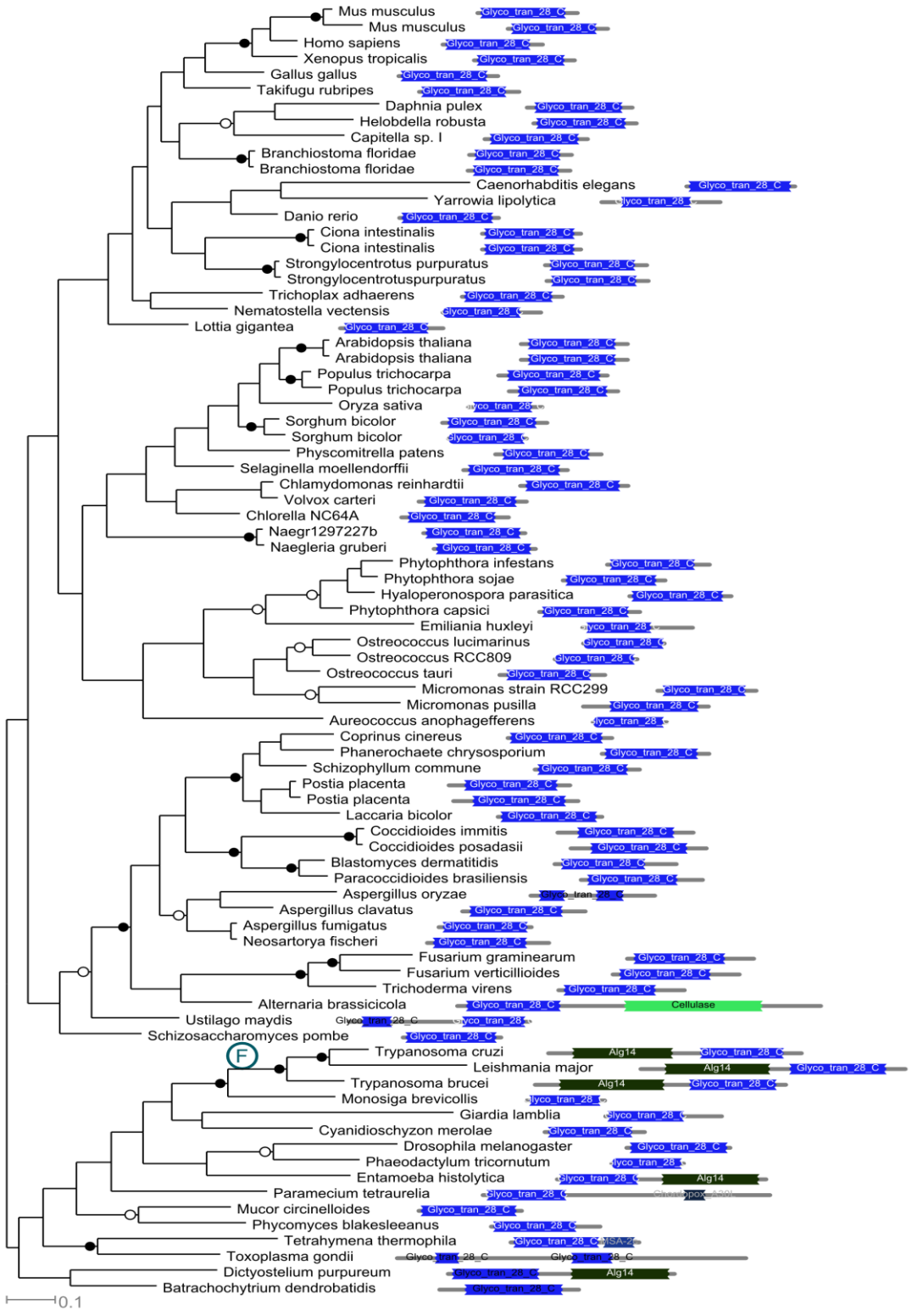


Figure 6:13- A tree topology representing the Glyco_tran_28_C domain present in discicristata fusion 7. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 are represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML.

Fusion 12 - Figure 6:14 and Section 11 (not shown in the text here as it is a large phylogeny spanning multiple pages) also show a gene fusion event between the two domains of Put_Phosphatase and DUF89 which is present in the three kinetoplastid genomes, suggesting that the gene fusion event occurred in the last common ancestor of these three parasites. This relationship is strongly supported in all the bootstrap analyses. We also note *Naegleria gruberi* branches separately in the phylogeny, which is inconsistent with current understanding of the branching relationships of the eukaryotes (Hampl, et al., 2009).

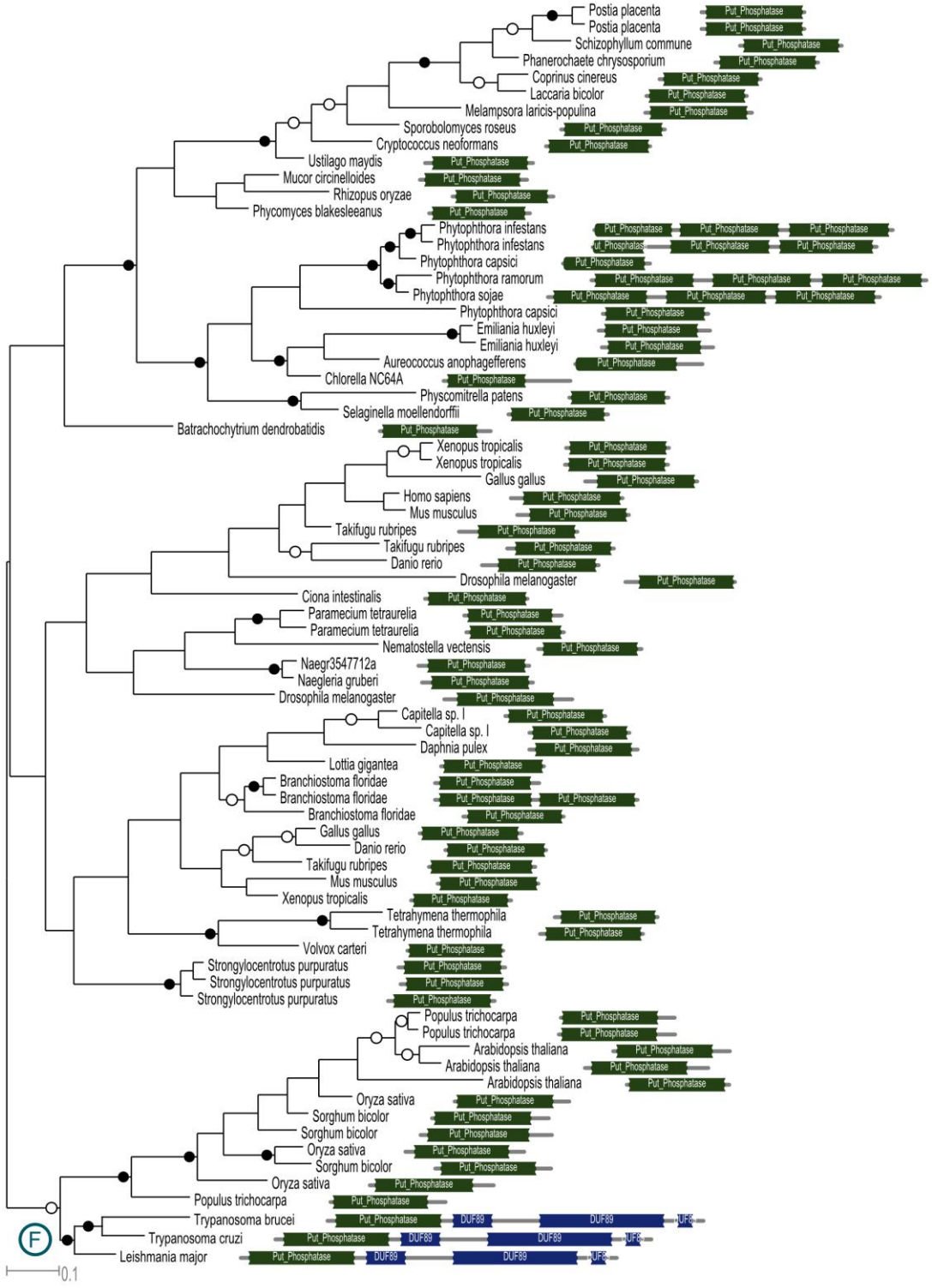


Figure 6:14 - A tree topology representing the Put_phosphotase domain present in discicristata fusion 12 and showing a gene fusion common to the Kinetoplastida. The tree topology is based on the results of a MrBayes analysis, where Bayesian inference values and both fast-ML bootstrap statistics are greater than 80 a filled black circle is shown on the corresponding branch, values greater than 60 are represented by an empty circle. Other values are not shown, except where major fusion or fission events occur, which appear in the order MrBayes/PhyML/RAxML.

6.3.5 Deuterostomia

The deuterostomia dataset returned the largest number of fdfBLAST gene fusion candidates, with 54 in total. The number of mis-predictions was also relatively higher with 19 altogether. Six trees were excluded due to paralogue and resolution problems in the subsequent phylogenies, so that we could not resolve the nature of the fusion or fission event (mostly due to the effects of LBA artefacts and/or lack of resolution within the tree topology). Four predicted fusions were removed from the predicted list due to genome annotation errors present across the genomes where domains/ORFs were mis-predicted as two discrete domains when there genome location could not rule out that they were a single composite fusion gene. As before the genome browsers for the respected genomes were queried and the position of each ORF identified on the scaffold. If they appeared next to each other they were discarded because they were suspected to be falsely annotated as separate genes.

Six candidate gene fusions did not produce a tree with 'Darren's Orchard' automatic tree building pipeline because the taxon and site sampling methods failed to accurately recover enough data to generate a phylogeny, suggesting the genes analysed are too divergent to be informative. In total, these seven candidate gene fusions were excluded. Overall fdfBLAST predicted four validated gene fusions events and 19 validated gene fission events for the Deuterostomia dataset.

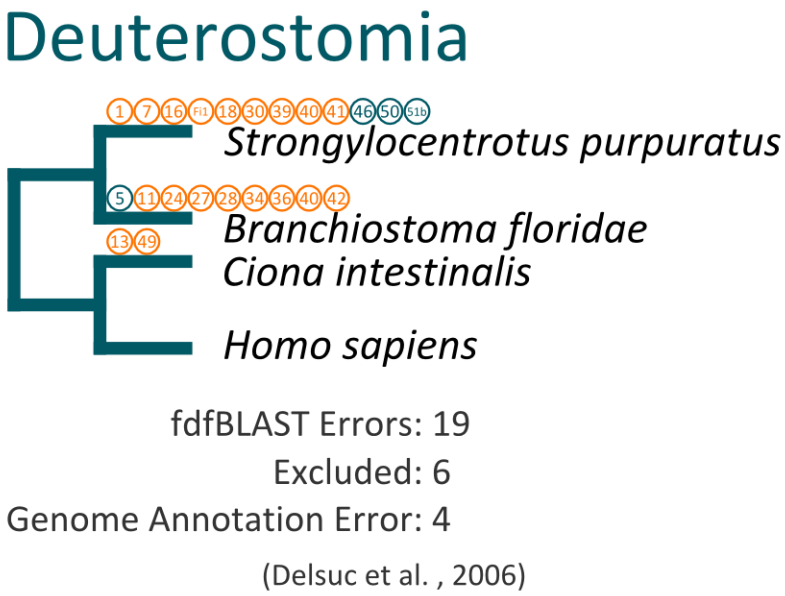


Figure 6:15 - A consensus topology for the Deuterostomia indicating the branching order for the four taxa included in this analysis. The blue circles represent predicted gene fusion events and the orange reversion events. The numbers within the circles correspond to the accessions and tree topologies in Section 10.

6.4 Discussion

6.4.1 Comparisons Between the 4-way Datasets

Figure 6:16 is an updated version of the first figure in this chapter, Figure 6:1, but this time it includes the validated gene fusion and fission events predicted by fdfBLAST. The figure also includes the numbers of putative gene fusions that were false-positives (fdfBLAST errors), the number of putative fusions excluded due to the effects of LBA artefacts and paralogue confusion (all excluded) and the number of putative gene fusions that were removed due to annotation problems contained within each individual genome project.

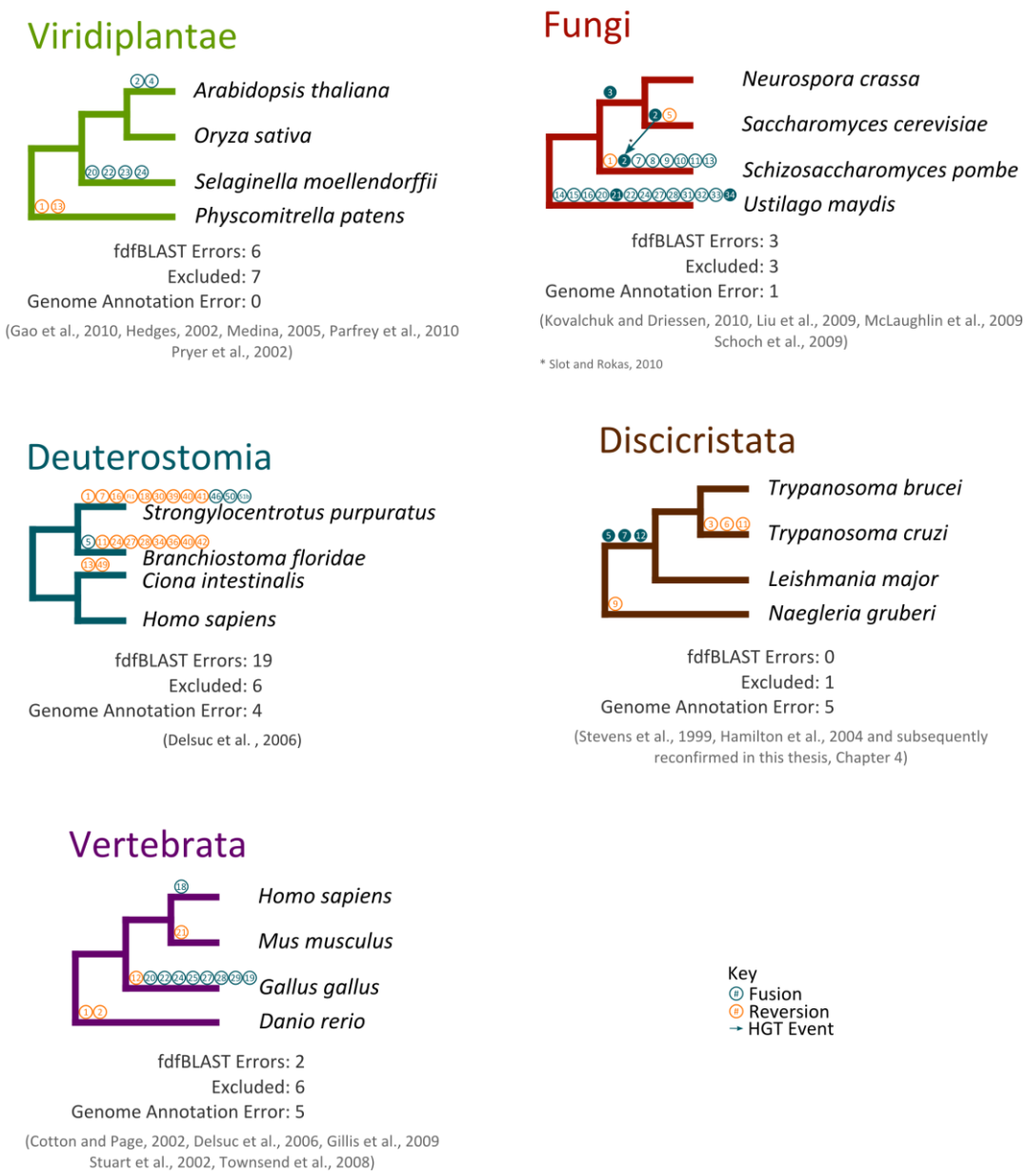


Figure 6:16 - An updated version of Figure 6:1 showing all the rates of fusion and fission (reversion) events predicted by fdfBLAST for the five 4-way analyses. Phylogenetically informative sites are represented by a circle with a solid coloured background. Note the identification of a HGT event in the Fungi dataset which has been recently independently published (Slot and Rokas, 2010). Fusion and reversion numbers correspond to the relevant phylogenies which can be found in Section 10

6.4.2 Comparative Rates of Fusion and Fission

It is interesting to note the prevalence of fusion and reversion rates between the individual datasets; which is visually demonstrated in Figure 6:17. It is immediately noticeable that the dataset representing the fungi contains the most number of fusion events compared to any other group. The findings of Durrens et al. (Durrens, et al., 2008) showed that gene fusion events occurred more often than gene fission events, although they stressed that they were not as prevalent as those that had been found in other studies (Kummerfeld & Teichmann, 2005; Snel, et al., 2000). Our fdfBLAST analyses searched only four genomes compared to their twelve, which may indicate the reasons for this disparity, as with a further eight genomes it stands to reason that fusion and fission events may not be conserved across all the taxa present in both studies. A similar analysis could be completed with fdfBLAST with the inclusion of more taxa to draw a direct comparison.

The relative rates of reversion (fission) events are low in four out of five of the datasets except the deuterostomia where it is much higher than any other group. This overall trend is contradictory to previous statements that fission events occur more often (Nakamura, et al., 2006; Snel, et al., 2000) and reflects the results of the two-way plant genome analysis and comparisons from Chapter 4. Moreover, it gives greater credence to the earlier assertion that fission events are evolutionary complex genetic events, requiring the insertion of a stop codon, promoter, and a start codon, whilst maintaining the correct codon reading frame and are therefore 'less parsimonious' (Stechmann & Cavalier-Smith, 2002). Therefore, many authors have assumed that fission events are less likely to occur within eukaryotic genomes (Stechmann & Cavalier-Smith, 2002, 2003), our results in general support this assumption, although

we would advise caution and appropriate phylogenomic analyses to accompany any claims based on gene fusions as evolutionary synapomorphies.

Furthermore, all our fusions/fission events, when possible, have been further investigated using taxon rich phylogenetic analysis. This means that these fusion analyses, unlike previous work (Durrens, et al., 2008; Kummerfeld & Teichmann, 2005; Nakamura, et al., 2006; Snel, et al., 2000), have all been individually tested for cases of secondary reversion i.e. fission events. Even with this extra caveat our data still supports a high fusion to low fission ratio.

Fusions occur more often in general, although in some datasets gene fission events have taken precedence (i.e. Deuterostomia and Discicristata). This, therefore, highlights the importance of comparing multiple groups, and multiple genomes of taxa within those groups, and not just forming a pattern based on single and select datasets.

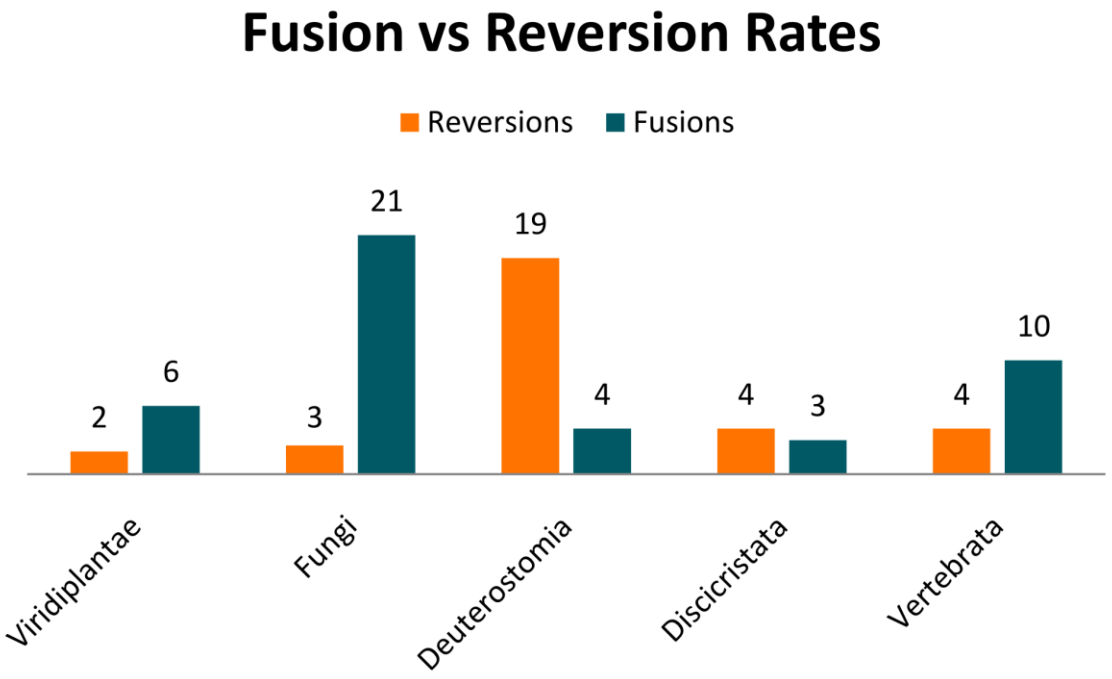


Figure 6:17 - Fusion vs. Reversion rates between the five calibration datasets. Note the high occurrence of fusion events in the Fungi but the relatively low occurrence of reversions across most of the other datasets apart from the Deuterostomia.

It is also interesting to compare the results from the previous chapter with those of the plants represented in this chapter. Figure 6:18 attempts this by showing the Nakamura et al. results, the revised Nakamura et al. results, fdfBLAST's predictions for the two plant genomes and fdFBLAST's predictions for the 4-way plant genome analysis. As we can see, in every case, except the original dataset, gene fusion events compared to gene fission events occur relatively more often. Once again, this pattern demonstrates the expected prevalence for these events within genomes under the most parsimonious explanation. That is to say, the occurrence of gene fusion events is expected to be more frequent than gene fission events, as grouping of similar biochemical functions together is more likely than the inclusion of several gene components during a gene fission event.

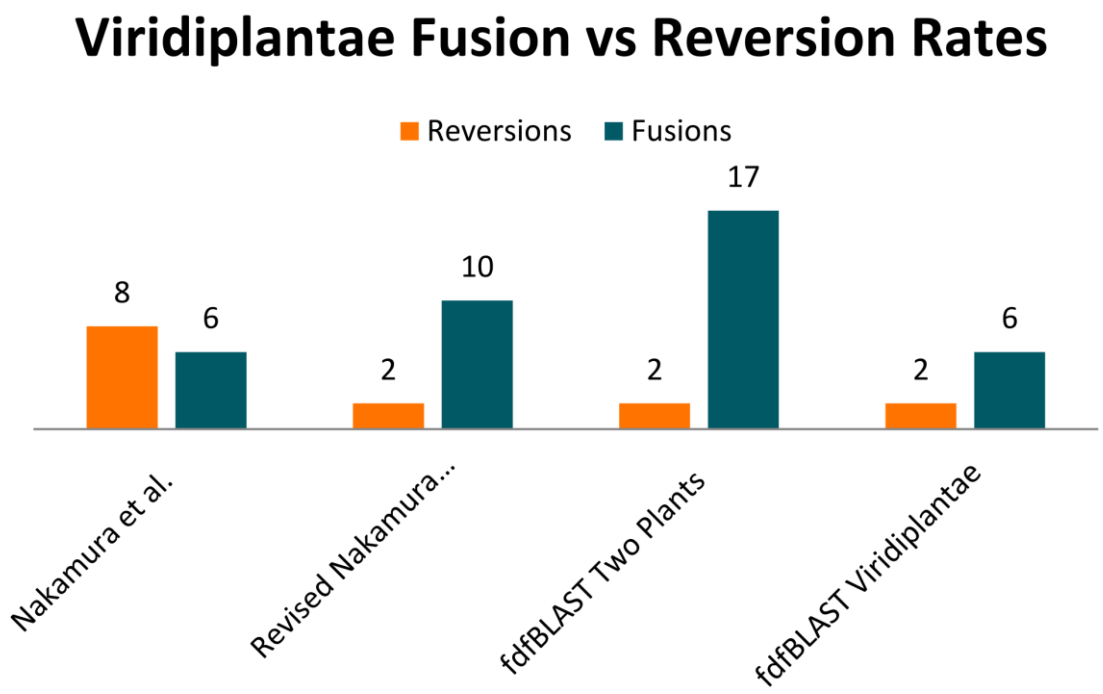


Figure 6:18 - Fusion vs. Fission rates between all the plant genome datasets tested in this thesis and those from Nakamura et al. (2006). Note the contrasting results between the two fdfBLAST analyses coupled with the revised Nakamura dataset compared to those of the original analysis. Demonstrably gene fusions events occur more often than gene fission events within the Viridiplantae.

6.5 Conclusion

The primary aim of this chapter was to test the fdfBLAST program on several sets of data as a form of calibration, in order to understand the capabilities of the methods employed by my program fdfBLAST. It is important to understand the abilities of the program to get a sense of how many false positive and true-positives are predicted and at what rate gene fusion and fission events are observed and whether these corroborate with previous findings. Earlier in this thesis, Chapter 4, fdfBLAST was used in a comparison test with the findings of Nakamura et al (2006), although in the process of this it was found that their results were not a true reflection of the data at present (due to re-annotation of the genome over the last 4 years), and a re-evaluation of their data and results found that gene fusion events occurred more often

than gene fission events in contradiction to their central conclusion. In their initial analysis Nakamura et al., predicted 8 gene fission events compared to six fusion events, however, using the re-evaluated dataset the numbers showed 10 gene fusion events and only two gene fission events.

This was also similar to the findings of fdfBLAST's predictions for the two plant genomes, two putative gene fission events and 17 putative gene fusion events. Part of the interest in extending the analysis to new groups and by expanding the number of taxa involved was to see if this 'pattern' was continued across the tree of life. Indeed, it was the findings of this thesis that the occurrence of gene fusion events was generally greater than those of gene fission events in nearly all the groups evaluated, see Figure 6:17. The deuterostomia were an exception to this pattern, which suggests that the relative rates of gene fusion and fission events show a heterogeneous pattern across phylogenetic groups of the eukaryotic tree of life. This suggests the need to survey further groups along with the expansion of the taxa involved in order to confirm that fusions occur at a higher rate than gene fission events.

This could be established in several ways: further sets of 4-way comparisons encompassing more groups, the replacement of taxa within the already selected groups in order to complete additional comparisons, increasing the number of taxa in the currently selected groups, or even surveying several genomes from across the eukaryotic tree of life in one analysis. The latter would prove more interesting on the grounds that it would further test the effectiveness of fdfBLAST under a different situation, the former suggestions are more of the same and whilst may provide

phylogenomically interesting results do not stretch (in order to test) the usefulness of my program, fdfBLAST.

A further aim of this chapter was the anticipation of the identification of shared derived characters in the form of gene fusion and fission events that could be used to help polarise contentious evolutionary relationships; however, fdfBLAST only identified six in total covering just two of the five groups. Three were synapomorphies for the kinetoplastids (Figure 6:11 through Figure 6:14). The use of these gene fusion characteristics could therefore prove useful for polarising evolutionary relationships in and around the kinetoplastids. A further three (Figure 6:4 through Figure 6:8) were putative synapomorphies for relationships within the fungi. Sources of artefact were still to be found in some of the phylogenies built from the gene fusion and fission event predictions. In one tree, Fungi putative gene fusion 2, a horizontal gene transfer (HGT) event was apparent, and had been recently independently published by Slot and Rokas (Slot & Rokas, 2010). Further trees lacked any resolution in branching order which may point towards independent mechanisms of evolution or massive paralogue duplication events. While other datasets provided evidence for gene fission events. This result therefore illustrates the importance of combining gene fusion analyses with appropriate phylogenomic analyses.

7 Discussion

The main aim of this thesis was to develop a methodological approach for the automatic discovery of differentially distributed putative gene fusion and fission events that are shared between a set of predicted proteomes. The motivation for this aim was in order to help further resolve and polarise phylogenetic relationships (and so tree topologies) that are not well established and where they form trifurcation events, by the inclusion of synapomorphic characters. Trifurcations (three branched unresolved tree topologies) can occur when taxon sampling of a particular group is relatively low (which is most likely due to the number of available sequenced genomes within that group) or they may occur 'naturally' as with the root of the tree of life and relative branching order between the Eukarya, Archaea and Bacteria (Forterre & Philippe, 1999; Gribaldo, et al., 2010; Koonin, 2010; Hervé Philippe & Forterre, 1999; Maria C. Rivera & Lake, 2004). Synapomorphies or shared derived characters (SDCs) can then be used to polarise the relative branching order between the taxa being studied given that proper phylogenetic reconstruction methods are used to confirm that the SDC is reliable as a synapomorphy.

The full methodological approach can be found in Chapter 3, but briefly a set of Perl scripts are used to query a set of genomes with the help of a local version of the BLAST tool and differentially distributed hits are tested for their possibility of representing a differentially distributed gene fusion event. They are tested by a series of ranking and sorting of the best hits and by the comparison of ORFs to discrete functional protein domains from PFAM (Bateman, et al., 2004; Finn et al., 2010) and/or CDD (Marchler-Bauer, et al., 2005). The resulting program, called fdfBLAST, necessitated testing of its

ability to produce sets of identified putative gene fusion and fission events. It was therefore favourable to base a comparison on a previously identified set of gene fusion events; indeed, this was achieved with the inclusion of a dataset from Nakamura et al. (2006) (Nakamura, et al., 2006).

Nakamura et al (2006) investigated the number of gene fusion and fission events between the predicted proteomes of *Arabidopsis thaliana* and *Oryza sativa* using an extrapolation method to identify them. However, their dataset needed re-evaluation before comparison to fdFBLAST's results could take place, due to both genomes having undergone subsequent re-annotation over the last four years - this dramatically reduced the number of putative gene fusion events (60 to 12) claimed by Nakamura et al., and also reversed their main findings (that fissions were greater in number than fusions). fdfBLAST predicted 266 putative gene fusions which was reduced to 19 once the putatively split-ORFs were compared with the PFAM discrete functional domains database. Of these, just three were shared between the fdfBLAST and Nakamura et al. (2006) manual analyses datasets.

In order to further test the method for finding differentially distributed gene fusion events (with the program fdfBLAST) I decided to compile a group of five datasets that each presented an independently resolved and robustly inferred phylogenetic tree topology. Among the eukaryotic genomes only the plants, animals and fungi are relatively well sampled, with 3+ genomes. We therefore specifically targeted four sets of genomes across these three groups. Three were new datasets (comprising of four taxa from each of the Deuterostomia, Fungi and Vertebrata), and one was an extension of the first testing dataset (including two new plant taxa). However, these

datasets were generally biased toward multi-cellular life forms. Therefore, we sought a fifth set of genomes for comparison, specifically comprising of single celled organisms select from the group Discicristata. To use this dataset it was important to unambiguously confirm the branching order among this group. We therefore used a complex set of systematic phylogenetic methods designed specifically to investigate low taxon number with complete genome sampling. These analyses confirmed that the *Trypanosoma* are monophyletic resolving 11 years of phylogenetic debate (Haag, O'HUigin, & Overath, 1998; Hamilton, et al., 2004; A. Hughes & H. Piontkivska, 2003; A. L. Hughes & H. Piontkivska, 2003; Lukeš, et al., 1997; Maslov, et al., 1996; Moreira, et al., 2004; Piontkivska & Hughes, 2005; A. G. Simpson, Y. Inagaki, et al., 2006; A. G. Simpson, et al., 2002; A. G. Simpson, Stevens, & Lukes, 2006; A. G. B. Simpson, et al., 2004; A. G. B. Simpson, et al., 2006; Stevens, 2008; Stevens & Gibson, 1999; Stevens, et al., 1999; Stevens, et al., 2001).

We then used these five sets of four genome analyses to re-test the fdfBLAST program and investigate the relative rates of gene fusion and fission events in a phylogenetic context. The literature cited in my introduction indicates contradictory results for the prediction of gene fusion and fission events across the tree of life. With some analyses suggesting that fission events occur at a relatively high rate (Snel, et al., 2000) whilst other analysis suggests that fusion events occur at a low rate (Kummerfeld & Teichmann, 2005). These studies were completed by the analysis of few taxa and by different gene fusion prediction methods. My program fdfBLAST allowed for both of these problems to be overcome. We were able to sample taxa from across the tree of life in five datasets whereby each dataset was tested via the same methodology. This allowed us to build a comparable and contrastable set of data. These data indicated

that although the rate of gene fusion and fission events across the tree of life are not homogenous they do trend towards the presence of more fusion events and fewer fission events. This result has direct implications for our use of gene fusions as evolutionary informative synapomorphies, because the identification of a lower rate of reversion suggests that these characters are less likely to be homoplasious. This result is tentative because our analysis is only based on five comparisons of four genomes. It is important that future work focus on retesting these conclusions for larger datasets.

During the course of our analyses we used the fdfBLAST program to identify several gene fusion synapomorphies for understanding deeper-level phylogenetic relationships (see Sections 6.3.2.1 and 6.3.4.1)

7.1 Future Directions

7.1.1 fdfBLAST Applications

The program fdfBLAST has been shown to be a valid and useful tool in the automated discovery of shared derived gene fusion characters between many sets of predicted proteomes. Subsequently these identified gene fusion and fission events have been proven useful in the resolving of phylogenetic relationships. However, there were many gene fusion events that were predicted fdfBLAST that only appeared on a single branch for each of the analyses. They confer little resolution directly to the phylogenetic topologies within each dataset; however, we should not limit ourselves to the taxa present in these already established datasets. For example, further analysis could be carried out on the extended Viridiplantae analysis, the two *Arabidopsis thaliana* specific fusion events (Figure 6:2), which could be further tested by expanding the clade between *Arabidopsis thaliana* and *Oryza sativa* by adding the *Populus*

trichocarpa and *Sorghum bicolor* genomes and also by, potentially, removing *Selaginella moellendorffii* and *Physcomitrella patens* (obviously, the more genomes you test then the longer the analysis may be). Moreover, as new genomes are being sequenced, annotated and released at an ever increasing rate (over 1000 in 2009 compared with only around 200 in 2004) (Liolios et al., 2010) they too can be added into new analyses for gene fusion events that currently do not convey any phylogenetic importance. However, it is important to remember that even though a genome sequence is complete and released to the public it may not be fully and accurately annotated. Nevertheless, it would be interesting to explore the use of fdfBLAST as another tool for the prediction of areas of a genome sequence that are badly annotated compared to another genome sequence that is relatively well annotated. This could establish a baseline for genome sequencing quality prior to fdfBLAST analyses.

7.1.2 fdfBLAST Program Design

Currently the method for prediction of putative gene fusion relies on a search method that can only predict two-domain architectures. The program, therefore, cannot distinguish between two separate bi-fusions and a ≥Tri-fusion. This is a prime target for improvement as it would be extremely useful to be able to investigate multiple gene fusion events within one gene family. As yet, we have not formulated a strategy to search for multiple domain gene fusions, other than the omission of a domain between two identified domains that occur as an artefact of the current method

Minor tweaks can also be applied to the differentially distributed and reciprocal search technique stages; currently if a gene in genome A has numerous hits to genes in other

genomes, it can take a very long time to search for each one. Even at a very high e-value threshold, there can be up to 500 (seemingly BLAST's own cut-off value) hits, therefore, a user-editable cut-off may be useful tactic to restrict the number of extraneous searches completed and so reduce the time spent searching.

I also understand that the reliance on PFAM (Bateman, et al., 2004; Finn, et al., 2010) or CDD (Marchler-Bauer, et al., 2005) conserved discrete domain predictions may be a source of bias in the results for the putative gene fusion predictions. Not only because the databases are already biased as they are databases for well characterised and conserved domains but by discounting many of fdfBLAST's predictions, simply as they do no not appear to contain any previously predicted conserved domains, therefore we may be missing several important SDCs. Therefore, in order to help with this problem the inclusion of further sources of domain predictions, perhaps using the PFAM-B database (Sonnhammer, Eddy, & Durbin, 1997), NCBI COGS database (Tatusov et al., 2003; Tatusov, et al., 1997), SMART (Simple Modular Architecture Research Tool) (Copley, Schultz, Ponting, & Bork, 1999; Letunic, Doerks, & Bork, 2009; Schultz, Milpetz, Bork, & Ponting, 1998) or any other database containing putative gene architecture could be adapted to work with fdfBLAST. In addition, it may be worth generating HMMs (Durbin, et al., 1998; Krogh, et al., 1994) for all differential distributed blast hits with asymmetric sequence coverage, which therefore suggests a gene/domain fusion. This would involved the recovery of additional putative homologues using BLAST, alignment of the data and the generation of HMMs, subsequently the results would represent *de facto* protein domains, given a wide taxon distribution and conserved HMM recovery. Of course, the user of fdfBLAST is most

welcome to compute phylogenies for all of the putative differentially distributed gene fusions that do not show currently recognised discrete domains.

As the first stage of fdfBLAST relies on serial genome-to-genome BLASTp analyses, this is very inefficient for repeat analysis and therefore could be replaced by a pre-computed database of genome-to genome blast hits. For example, given the list of taxa from the extended Viridiplantae analysis, if I wanted to complete another run with three of the same taxa but substitute one of them for a new genome, it makes no sense to have to run the BLASTp analyses again between the three that have already been completed. Instead, only four BLASTp analyses need be computed (the new genome against the three old). fdfBLAST is already modular in its approach, each stage can be run independently from the other (although each stage implies that previous stage has been completed); however, with the inclusion of a database of already computed whole genome BLASTp analyses this initial stage could be significantly easier to achieve.

Finally, this thesis reports the development and testing of a new tool to identify differentially distributed gene fusion events. Our tests demonstrate that the program works and can be used to find phylogenetically informative gene fusion characters. The program scripts will be made publicly available, upon publishing (and most likely under GPL v3) and, for now, the code is included as appendix 9 for further testing and evolutionary analysis.

8 Appendix: A List of the 795 Taxa included in 'Darren's Orchard' Automatic Phylogeny Pipeline

Acaryochloris marina MBIC11017
Acholeplasma laidlawii PG-8A
Acidiphilium cryptum JF-5
Acidobacteria bacterium Ellin345
Acidothermus cellulolyticus 11B
Acidovorax avenae subsp. citrulli AAC00-1
Acidovorax sp. JS42
Acinetobacter baumannii
Acinetobacter baumannii ATCC 17978
Acinetobacter sp. ADP1
Actinobacillus pleuropneumoniae L20
Actinobacillus pleuropneumoniae serovar 3 str. JL03
Actinobacillus succinogenes 130Z
Aeromonas hydrophila subsp. hydrophila ATCC 7966
Aeromonas salmonicida subsp. salmonicida A449
Aeropyrum pernix K1
Agrobacterium tumefaciens str. C58
Alcanivorax borkumensis SK2
Alkalilimnicola ehrlichei MLHE-1
Alkaliphilus metalliredigens QYMF
Alkaliphilus oremlandii OhILAs
Alternaria brassicicola
Anabaena variabilis ATCC 29413
Anaeromyxobacter dehalogenans 2CP-C
Anaeromyxobacter sp. Fw109-5
Anaplasma marginale str. St. Maries
Anaplasma phagocytophilum HZ
Aquifex aeolicus VF5
Arabidopsis thaliana
Archaeoglobus fulgidus DSM 4304
Arcobacter butzleri RM4018
Arthrobacter aurescens TC1
Arthrobacter sp. FB24
Ashbya gossypii
Aspergillus clavatus
Aspergillus flavus
Aspergillus fumigatus
Aspergillus nidulans
Aspergillus niger
Aspergillus oryzae
Aspergillus terreus
Aster yellows witches-broom phytoplasma AYWB
Aureococcus anophagefferens
Azoarcus sp. BH72
Azoarcus sp. EbN1
Azorhizobium caulinodans ORS 571
Bacillus amyloliquefaciens FZB42
Bacillus anthracis str. Ames
Bacillus anthracis str. Ames Ancestor
Bacillus anthracis str. Sterne
Bacillus cereus ATCC 10987
Bacillus cereus ATCC 14579
Bacillus cereus E33L
Bacillus cereus subsp. cytotoxis NVH 391-98
Bacillus clausii KSM-K16
Bacillus halodurans C-125
Bacillus licheniformis ATCC 14580
Bacillus pumilus SAFR-032
Bacillus subtilis subsp. subtilis str. 168
Bacillus thuringiensis serovar konkukian str. 97-27
Bacillus thuringiensis str. Al Hakam
Bacillus weihenstephanensis KBAB4
Bacteroides fragilis NCTC 9343
Bacteroides fragilis YCH46
Bacteroides thetaiotaomicron VPI-5482
Bacteroides vulgatus ATCC 8482
Bartonella bacilliformis KC583
Bartonella henselae str. Houston-1
Bartonella quintana str. Toulouse
Bartonella tribocorum CIP 105476
Batrachochytrium dendrobatidis
Baumannia cicadellinicola str. Hc
Bdellovibrio bacteriovorus HD100
Beijerinckia indica subsp. indica ATCC 9039
Bifidobacterium adolescentis ATCC 15703
Bifidobacterium longum NCC2705
Blastomyces dermatitidis
Bordetella avium 197N
Bordetella bronchiseptica RB50
Bordetella parapertussis 12822
Bordetella pertussis Tohama I
Bordetella petrii DSM 12804
Borrelia afzelii PKo
Borrelia burgdorferi B31
Borrelia garinii PBi
Botrytis cinerea
Bradyrhizobium japonicum USDA 110
Bradyrhizobium sp. BTAi1
Bradyrhizobium sp. ORS278
Brucella abortus biovar 1 str. 9-941
Brucella abortus S19
Brucella canis ATCC 23365
Brucella melitensis 16M
Brucella melitensis biovar Abortus 2308
Brucella ovis ATCC 25840

Brucella suis 1330
Brucella suis ATCC 23445
Buchnera aphidicola str. APS
Buchnera aphidicola str. Bp
Buchnera aphidicola str. Cc
Buchnera aphidicola str. Sg
Burkholderia ambifaria AMMD
Burkholderia ambifaria MC40-6
Burkholderia cenocepacia AU 1054
Burkholderia cenocepacia HI2424
Burkholderia cenocepacia MC0-3
Burkholderia mallei ATCC 23344
Burkholderia mallei NCTC 10229
Burkholderia mallei NCTC 10247
Burkholderia mallei SAVP1
Burkholderia multivorans ATCC 17616
Burkholderia pseudomallei 1106a
Burkholderia pseudomallei 1710b
Burkholderia pseudomallei 668
Burkholderia pseudomallei K96243
Burkholderia sp. 383
Burkholderia thailandensis E264
Burkholderia vietnamiensis G4
Burkholderia xenovorans LB400
Caenorhabditis elegans
Caldicellulosiruptor saccharolyticus DSM 8903
Caldivirga maquilingensis IC-167
Campylobacter concisus 13826
Campylobacter curvus 525.92
Campylobacter fetus subsp. fetus 82-40
Campylobacter hominis ATCC BAA-381
Campylobacter jejuni RM1221
Campylobacter jejuni subsp. doylei 269.97
Campylobacter jejuni subsp. jejuni 81-176
Campylobacter jejuni subsp. jejuni 81116
Campylobacter jejuni subsp. jejuni NCTC 11168
Candida albicans SC5314
Candida glabrata
Candidatus Blochmannia floridanus
Candidatus Blochmannia pennsylvanicus str. BPEN
Candidatus Carsonella ruddii PV
Candidatus Desulforudis audaxviator MP104C
Candidatus Korarchaeum cryptofilum OPF8
Candidatus Methanoregula boonei 6A8
Candidatus Pelagibacter ubique HTCC1062
Candidatus Protochlamydia amoebophila UWE25
Candidatus Ruthia magnifica str. Cm
Candidatus Sulcia muelleri GWSS
Candidatus Vesicomyosocius okutanii HA
Capitella sp. I
Carboxydothermus hydrogenoformans Z-2901
Caulobacter crescentus CB15
Caulobacter sp. K31
Chaetomium globosum
Chlamydia muridarum Nigg
Chlamydia trachomatis 434 Bu
Chlamydia trachomatis A HAR-13
Chlamydia trachomatis D UW-3 CX
Chlamydia trachomatis L2b UCH-1 proctitis
Chlamydomonas reinhardtii
Chlamydophila abortus S26 3
Chlamydophila caviae GPIC
Chlamydophila felis Fe C-56
Chlamydophila pneumoniae AR39
Chlamydophila pneumoniae CWL029
Chlamydophila pneumoniae J138
Chlamydophila pneumoniae TW-183
Chlorella NC64A
Chlorobium chlorochromatii CaD3
Chlorobium phaeobacteroides DSM 266
Chlorobium tepidum TLS
Chloroflexus aurantiacus J-10-fl
Chromobacterium violaceum ATCC 12472
Chromohalobacter salexigens DSM 3043
Ciona intestinalis
Citrobacter koseri ATCC BAA-895
Clavibacter michiganensis subsp. michiganensis NCPPB 382
Clavibacter michiganensis subsp. sepedonicus
Clostridium acetobutylicum ATCC 824
Clostridium beijerinckii NCIMB 8052
Clostridium botulinum A str. ATCC 19397
Clostridium botulinum A str. ATCC 3502
Clostridium botulinum A str. Hall
Clostridium botulinum A3 str. Loch Maree
Clostridium botulinum B str. Eklund 17B
Clostridium botulinum B1 str. Okra
Clostridium botulinum F str. Langeland
Clostridium difficile 630
Clostridium kluyveri DSM 555
Clostridium novyi NT
Clostridium perfringens ATCC 13124
Clostridium perfringens phage phiSM101
Clostridium perfringens SM101
Clostridium perfringens str. 13
Clostridium phytofermentans ISDg
Clostridium tetani E88
Clostridium thermocellum ATCC 27405
Coccidioides immitis
Coccidioides posadasii
Cochliobolus heterostrophus
Colwellia psychrerythraea 34H
Coprinus cinereus
Corynebacterium diphtheriae NCTC 13129
Corynebacterium efficiens YS-314
Corynebacterium glutamicum ATCC 13032
Corynebacterium glutamicum R
Corynebacterium jeikeium K411
Corynebacterium urealyticum DSM 7109
Coxiella burnetii Dugway 5J108-111
Coxiella burnetii RSA 331
Coxiella burnetii RSA 493
Cryphonectria parasitica
Cryptococcus neoformans
Cryptosporidium parvum
Cupriavidus taiwanensis

Cyanidioschyzon merolae
Cyanothece sp. ATCC 51142
Cytophaga hutchinsonii ATCC 33406
Daphnia pulex
Debaryomyces hansenii
Dechloromonas aromatica RCB
Dehalococcoides ethenogenes 195
Dehalococcoides sp. BAV1
Dehalococcoides sp. CBDB1
Deinococcus geothermalis DSM 11300
Deinococcus radiodurans R1
Delftia acidovorans SPH-1
Desulfitobacterium hafniense Y51
Desulfococcus oleovorans Hxd3
Desulfotalea psychrophila LSv54
Desulfotomaculum reducens MI-1
Desulfovibrio desulfuricans G20
Desulfovibrio vulgaris subsp. vulgaris DP4
Desulfovibrio vulgaris subsp. vulgaris str. Hildenborough
Dichelobacter nodosus VCS1703A
Dictyostelium discoideum
Dictyostelium purpureum
Dinoroseobacter shibae DFL 12
Drosophila melanogaster
Ehrlichia canis str. Jake
Ehrlichia chaffeensis str. Arkansas
Ehrlichia ruminantium str. Gardel
Ehrlichia ruminantium str. Welgevonden
Emiliania huxleyi
Encephalitozoon cuniculi
Entamoeba histolytica
Enterobacter sakazakii ATCC BAA-894
Enterobacter sp. 638
Enterococcus faecalis V583
Erwinia carotovora subsp. atroseptica SCRI1043
Erythrobacter litoralis HTCC2594
Escherichia coli 536
Escherichia coli APEC O1
Escherichia coli ATCC 8739
Escherichia coli CFT073
Escherichia coli E24377A
Escherichia coli HS
Escherichia coli O157 H7 EDL933
Escherichia coli O157 H7 str. Sakai
Escherichia coli SECEC SMS-3-5
Escherichia coli str. K-12 substr. DH10B
Escherichia coli str. K-12 substr. MG1655
Escherichia coli UTI89
Escherichia coli W3110
Exiguobacterium sibiricum 255-15
Fervidobacterium nodosum Rt17-B1
Finegoldia magna ATCC 29328
Flavobacterium johnsoniae UW101
Flavobacterium psychrophilum JIP02 86
Francisella philomiragia subsp. philomiragia ATCC 25017
Francisella tularensis subsp. holarctica
Francisella tularensis subsp. holarctica FTNF002-00
Francisella tularensis subsp. holarctica OSU18
Francisella tularensis subsp. mediasiatica FSC147
Francisella tularensis subsp. novicida U112
Francisella tularensis subsp. tularensis FSC198
Francisella tularensis subsp. tularensis SCHU S4
Francisella tularensis subsp. tularensis WY96-3418
Frankia alni ACN14a
Frankia sp. CcI3
Frankia sp. EAN1pec
Fusarium graminearum
Fusarium oxysporum
Fusarium verticillioides
Fusobacterium nucleatum subsp. nucleatum ATCC 25586
Gallus gallus
Geobacillus kaustophilus HTA426
Geobacillus thermodenitrificans NG80-2
Geobacter metallireducens GS-15
Geobacter sulfurreducens PCA
Geobacter uraniireducens Rf4
Giardia lamblia
Gloeobacter violaceus PCC 7421
Gluconacetobacter diazotrophicus PAl 5
Gluconobacter oxydans 621H
Gramella forsetii KT0803
Granulibacter bethesdensis CGDNIH1
Haemophilus ducreyi 35000HP
Haemophilus influenzae 86-028NP
Haemophilus influenzae PittEE
Haemophilus influenzae PittGG
Haemophilus influenzae Rd KW20
Haemophilus somnus 129PT
Haemophilus somnus 2336
Hahella chejuensis KCTC 2396
Haloarcula marismortui ATCC 43049
Halobacterium salinarum R1
Halobacterium sp. NRC-1
Haloquadratum walsbyi DSM 16790
Halorhodospira halophila SL1
Helicobacter acinonychis str. Sheeba
Helicobacter hepaticus ATCC 51449
Helicobacter pylori 26695
Helicobacter pylori HPAG1
Helicobacter pylori J99
Heliobacterium modesticaldum Ice1
Helobdella robusta
Herminiimonas arsenicoxydans
Herpetosiphon aurantiacus ATCC 23779
Histoplasma capsulatum
Homo sapiens
Hyaloperonospora parasitica
Hyperthermus butylicus DSM 5456
Hyphomonas neptunium ATCC 15444
Idiomarina loihiensis L2TR
Ignicoccus hospitalis KIN4 I

Jannaschia sp. CCS1
Janthinobacterium sp. Marseille
Kineococcus radiotolerans SRS30216
Klebsiella pneumoniae subsp. pneumoniae MGH 78578
Laccaria bicolor
Lactobacillus acidophilus NCFM
Lactobacillus brevis ATCC 367
Lactobacillus casei ATCC 334
Lactobacillus delbrueckii subsp. bulgaricus ATCC 11842
Lactobacillus delbrueckii subsp. bulgaricus ATCC BAA-365
Lactobacillus fermentum IFO 3956
Lactobacillus gasseri ATCC 33323
Lactobacillus helveticus DPC 4571
Lactobacillus johnsonii NCC 533
Lactobacillus plantarum WCFS1
Lactobacillus reuteri F275
Lactobacillus sakei subsp. sakei 23K
Lactobacillus salivarius UCC118
Lactococcus lactis subsp. cremoris MG1363
Lactococcus lactis subsp. cremoris SK11
Lactococcus lactis subsp. lactis Il1403
Lawsonia intracellularis PHE MN1-00
Legionella pneumophila str. Corby
Legionella pneumophila str. Lens
Legionella pneumophila str. Paris
Legionella pneumophila subsp. pneumophila str. Philadelphia 1
Leifsonia xyli subsp. xyli str. CTCB07
Leishmania major
Leptospira biflexa serovar Patoc strain Patoc 1
Leptospira borgpetersenii serovar Hardjo-bovis JB197
Leptospira borgpetersenii serovar Hardjo-bovis L550
Leptospira interrogans serovar Copenhageni str. Fiocruz L1-130
Leptospira interrogans serovar Lai str. 56601
Leptothrix cholodnii SP-6
Leuconostoc citreum KM20
Leuconostoc mesenteroides subsp. mesenteroides ATCC 8293
Listeria innocua Clip11262
Listeria monocytogenes EGD-e
Listeria monocytogenes str. 4b F2365
Listeria welshimeri serovar 6b str. SLCC5334
Lottia gigantea
Lysinibacillus sphaericus C3-41
Magnaporthe grisea
Magnetococcus sp. MC-1
Magnetospirillum magneticum AMB-1
Mannheimia succiniciproducens MBEL55E
Maricaulis maris MCS10
Marinobacter aquaeolei VT8
Marinomonas sp. MWYL1
Melampsora laricis-populina
Mesoplasma florum L1
Mesorhizobium loti MAFF303099
Mesorhizobium sp. BNC1
Metallosphaera sedula DSM 5348
Methanobrevibacter smithii ATCC 35061
Methanocaldococcus jannaschii DSM 2661
Methanococcoides burtonii DSM 6242
Methanococcus aeolicus Nankai-3
Methanococcus maripaludis C5
Methanococcus maripaludis C6
Methanococcus maripaludis C7
Methanococcus maripaludis S2
Methanococcus vannielii SB
Methanocorpusculum labreanum Z
Methanoculleus marisnigri JR1
Methanopyrus kandleri AV19
Methanosaeta thermophila PT
Methanosarcina acetivorans C2A
Methanosarcina barkeri str. Fusaro
Methanosarcina mazei Go1
Methanosphaera stadtmanae DSM 3091
Methanospirillum hungatei JF-1
Methanothermobacter thermautotrophicus str. Delta H
Methylibium petroleiphilum PM1
Methylobacillus flagellatus KT
Methylobacterium extorquens PA1
Methylobacterium radiotolerans JCM 2831
Methylobacterium sp. 4-46
Methylococcus capsulatus str. Bath
Microcystis aeruginosa NIES-843
Micromonas pusilla
Micromonas strain RCC299
Microsporum canis
Microsporum gypseum
Monosiga brevicollis
Moorella thermoacetica ATCC 39073
Mucor circinelloides
Mus musculus
Mycobacterium abscessus
Mycobacterium avium 104
Mycobacterium avium subsp. paratuberculosis K-10
Mycobacterium bovis AF2122 97
Mycobacterium bovis BCG str. Pasteur 1173P2
Mycobacterium gilvum PYR-GCK
Mycobacterium leprae TN
Mycobacterium marinum M
Mycobacterium smegmatis str. MC2 155
Mycobacterium sp. JLS
Mycobacterium sp. KMS
Mycobacterium sp. MCS
Mycobacterium tuberculosis CDC1551
Mycobacterium tuberculosis F11
Mycobacterium tuberculosis H37Ra
Mycobacterium tuberculosis H37Rv
Mycobacterium ulcerans Agy99
Mycobacterium vanbaalenii PYR-1
Mycoplasma agalactiae PG2

Mycoplasma capricolum subsp. capricolum ATCC 27343
Mycoplasma gallisepticum R
Mycoplasma genitalium G37
Mycoplasma hyopneumoniae 232
Mycoplasma hyopneumoniae 7448
Mycoplasma hyopneumoniae J
Mycoplasma mobile 163K
Mycoplasma mycoides subsp. mycoides SC str. PG1
Mycoplasma penetrans HF-2
Mycoplasma pneumoniae M129
Mycoplasma pulmonis UAB CTIP
Mycoplasma synoviae 53
Mycosphaerella fijiensis
Mycosphaerella graminicola
Myxococcus xanthus DK 1622
Naegleria gruberi
Nanoarchaeum equitans Kin4-M
Natronomonas pharaonis DSM 2160
Nectria haematococca
Neisseria gonorrhoeae FA 1090
Neisseria meningitidis 053442
Neisseria meningitidis FAM18
Neisseria meningitidis MC58
Neisseria meningitidis Z2491
Nematostella vectensis
Neorickettsia sennetsu str. Miyayama
Neosartorya fischeri
Neurospora crassa
Nitratiruptor sp. SB155-2
Nitrobacter hamburgensis X14
Nitrobacter winogradskyi Nb-255
Nitrosococcus oceani ATCC 19707
Nitrosomonas europaea ATCC 19718
Nitrosomonas eutropha C91
Nitrosopumilus maritimus SCM1
Nitrosospira multiformis ATCC 25196
Nocardia farcinica IFM 10152
Nocardioides sp. JS614
Nostoc sp. PCC 7120
Novosphingobium aromaticivorans DSM 12444
Oceanobacillus iheyensis HTE831
Ochrobactrum anthropi ATCC 49188
Oenococcus oeni PSU-1
Onion yellows phytoplasma OY-M
Opitutus terrae PB90-1
Orientia tsutsugamushi Boryong
Oryza sativa
Ostreococcus lucimarinus
Ostreococcus RCC809
Ostreococcus tauri
Parabacteroides distasonis ATCC 8503
Paracoccidioides brasiliensis
Paracoccus denitrificans PD1222
Paramecium tetraurelia
Parvibaculum lavamentivorans DS-1
Pasteurella multocida subsp. multocida str. Pm70
Pediococcus pentosaceus ATCC 25745
Pelobacter carbinolicus DSM 2380
Pelobacter propionicus DSM 2379
Pelodictyon luteolum DSM 273
Pelotomaculum thermopropionicum SI
Petrotoga mobilis SJ95
Phaeodactylum tricornutum
Phanerochaete chrysosporium
Photobacterium profundum SS9
Photorhabdus luminescens subsp. laumondii TTO1
Phycomyces blakesleeanus
Physcomitrella patens
Phytophthora capsici
Phytophthora infestans
Phytophthora ramorum
Phytophthora sojae
Pichia stipitis
Picrophilus torridus DSM 9790
Plasmodium yoelii
Podospora anserina
Polaromonas naphthalenivorans CJ2
Polaromonas sp. JS666
Polynucleobacter necessarius STIR1
Polynucleobacter sp. QLW-P1DMWA-1
Populus trichocarpa
Porphyromonas gingivalis W83
Postia placenta
Prochlorococcus marinus str. AS9601
Prochlorococcus marinus str. MIT 9211
Prochlorococcus marinus str. MIT 9215
Prochlorococcus marinus str. MIT 9301
Prochlorococcus marinus str. MIT 9303
Prochlorococcus marinus str. MIT 9312
Prochlorococcus marinus str. MIT 9313
Prochlorococcus marinus str. MIT 9515
Prochlorococcus marinus str. NATL1A
Prochlorococcus marinus str. NATL2A
Prochlorococcus marinus subsp. marinus str. CCMP1375
Prochlorococcus marinus subsp. pastoris str. CCMP1986
Propionibacterium acnes KPA171202
Prosthecochloris vibrioformis DSM 265
Pseudoalteromonas atlantica T6c
Pseudoalteromonas haloplanktis TAC125
Pseudomonas aeruginosa PA7
Pseudomonas aeruginosa PAO1
Pseudomonas aeruginosa UCBPP-PA14
Pseudomonas entomophila L48
Pseudomonas fluorescens Pf-5
Pseudomonas fluorescens PfO-1
Pseudomonas mendocina ymp
Pseudomonas putida F1
Pseudomonas putida GB-1
Pseudomonas putida KT2440
Pseudomonas putida W619
Pseudomonas stutzeri A1501
Pseudomonas syringae pv. phaseolicola 1448A

Pseudomonas syringae pv. syringae B728a
Pseudomonas syringae pv. tomato str. DC3000
Psychrobacter arcticus 273-4
Psychrobacter cryohalolentis K5
Psychrobacter sp. PRwf-1
Psychromonas ingrahamii 37
Puccinia graminis
Pyrenophora tritici-repentis
Pyrobaculum aerophilum str. IM2
Pyrobaculum arsenaticum DSM 13514
Pyrobaculum calidifontis JCM 11548
Pyrobaculum islandicum DSM 4184
Pyrococcus abyssi GE5
Pyrococcus furiosus DSM 3638
Pyrococcus horikoshii OT3
Ralstonia eutropha H16
Ralstonia eutropha JMP134
Ralstonia metallidurans CH34
Ralstonia solanacearum GMI1000
Renibacterium salmoninarum ATCC 33209
Rhizobium etli CFN 42
Rhizobium leguminosarum bv. viciae 3841
Rhizopus oryzae
Rhodobacter sphaeroides 2.4.1
Rhodobacter sphaeroides ATCC 17025
Rhodobacter sphaeroides ATCC 17029
Rhodococcus sp. RHA1
Rhodoferax ferrireducens T118
Rhodopirellula baltica SH 1
Rhodopseudomonas palustris BisA53
Rhodopseudomonas palustris BisB18
Rhodopseudomonas palustris BisB5
Rhodopseudomonas palustris CGA009
Rhodopseudomonas palustris HaA2
Rhodospirillum rubrum ATCC 11170
Rickettsia akari str. Hartford
Rickettsia bellii OSU 85-389
Rickettsia bellii RML369-C
Rickettsia canadensis str. McKiel
Rickettsia conorii str. Malish 7
Rickettsia felis URRWXCal2
Rickettsia massiliae MTU5
Rickettsia prowazekii str. Madrid E
Rickettsia rickettsii str. Iowa
Rickettsia rickettsii str. Sheila Smith
Rickettsia typhi str. Wilmington
Roseiflexus castenholzii DSM 13941
Roseiflexus sp. RS-1
Roseobacter denitrificans OCh 114
Rubrobacter xylanophilus DSM 9941
Saccharomyces cerevisiae
Saccharophagus degradans 2-40
Saccharopolyspora erythraea NRRL 2338
Salinibacter ruber DSM 13855
Salinispora arenicola CNS-205
Salinispora tropica CNB-440
Salmonella enterica subsp. arizonae serovar 62
Salmonella enterica subsp. enterica serovar Choleraesuis str. SC-B67
Salmonella enterica subsp. enterica serovar Paratyphi A str. ATCC 9150
Salmonella enterica subsp. enterica serovar Paratyphi B str. SPB7
Salmonella enterica subsp. enterica serovar Typhi str. CT18
Salmonella enterica subsp. enterica serovar Typhi Ty2
Salmonella typhimurium LT2
Schizophyllum commune
Schizosaccharomyces pombe
Sclerotinia sclerotiorum
Selaginella moellendorffii
Serratia proteamaculans 568
Shewanella amazonensis SB2B
Shewanella baltica OS155
Shewanella baltica OS185
Shewanella baltica OS195
Shewanella denitrificans OS217
Shewanella frigidimarina NCIMB 400
Shewanella halifaxensis HAW-EB4
Shewanella loihica PV-4
Shewanella oneidensis MR-1
Shewanella pealeana ATCC 700345
Shewanella putrefaciens CN-32
Shewanella sediminis HAW-EB3
Shewanella sp. ANA-3
Shewanella sp. MR-4
Shewanella sp. MR-7
Shewanella sp. W3-18-1
Shewanella woodyi ATCC 51908
Shigella boydii CDC 3083-94
Shigella boydii Sb227
Shigella dysenteriae Sd197
Shigella flexneri 2a str. 2457T
Shigella flexneri 2a str. 301
Shigella flexneri 5 str. 8401
Shigella sonnei Ss046
Silicibacter pomeroyi DSS-3
Silicibacter sp. TM1040
Sinorhizobium medicae WSM419
Sinorhizobium meliloti 1021
Sodalis glossinidius str. morsitans
Solibacter usitatus Ellin6076
Sorangium cellulosum So ce 56
Sorghum bicolor
Sphingomonas wittichii RW1
Sphingopyxis alaskensis RB2256
Sporobolomyces roseus
Stagonospora nodorum
Staphylococcus aureus RF122
Staphylococcus aureus subsp. aureus COL
Staphylococcus aureus subsp. aureus JH1
Staphylococcus aureus subsp. aureus JH9
Staphylococcus aureus subsp. aureus MRSA252
Staphylococcus aureus subsp. aureus MSSA476
Staphylococcus aureus subsp. aureus Mu3
Staphylococcus aureus subsp. aureus Mu50
Staphylococcus aureus subsp. aureus MW2

Staphylococcus aureus subsp. aureus N315
Staphylococcus aureus subsp. aureus NCTC 8325
Staphylococcus aureus subsp. aureus str. Newman
Staphylococcus aureus subsp. aureus USA300
Staphylococcus aureus subsp. aureus USA300_TCH1516
Staphylococcus epidermidis ATCC 12228
Staphylococcus epidermidis RP62A
Staphylococcus haemolyticus JCSC1435
Staphylococcus saprophyticus subsp. saprophyticus ATCC 15305
Staphylothermus marinus F1
Streptococcus agalactiae 2603V R
Streptococcus agalactiae A909
Streptococcus agalactiae NEM316
Streptococcus gordonii str. Challis substr. CH1
Streptococcus mutans UA159
Streptococcus pneumoniae CGSP14
Streptococcus pneumoniae D39
Streptococcus pneumoniae Hungary19A-6
Streptococcus pneumoniae R6
Streptococcus pneumoniae TIGR4
Streptococcus pyogenes M1 GAS
Streptococcus pyogenes MGAS10270
Streptococcus pyogenes MGAS10394
Streptococcus pyogenes MGAS10750
Streptococcus pyogenes MGAS2096
Streptococcus pyogenes MGAS315
Streptococcus pyogenes MGAS5005
Streptococcus pyogenes MGAS6180
Streptococcus pyogenes MGAS8232
Streptococcus pyogenes MGAS9429
Streptococcus pyogenes SSI-1
Streptococcus pyogenes str. Manfredo
Streptococcus sanguinis SK36
Streptococcus suis 05ZYH33
Streptococcus suis 98HAH33
Streptococcus thermophilus CNRZ1066
Streptococcus thermophilus LMD-9
Streptococcus thermophilus LMG 18311
Streptomyces avermitilis MA-4680
Streptomyces coelicolor A3
Streptomyces griseus subsp. griseus NBRC 13350
Sulfolobus acidocaldarius DSM 639
Sulfolobus solfataricus P2
Sulfolobus tokodaii str. 7
Sulfurimonas denitrificans DSM 1251
Sulfurovum sp. NBC37-1
Symbiobacterium thermophilum IAM 14863
Synechococcus elongatus PCC 6301
Synechococcus elongatus PCC 7942
Synechococcus sp. CC9311
Synechococcus sp. CC9605
Synechococcus sp. CC9902
Synechococcus sp. JA-2-3Ba
Synechococcus sp. JA-3-3Ab
Synechococcus sp. PCC 7002
Synechococcus sp. RCC307
Synechococcus sp. WH 7803
Synechococcus sp. WH 8102
Synechocystis sp. PCC 6803
Syntrophobacter fumaroxidans MPOB
Syntrophomonas wolfei subsp. wolfei str. Goettingen
Syntrophus aciditrophicus SB
Takifugu rubripes
Tetrahymena thermophila
Thalassiosira pseudonana
Thermoanaerobacter pseudethanolicus ATCC 33223
Thermoanaerobacter sp. X514
Thermoanaerobacter tengcongensis MB4
Thermobifida fusca YX
Thermococcus kodakarensis KOD1
Thermofilum pendens Hrk 5
Thermoplasma acidophilum DSM 1728
Thermoplasma volcanium GSS1
Thermoproteus neutrophilus V24Sta
Thermosipho melanesiensis BI429
Thermosynechococcus elongatus BP-1
Thermotoga lettingae TMO
Thermotoga maritima MSB8
Thermotoga petrophila RKU-1
Thermus thermophilus HB27
Thermus thermophilus HB8
Thiobacillus denitrificans ATCC 25259
Thiomicrospira crunogena XCL-2
Toxoplasma gondii
Treponema denticola ATCC 35405
Treponema pallidum subsp. pallidum str. Nichols
Trichoderma atoviride
Trichoderma reesei
Trichoderma virens
Trichodesmium erythraeum IMS101
Trichomonas vaginalis
Trichophyton equinum
Trichoplax adhaerens
Tropheryma whipplei str. Twist
Tropheryma whipplei TW08 27
Trypanosoma brucei
Trypanosoma cruzi
Uncinocarpus reesii
uncultured methanogenic archaeon RC-I
Ureaplasma parvum serovar 3 str. ATCC 27815
Ureaplasma parvum serovar 3 str. ATCC 700970
Ustilago maydis
Verminephrobacter eiseniae EF01-2
Verticillium albo-atrum
Verticillium dahliae
Vibrio cholerae O1 biovar eltor str. N16961
Vibrio cholerae O395
Vibrio fischeri ES114
Vibrio harveyi ATCC BAA-1116
Vibrio parahaemolyticus RIMD 2210633

Vibrio vulnificus CMCP6
Vibrio vulnificus YJ016
Volvox carteri
Wigglesworthia glossinidia endosymbiont of Glossina brevipalpis
Wolbachia endosymbiont of Drosophila melanogaster
Wolbachia endosymbiont strain TRS of Brugia malayi
Wolinella succinogenes DSM 1740
Xanthobacter autotrophicus Py2
Xanthomonas axonopodis pv. citri str. 306
Xanthomonas campestris pv. campestris str. 8004
Xanthomonas campestris pv. campestris str. ATCC 33913
Xanthomonas campestris pv. vesicatoria str. 85-10
Xanthomonas oryzae pv. oryzae KACC10331
Xanthomonas oryzae pv. oryzae MAFF 311018
Xenopus tropicalis
Xylella fastidiosa 9a5c
Xylella fastidiosa M12
Xylella fastidiosa M23
Xylella fastidiosa Temecula1
Yarrowia lipolytica
Yersinia enterocolitica subsp. enterocolitica 8081
Yersinia pestis Angola
Yersinia pestis Antiqua
Yersinia pestis biovar Microtus str. 91001
Yersinia pestis CO92
Yersinia pestis KIM
Yersinia pestis Nepal516
Yersinia pestis Pestoides F
Yersinia pseudotuberculosis IP 31758
Yersinia pseudotuberculosis IP 32953
Yersinia pseudotuberculosis YPIII
Zymomonas mobilis subsp. mobilis ZM4

9 Appendix: fdfBLAST Perl Code Listing

```
#!/usr/bin/perl
##########################################
# fdfBLAST 2010-01-03 17:01              #
$VERSION = "1.9.9";

# (c) CEEM MMX                           #
##########################################

# 2010-03-11 Added -o T command to formatDB - needed for formatcmd to extract
sequences from non-ncbi deflines
# 2010-03-xx Ignore subject sequences <50 bases
# 2010-03-xx Ignore data when paralogues are detected and where "fusions"
exist...

# Import Modules
use Cwd;                # Gets pathname of current working directory
use Switch;             # A switch statement for Perl
use Math::BigFloat;     # Arbitrary size floating point math package (e-
values)
use Bio::SearchIO;      # Bioperl for input/output of BLAST etc
use File::Basename;     # Remove path information and extract 8.3 filename
use GD;                 # Creates PNG images
#use GD::SVG;           # Creates SVG images
use Time::Local;        # For time elapsed when running different stages

# Directory Information
$working_directory = getcwd;
### These should be safe to change if needed
#$genome_directory = "$working_directory/genomes";
$bin_directory     = "$working_directory/blast/bin";
###
&set_genome_dir;
sub set_genome_dir {

        @genome_directories = ("$working_directory/genomes",
"/media/Lofn\_/genomes"); # Add locations to this array...

        print "Genomes Directory Menu\n";
        print "**********************\n";
        for ( my $i = 0 ; $i <= $#genome_directories ; $i++ ) {
                print "$i) $genome_directories[$i]\n";
        }
        print "O) Other\nChoose Genome Directory?\n>:";
        chomp( my $menu_choice = <STDIN> );
        if ( $menu_choice =~ m/O/is ) {
                print "Please enter location of genome directory\n>:";
                chomp ( my $user_dir = <STDIN> );
                $genome_directory = $user_dir;
                if ( -e $genome_directory && -d $genome_directory ) {
                        # Do nothing
                }
                else {
                        &set_genome_dir;
                }
        }
        else {
```

```
                $genome_directory = "$genome_directories[$menu_choice]";
        }
        print "\nYou chose $genome_directory\n";
}

# Run these subroutines first

&detect_multi_core;     # Check for single or multi-core machine for BLAST
&run_blast_check;       # Check for BLAST
&which_genomes;         # Check which genome folder to use!
&initial_menu;          # User input for run number or next
&run_ID;                # Sets run number if none set, sets directory paths
&menu;                  # Main menu

sub detect_multi_core {

     $pprocs = `grep -i "physical id" /proc/cpuinfo | sort -u | wc -l | tr -d
'\n'`;
     $lprocs = `grep -i "processor" /proc/cpuinfo | sort -u | wc -l | tr -d
'\n'`;

     if ( $lprocs >= 2 ) {

          #print "$int1 cores have been detected. BLAST will attempt to use
them.\n";
          $core_num = "$lprocs";
          $cores    = "multi";
     }
     else {

          #print "You do not have a multi-core processor or we cannot
identify more than 1 core.\n";
          $core_num = "1";
          $cores    = "single";
     }
}

sub run_blast_check {

     # Very simple check - only if BLAST directory exists.
     # Ideally we would run blast and check version.
     if ( -e $bin_directory && -d $bin_directory ) {
          print "Blast Detected\n";
          print `clear`, "\n";
     }
     else {
          print
            "\n****\n\nYou do not appear to have LEGACY BLAST installed or it
is installed in the wrong location.\nPlease download the LEGACY version from
http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=
Download and extract the contents to ../fdf/blast/ or see the Readme
file.\n\n****\n";
          print "\nContinue without BLAST? (y/N)\n";
          print ">:";
          chomp( $blast_choice = <STDIN> );

          switch ($blast_choice) {

               case /Y/i {
                    print "Blast Detection Overide\n";
                    print `clear`, "\n";
```

```
                }
                case /N/i {
                     print `clear`, "\n";
                     print "Thanks for using fdfBLAST, goodbye...\n\nDeveloped
by;\n";
                     print "|C|E| Centre for Eukaryotic Evolutionary
Microbiology
|E|M| Cell Biology * Ecology * Evolution
      School of Biosciences, Univeristy of Exeter, UK
      http://www.ex.ac.uk/ceem
";
                     exit;
                }
                else {
                     &run_blast_check;
                }
           }
     }
}

sub which_genomes {

     @all_files = glob("$genome_directory/*");
     @folders;
     foreach my $item (@all_files) {
          if ( -d $item ) {    #Put all folders into array
               push( @folders, $item );
          }
     }
     $folder_num = @folders;

     print "Genome Folder Menu\n";
     print "******************\n";
     for ( $i = 0 ; $i < $folder_num ; $i++ ) {
          print "$i) $folders[$i]\n";
     }

     print "Please enter the number for the directory where your genomes are
located.\n";
     print ">:";

     chomp( $menu_choice = <STDIN> );
     if (    $menu_choice >= $folder_num
          || $menu_choice < "0"
          || $menu_choice eq m/[a-z]/ ) {
          print `clear`, "\n";
          print "\nIncorrect Menu Choice!\n\n";
          &which_genomes;

     }
     else {
          $genome_directory = "$folders[$menu_choice]";
     }
}

sub initial_menu {
     print `clear`, "\n";
     print "**************************************************\n";
     print "  __        _    __  ____    _              _     ____  _____  \n";
     print " / _|  __| | / _|| __ ) | |       / \\   / ___||_   _|\n";
     print "| |_  / _` || |_ |  _ \\ | |      / _ \\  \\___ \\  | |  \n";
```

```
     print "|  _|| (_| ||  _|| |_) || |___  / ___ \\  ___) | | |  \n";
     print "|_|   \\__,_||_|  |____/ |_____|/_/   \\_\\|____/  |_|  \n";
     print "|C|E|   $lprocs logical in $pprocs physical processor(s)
detected\n";
     print "|E|M|           v$VERSION © Guy Leonard & CEEM 2008-2009\n";
     print "*************************************************\n";

     print
       "Please indicate your previous 'run' number or enter a new number.\nIf
no 'run' is entered, a number will be generated for you.\n";
}

sub run_ID {
     print ">:";
     chomp( $run_ID = <STDIN> );
     if ( $run_ID eq "" ) {
          $run_ID = time();
          print "Your ID is now: $run_ID\n";
     }
     $run_directory = "$working_directory/run/$run_ID";

     if ( -e $run_directory && -d $run_directory ) {
          $g2gc_directory = "$genome_directory/g2gc";
          $run_directory  = "$working_directory/run/$run_ID";

          #
          $gene_hits_directory = "$run_directory/gene_hits";
          $gene_hits_initial   = "$gene_hits_directory/initial";
          $gene_hits_lookup    = "$gene_hits_directory/lookup";

          #
          $gene_hits_results       = "$run_directory/results";
          $gene_hits_differentials = "$gene_hits_results/differentials";
          $gene_hits_duplications  = "$gene_hits_results/duplications";

          #
     }
     else {
          mkdir( $run_directory, 0755 )
            || die "Cannot be the same name as existing directory!  or talk
to the rubber duck";
          $g2gc_directory = "$genome_directory/g2gc";
          $run_directory  = "$working_directory/run/$run_ID";

          #
          $gene_hits_directory = "$run_directory/gene_hits";
          $gene_hits_initial   = "$gene_hits_directory/initial";
          $gene_hits_lookup    = "$gene_hits_directory/lookup";

          #
          $gene_hits_results       = "$run_directory/results";
          $gene_hits_differentials = "$gene_hits_results/differentials";
          $gene_hits_duplications  = "$gene_hits_results/duplications";

          #
     }
}

sub menu {
     print `clear`, "\n";
```

```
# Program title
print "  __      _   __  ___  _          _     ___  _____  \n";
print " / _| __| | / _||_ _)| |     /_\\  / __||_   _|\n";
print "|  _|/ _` ||  _|| _ \\| |__  / _ \\ \\__ \\  | |  \n";
print "|_|  \\__,_||_|  |___/|____|/_/ \\_\\|___/  |_|  \n";
print "|C|E|\n";
print "|E|M|                                    v$VERSION\n\n";

# Options for BLAST
print "BLAST Options\n-------------\n";
print "1) FORMATDB\t\t\tB) 1 & 2\n";
print "2) BLASTALL\n\n";

# Options for fdf
print "fdfBLAST Options\n----------------\n";
print "3) Hit Lists & Lookup Tables\t4) Differential Lists\n";
print "5) Identify Fusions\t\t6) Identify Duplications\n\n";

# Other Options
print "Other Options\n-------------\n";
print "A) All (1 - 6)\t\t\tF) 3 to 6\n\n";
print "8) Change Run ID\t\tQ) Quit\n\n";

#
print "$run_ID>:";

chomp( $menu_choice = <STDIN> );

switch ($menu_choice) {

     case "1" {
          &menu_one;
          &return_or_quit;
     }
     case "2" {
          &menu_two;
          &return_or_quit;
     }
     case /B/i {
          &menu_one;
          &menu_two;
          &return_or_quit;
     }
     case "3" {
          &menu_three;

          #&menu_l;
          &return_or_quit;
     }
     case "4" {
          &menu_four;
          &return_or_quit;
     }

     # Hidden setting for running Lookup tables only
     case /L/i {
          &menu_l;
          &return_or_quit;
     }
     case "5" {
          &menu_five;
```

```
                &return_or_quit;
            }
            case "6" {
                &menu_six;
                &return_or_quit;
            }

            # Hidden settings for testing purposes
            case /T/i {
                &fusion_html_output;
            }
            case /r/i {
                 &read_domain_file;
            }
            #
            case /A/i {
                &menu_a;
                &return_or_quit;
            }
            case /F/i {
                &menu_f;
                &return_or_quit;
            }
            case "8" {
                print `clear`, "\n";
                print "Please indicate your desired run number or enter a new
number\n";
                &run_ID;
                &menu;
            }
            case /Q/i {
                print `clear`, "\n";
                print "Thanks for using fdfBLAST, goodbye...\n\nDeveloped
by;\n";
                print "|C|E| Centre for Eukaryotic Evolutionary Microbiology
|E|M| Cell Biology * Ecology * Evolution
      School of Biosciences, Univeristy of Exeter, UK
      http://www.ex.ac.uk/ceem
";
                last;
            }
            else {
                print `clear`, "\n";
                print "*! Wrong Menu Choice - Please Try Again !*\n";
                &menu;
            }
        }
}

sub menu_one {
     $menu_choice = "FormatDB";
     &print_time("start");
     &get_genomes;
     &formatdb;
     &print_time("end");
}

sub menu_two {
     $menu_choice = "BlastAll";
     &print_time("start");
     &get_genomes;
```

```
	&blastall;
	&print_time("end");
}

sub menu_three {
	$menu_choice = "Extract Gene Hit Lists";
	&print_time("start");
	&get_genomes;
	&get_g2gc_files;
	&run_gene_hits_helper;
	&run_gene_hits;
	&generate_lookup_tables;
	&print_time("end");
}

sub menu_l {
	$menu_choice = "Lookup Tables";
	&print_time("start");
	&get_genomes;
	&generate_lookup_tables;
	&print_time("end");
}

sub menu_four {
	$menu_choice = "Parse Tables";
	print "Limit by Hit Number? (default: 250)\n";
	$hit_limit = "250";
	print ">:";
	chomp( $hit_limit = <STDIN> );
	&print_time("start");
	&get_genomes;
	&differential_new;
	&print_time("end");
}

sub menu_five {
	$menu_choice = "Identify Fusions";
	&print_time("start");
	&parse_lookup;
	&read_domain_file;
	&fusion_scan;
	&print_time("end");
}

sub menu_six {
	$menu_choice = "Identify Duplications";
	&print_time("start");
	&duplication_scan;
	&print_time("end");
}

sub menu_a {
	$menu_choice = "All";
	&print_time("start");
	&run_gene_hits_helper;
	&parse_lookup;
	&menu_one;
	&menu_two;
	&get_genomes;
	&get_g2gc_files;
	&run_gene_hits;
```

```
	&menu_l;
	&get_genomes;
	&differential_new;
	&fusion_scan;
	&menu_six;
	&print_time("end");
}

sub menu_f {
	$menu_choice = "3 to 6";
	&print_time("start");
	&run_gene_hits_helper;
	&parse_lookup;
	&get_genomes;
	&get_g2gc_files;
	&run_gene_hits;
	&menu_l;
	&get_genomes;
	&differential_new;
	&fusion_scan;
	&menu_six;
	&print_time("end");
}

sub return_or_quit {
	print "\nReturn to (M)enu or (Q)uit\n";
	print ">:";
	chomp( $menu_choice = <STDIN> );
	switch ($menu_choice) {
		case /Q/i {
			print `clear`, "\n";
			print "Thanks for using fdfBLAST, goodbye...\n\nDeveloped
by;\n";
			print "|C|E| Centre for Eukaryotic Evolutionary Microbiology
|E|M| Cell Biology * Ecology * Evolution
	School of Biosciences, Univeristy of Exeter, UK
	http://www.ex.ac.uk/ceem
";
		   last;
		}
		case /M/i {
			print `clear`, "\n";
			&menu;
		}
		else {
			&return_or_quit;
		}
	}
}

sub print_time {
	my $time = shift;
	if ( $time eq "start" ) {
		open TIME, ">>$run_directory/time.txt";
		print TIME "Menu Choice = $menu_choice\n";
		$start_time = scalar localtime(time);
		print TIME " Start Time = $start_time\n";
		close(TIME);
	}
	elsif ( $time eq "end" ) {
		open TIME, ">>$run_directory/time.txt";
```

```
          $end_time = scalar localtime(time);
          print TIME "   End Time = $end_time\n";

          &dhms;

          printf TIME ( " Total Time = %4d days%4d hours%4d minutes%4d
seconds\n\n", @parts[ 7, 2, 1, 0 ] );

          close(TIME);
     }
}

sub dhms {

     my %months = (
                    Jan => 1,
                    Feb => 2,
                    Mar => 3,
                    Apr => 4,
                    May => 5,
                    Jun => 6,
                    Jul => 7,
                    Aug => 8,
                    Sep => 9,
                    Oct => 10,
                    Nov => 11,
                    Dec => 12
     );

     my $beginning = $start_time;
     my $end       = $end_time;

     my @b = split( /\s+|:/, $beginning );
     my @e = split( /\s+|:/, $end );

     my $b = timelocal( $b[5], $b[4], $b[3], $b[2], $months{ $b[1] } - 1,
$b[-1] );
     my $e = timelocal( $e[5], $e[4], $e[3], $e[2], $months{ $e[1] } - 1,
$e[-1] );

     my $elapsed = $e - $b;

     @parts = gmtime($elapsed);

   return @parts;

}

sub print_log {
     my $message = shift;
     open LOG, ">>$run_directory/log.txt";
     print LOG "$message";
     close(LOG);
}

## Get all genome names from the genome directory and put into an array
# Files must have .fas as their extension
sub get_genomes {

     # Assign all files with extension .fas (a small assumption) to the array
@file_names
```

```
	@file_names = <$genome_directory/*.fas>;

	# Loop for all file names in @file_names array
	foreach $file (@file_names) {
		$file = fileparse($file);
	}

	# This is the number of .fas files in the genome directory
	$genome_num = @file_names;
}

## A method to call the formatdb program from the command line
# and run it for each .fas FASTA formatted genome in a directory
# This will probably need updating to include the new BLAST+ programs
sub formatdb {

	# User output, number of genomes to format

	print "Number of Genomes = $genome_num\n";
	print "FORTMATDB - Formating Databases...\n";

	# For loop, $i up to number of genomes do

	for ( $i = 0 ; $i < $genome_num ; $i++ ) {
		print "$file_names[$i] \x3E\x3E\x3E";

		# This line calls the formatdb program within a previously set
directory
		# it then outputs the databse genomes files to another directory
		$results =
		  `$bin_directory/formatdb -t $genome_directory/$file_names[$i] -i
$genome_directory/$file_names[$i] -p T -o T`;
		print " Completed\n";
	}
	print "FORMATDB - Finished...\n\n";
}

# This will probably need updating to include the new BLAST+ programs
sub blastall {

	# Create the g2gc output folder, if not then error, quit. This is a
little harsh...
	mkdir( $g2gc_directory, 0777 )
	  || die "This procedure has already been run, please refer to you
previous run, $run_ID.\n";
	print "Number of\: Genomes = $genome_num\tOutput Files = $genome_num x
$genome_num = "
	  . $genome_num * $genome_num . "\n";
	print "Performing BLASTP\n";

	for ( $i = 0 ; $i < $genome_num ; $i++ ) {
		for ( $j = 0 ; $j < $genome_num ; $j++ ) {
			( $file_i, $dir, $ext ) = fileparse( $file_names[$i], '\..*'
);
			( $file_j, $dir, $ext ) = fileparse( $file_names[$j], '\..*'
);
			print "$file_i to $file_j \x3E\x3E\x3E ";
			$results =
			  `$bin_directory/blastall -p blastp -d
$genome_directory/$file_names[$i] -i $genome_directory/$file_names[$j] -m 0 -
a $core_num -F F -o $g2gc_directory/$file_j\_$file_i.bpo`;
```

```
                print "Completed\n";
            }
        }
        print "\nBLASTALL Successful\n";
}

# Retrieve list of Blast Parse Output files from directory
sub get_g2gc_files {

        @g2gc_file_names = <$g2gc_directory/*.bpo>;
        $g2gc_length     = @g2gc_file_names;
}

# Change user submission from 1e-10 to 0.01 etc
sub evaluate {
        $value = shift;

        # Perl method bstr will only change '1e' not 'e' to decimal, therefore
prefix value with '1'
        if ( $value =~ m/^e/ ) {
                $value = "1" . $value;
                $value = Math::BigFloat->bstr($value);
        }
        if ( $value =~ m/e/ ) {
                $value = Math::BigFloat->bstr($value);
        }
     return $value;
}

sub run_gene_hits_helper {

        # Create the gene_hits output folder, if unable then on error, quit.
        # This could probably be handled better - overwrite? sub directory? etc
        mkdir( $gene_hits_directory, 0755 )
          || die "This procedure has already been run, please refer to you
previous run or talk to the rubber duck, $run_ID.\n";

        mkdir( $gene_hits_initial, 0755 );
        print "E Value Upper Limit - e.g 1e-10\n>:";
        chomp( $upper_limit = <STDIN> );
        if ( $upper_limit == "" ) {
                $upper_limit = "1e-10";
        }
        &print_log("Gene Hit List E-Value Range\n");
        &print_log("Upper Value = $upper_limit\n");
        &evaluate($upper_limit);
        $upper_limit = $value;
        print "E Value Lower Limit - e.g 0\n>:";
        chomp( $lower_limit = <STDIN> );
        if ( $lower_limit eq "" ) {
                $lower_limit = "0";
        }
        &print_log("Lower Value = $lower_limit\n");
        &evaluate($lower_limit);
        $lower_limit = $value;
        &print_log("Upper Value = $upper_limit\n");
        &print_log("Lower Value = $lower_limit\n\n");
        print "\nWorking\n";
}

sub run_gene_hits {
```

```
	for ( $i = 0 ; $i < $g2gc_length ; $i++ ) {
		$current = $i + 1;
		print "\n$current of $g2gc_length";

		# Assign file name in iterative loop
		$in_file = "$g2gc_file_names[$i]";

		( $in_filename, $dir, $ext ) = fileparse( $in_file, '\..*' );

		#$in_filename = fileparse($in_file);
		print "    $in_filename";

		# Internal iterators for 2D array
		# X = Blast sequence number, Y = Hits against query
		$x_pos              = 0;
		#@comparison_length = ();
		undef(@comparison_length);

		my $search = new Bio::SearchIO( '-format' => 'blast',
		                                '-file'   => $in_file );
		while ( my $result = $search->next_result ) {
			$y_pos                    = 0;
			$query_accession          = $result->query_accession;
			$query_length             = $result->query_length;
			$query_genome[$x_pos][0] = "$query_accession";
			$query_genome[$x_pos][1] = "$query_length";
			while ( my $hit = $result->next_hit ) {
				$hit_accession    = $hit->accession;
				$hit_length       = $hit->length;
				$hit_significance = $hit->significance;

				while ( my $hsp = $hit->next_hsp ) {
					$percent_identity = $hsp->percent_identity;

					# Range of query that hits the subject gene
					@query_range = $hsp->range('query');
					#####@hit_range   = $hsp->range('hit');

					# We don't really need 13 decimal places, round to
none..
					$percent_identity = sprintf( "%.0f",
$percent_identity );

					#####print
"Query\t$query_accession\t\t$query_range[0] \- $query_range[1]\n";
					#####print "HIt\t$hit_accession\t\t$hit_range[0] \-
$hit_range[1]\n";
				}

				$comparison_genome[$x_pos][$y_pos] =

"$hit_accession,$hit_length,$hit_significance,$percent_identity,$query_range[
0],$query_range[1],";
				$y_pos++;
				#####print "\nY$y\t$query_accession\t$hit_accession\n";
			}

			push( @comparison_length, $y_pos );
			$x_pos++;
			#####print "\nX$x_pos\tY$y_pos\n";
```

```
            }
            #####
            #print "\nXXXXX\n";
            #foreach (@comparison_length) {
            #     print $_;
            #     }
            #print "\nXXXXX\n";
            #####
            open OUTPUT, ">$gene_hits_initial/$in_filename.csv";
            ##### Surely we don't need this twice?
            $query_genome_length = $x_pos;
            ######$query_genome_length = @query_genome;

            #Append hit numbers to query_genome array
            for ( $x = 0 ; $x < $query_genome_length ; $x++ ) {

                  $comparison_genome_length = $comparison_length[$x];
                  $count                    = 0;
                  #####print "\nCGL$comparison_genome_length\tC$count";
                  for ( $d = 0 ; $d < $comparison_genome_length ; $d++ ) {

                        $evalue = "$comparison_genome[$x][$d]";
                        $evalue =~ m/(.*?)\,(\d*)\,(.*?)\,(.*?)\,/;
                        $evalue = $3;
                        &evaluate( $value = "$evalue" );
                        $evalue = $value;
                        if (    ( $evalue <= $upper_limit )
                              && ( $evalue >= $lower_limit ) ) {
                              $count++;
                              #####print "\tC$count";
                        }
                  }
                  $query_genome[$x][2] = $count;
                  #####print "\tTC$count\n";
            }

            for ( $x = 0 ; $x < $query_genome_length ; $x++ ) {

                  print OUTPUT
"$query_genome[$x][2],$query_genome[$x][0],$query_genome[$x][1],";

                  ######$comparison_genome_length1 = $comparison_length[$x];
                  $comparison_genome_length2 = $query_genome[$x][2];
                  #####print
"\nCGL$comparison_genome_length1\/$comparison_genome_length2";

                  for ( $y = 0 ; $y < $comparison_genome_length2 ; $y++ ) {

                        $evalue = "$comparison_genome[$x][$y]";
                        $evalue =~ m/(.*?)\,(\d*)\,(.*?)\,(.*?)\,/;
                        $evalue = $3;
                        &evaluate( $value = "$evalue" );
                        $evalue = $value;

                        ## 0 is the lower limit as the lower the E-value,
                        ## or the closer it is to zero, the more "significant"
the match is
                        ## therefore
                        #####print "\tX$x\tY$y\n";
                        #####print "$query_genome[$x][0] -
$comparison_genome[$x][$y]\n";
```

```
                    if (    ( $evalue <= $upper_limit )
                         && ( $evalue >= $lower_limit ) ) {
                         print OUTPUT "$comparison_genome[$x][$y]";
                    }
               }
               print OUTPUT "\n";
          }
          close(OUTPUT);
          #@comparison_genome = ();
          #@query_genome      = ();
          undef(@comparison_genome);
          undef(@query_genome);
     }
     print "\nFinished\n";

}

# A method to create tables of the recorded hits for each genome group
sub generate_lookup_tables {
     print "Generating Lookup Tables";

     # Check to see if the initial hit lists exist
     if ( -e $gene_hits_initial && -d $gene_hits_initial ) {

          # Number of gene hit files
          $gene_hits_length = @gene_hits_file_names;

          # Edited gene hit lists get their own directory
          mkdir( $gene_hits_lookup, 0755 );

          # Internal counter
          $run = 0;

          # As we have called get_genomes in menu_l we can use
          # genome_num and file_names
          # Two for loops to iterate through each gene hits initial file
          # and add the values to a 2d array for output to file
          for ( $i = 0 ; $i < $genome_num ; $i++ ) {
               for ( $j = 0 ; $j < $genome_num ; $j++ ) {

                    ( $file_i, $dir, $ext ) = fileparse( $file_names[$i],
'\..*' );
                    ( $file_j, $dir, $ext ) = fileparse( $file_names[$j],
'\..*' );
                    open IN, "<$gene_hits_initial/$file_i\_$file_j\.csv";

                    # We're using while with an iterator instead of foreach
to
                    # read in the file line by line adding each column to the
2d array
                    $x = 0;
                    while (<IN>) {
                         my ($line) = $_;
                         chomp($line);
                         @temp = split( /,/, $line );
                         $AoA[$x][$j] = $temp[0];
                         $x++;
                    }

                    # Then we add up the values in each column for each file
                    # using a for loop
```

```
                    $total_in_array_column = 0;
                    for $i ( 0 .. $#AoA ) {
                         $total_in_array_column += $AoA[$i][$j];
                    }

                    # Then those values are added plainly to an array,
                    # the sum of all values is calculated and if they are
                    # equal to 0 then there are no hits, else we continue...
                    push( @check_for_zero, $total_in_array_column );
                    $total_of_array_columns += $_ for @check_for_zero;
                    if ( $total_of_array_columns == 0 ) {
                         print "\nThere are no hits at all, please try some
different e-values...\n";
                       last;
                    }
                    ###
               }    # End second (internal) for loop
               close(IN);
               open OUT, ">$gene_hits_lookup/$file_i\.csv";

               # Here we use two for loops to iterate through the 2D array
               # for $i from 0 to length of @AoA
               for $i ( 0 .. $#AoA ) {

                    # A reference to the row at position $i
                    $aref = $AoA[$i];

                    # Length of row
                    $n          = @$aref - 1;
                    $line_total = 0;

                    # Second for, $j from 0 to length of row
                    for $j ( 0 .. $n ) {

                         # Sum the individual numbers of each row
                         $line_total = $line_total + $AoA[$i][$j];

                         # Print out each element of the row
                         print OUT "$AoA[$i][$j],";
                    }
                    print OUT "\n";
               }

               # Increment run
               $run = $run + 1;

               # Reset array, so length is not kept at largest
               # @AoA = ();
               undef(@AoA);
               print ".";
          }
          close(OUT);
     }
     print "\n";
}

sub parse_lookup {

     print "Lower Ratio Value (default: 0.1)\n>:";
     chomp( $lower_ratio = <STDIN> );
     if ( $lower_ratio == "" ) {
```

```
          $lower_ratio = "0.1";
     }
     print "Higher Ratio Value (default: 1.0)\n>:";
     chomp( $higher_ratio = <STDIN> );
     if ( $higher_ratio == "" ) {
          $higher_ratio = "1.0";
     }
}

sub differential_new {

     # Output directory
     mkdir( $gene_hits_results,       0755 );
     mkdir( $gene_hits_differentials, 0755 );
     mkdir( $gene_hits_duplications,  0755 );

     print "\nDifferential Extraction\n";
     $verbose          = 0;
     @variables        = @_;
     $limit            = $variables[0];
     @lookup_filenames = <$gene_hits_lookup/*.csv>;
     $lookup_number    = @lookup_filenames;

     for ( $k = 0 ; $k < $lookup_number ; $k++ ) {
          #####
          print "K$k\n" if $verbose == 1;
          #####
          for ( $i = 0 ; $i < $lookup_number ; $i++ ) {

               # An option to include self-genome comparison could go here,
e.g. Genome A to Genome A
               # currently they are excluded...
               if ( $k != $i ) {

                    # This is where the main comparisons occur
                    # e.g. a-b, a-c & a-d
                    # uses $k and $i
                    #print
"\n$lookup_filenames[$k]\t$lookup_filenames[$i]\n";
                    $file_k = fileparse( $lookup_filenames[$k] );
                    $file_i = fileparse( $lookup_filenames[$i] );

                    #$file_k = $lookup_filenames[$k];
                    #$file_i = $lookup_filenames[$i];
                    #####
                    print "Processing\t$file_k\tand\t$file_i\n";
                    #####
                    &comparison( $lookup_filenames[$k],
$lookup_filenames[$i], $k, $i,
                                 $lookup_filenames[$k] );
               }
          }
     }
     &remove_single_recips;
}

sub comparison {

     @variables     = @_;
     $ifile         = $variables[0];
     $jfile         = $variables[1];
```

```
	$ivar          = $variables[2];
	$jvar          = $variables[3];
	$kfile         = $variables[4];
	$kfile         = fileparse($kfile);
	$ifile_compare = fileparse($ifile);

	open QUERY, "<$ifile";
	#####
	print "\tOpening: $ifile\n" if $verbose == 1;
	#####
	@query_array = ();
	while (<QUERY>) {
		my ($line) = $_;
		chomp($line);
		my ($line_number) = $.;
		@temp = split( /,/, $line );
		push( @query_array, "$temp[$jvar]" );
	}

	$query_array_length = @query_array;
	for ( $l = 0 ; $l < $query_array_length ; $l++ ) {
		if ( $query_array[$l] == "0" && $ifile_compare eq $kfile ) {
			$line_number = $l + 1;
			&get_accession( $ifile, $jfile, $line_number );
			$missing_genome = fileparse($jfile);
			open NOHITS, ">>$gene_hits_results/no_hits_$missing_genome";
			print NOHITS "$accession,\n";
		}
		elsif ( $query_array[$l] == "1" ) {

			$line_number = $l + 1;
			&get_accession( $ifile, $jfile, $line_number );
			#####
			print "Q$accession_array[0],$accession_array[1]" if $verbose
== 1;
			#####
			&get_hit_info( $ifile, $jfile, $line_number, "0" );
			#####
			print " hits
S\_$return_array[0],$return_array[1],$return_array[2],$return_array[3]\n"
			  if $verbose == 1;
			#####
			&scan_subject_initial( $accession_array[0], $ifile, $jfile,
$return_array[0], "0" );
			#####
			print "NStatus = $scan_array[0]\n\n" if $verbose == 1;
			#####

			if ( $scan_array[0] eq "recip" ) {

				open NON_DIFFERENTIAL,
">>$gene_hits_duplications/duplications_$kfile";


&get_line_number_start("$gene_hits_duplications/duplications_$kfile");

				print NON_DIFFERENTIAL

"$last_line_number;$accession_array[0];$accession_array[1];$return_array[0];$
return_array[1];$return_array[2];$return_array[3]\n";
			}
```

```
          }
          elsif ( $query_array[$l] >= "2" && $query_array[$l] <= $hit_limit )
{
               $verbose = 0;
               #####
               print "\# Query Hits = $query_array[$l]\n" if $verbose == 2;
               #####
               for ( $n = 0 ; $n < $query_array[$l] ; $n++ ) {

                    $line_number = $l + 1;
                    &get_accession( $ifile, $jfile, $line_number );
                    $query_number = $n + 1;
                    #####
                    print
"Q$query_number\_$accession_array[0],$accession_array[1]" if $verbose == 2;
                    #####
                    &get_hit_info( $ifile, $jfile, $line_number, $n );
                    #####
                    print " hits
S\_$return_array[0],$return_array[1],$return_array[2],$return_array[3]\n"
                      if $verbose == 1;
                    #####
                    &scan_subject_initial( $accession_array[0], $ifile,
$jfile, $return_array[0], $n );

                    #####
                    print "Status = $scan_array[0]\n\n" if $verbose == 2;
                    #####

                    if ( $scan_array[0] eq "recip" ) {

                         open DIFFERENTIALS,
">>$gene_hits_differentials/differentials_$kfile";


&get_line_number_start("$gene_hits_differentials/differentials_$kfile");
                         #####$last_line_number = $n;

                         #####print "LA =
$last_accession\tA$accession_array[0]\n";
                         if ( $last_accession eq $accession_array[0] ) {
                              $last_line_number = $last_line_number + 1;
                         }
                         elsif ( $last_accession ne $accession_array[0] ) {
                              $last_line_number = 0;
                         }

                         print DIFFERENTIALS

"$last_line_number;$accession_array[0];$accession_array[1];$return_array[0];$
return_array[1];$return_array[2];$return_array[3];$return_array[4];$return_ar
ray[5]\n";
                    }    #endif
               }    #endfor
          }    #endelsif

     }    #endfor
     print NON_DIFFERENTIAL "\n";

     #print DIFFERENTIALS "\n";
     #print "\n";
```

```
}

sub remove_single_recips {

	@diff_file_names = <$gene_hits_differentials/*.csv>;
	$num_files       = @diff_file_names;

	for ( $q = 0 ; $q < $num_files ; $q++ ) {
		print "\nRemoving singles from $diff_file_names[$q]\n";
		open DIFF, "<$diff_file_names[$q]";

		#($init_file,$dir,$ext) = fileparse($init_file, '\..*');
		( $file, $dir, $ext ) = fileparse( $diff_file_names[$q], '\..*' );
		@temp_array = ();
		while (<DIFF>) {
			my ($line) = $_;
			chomp($line);
			my ($line_number) = $.;

			#@temp = split( /;/, $line );
			#push( @temp_array, "@temp" );
			push( @temp_array, "$line" );
		}

		$temp_array_length = @temp_array;

		#print "XX = $temp_array_length = XX\n";
		for ( $r = 0 ; $r < $temp_array_length ; $r++ ) {

			@line_n  = split( /;/, $temp_array[$r] );
			@line_n1 = split( /;/, $temp_array[ $r + 1 ] );

			$ln_n  = $line_n[0];
			$ln_n1 = $line_n1[0];

			if ( $ln_n eq 0 && $ln_n1 eq 0 ) {

				#print "\t\tDiscarding @line_n\n";
			}
			else {
				open OUTPUT, ">>$dir$file\_fixed$ext";
				foreach $element (@line_n) {
					print OUTPUT $element . "\;";
				}

				#print OUTPUT "@line_n\n";
				print OUTPUT "\n";
			}
		}
		print "End\n";
		close(OUTPUT);
	}
}

sub fusion_scan {

	$verbose = 1;

	print "\nPerforming Fusion Scans\n";

	@fusion_filenames = <$gene_hits_differentials/*fixed.csv>;
```

```
	$fusion_number    = @fusion_filenames;

	print "Number of files = $fusion_number\n";

	for ( my $i = 0 ; $i < $fusion_number ; $i++ ) {

		undef(@fusion);
		open FUSION, "<$fusion_filenames[$i]";
		$input_file = fileparse( $fusion_filenames[$i] );
		print "Opening $fusion_filenames[$i]\n";

		while (<FUSION>) {
			my ($line) = $_;
			chomp($line);
			push( @fusion, "$line" );
		}

		$fusion_array_length = @fusion;

		for ( my $j = 0 ; $j < $fusion_array_length ; $j++ ) {

			@array_line = split( /;/, $fusion[$j] );
			$line_number = $array_line[0];

			if ( $line_number eq 0 ) {

				$query_accession = $array_line[1];
				$query_length    = $array_line[2];

				$subject_accession = $array_line[3];
				$subject_length    = $array_line[4];
				$subject_evalue    = $array_line[5];
				$subject_hit_range_start = $array_line[7];
				$subject_hit_range_end   = $array_line[8];
				$match_length = $subject_hit_range_end -
$subject_hit_range_start;
				push( @subject_hits,

"$subject_accession,$subject_hit_range_start,$subject_hit_range_end,$subject_
length"
				) if $match_length > 50 && $subject_length > 50;
				#####
				print "\nQ $query_accession -> $subject_accession\n" if
$verbose == 1;
				#####

				for ( my $k = $j + 1 ; $k < $fusion_array_length ; $k++ )
{

					@next_array_line = split( /;/, $fusion[$k] );
					$next_line_number = $next_array_line[0];

					if ( $next_line_number ne 0 ) {

						$next_query_accession = $next_array_line[1];

						$next_subject_accession       =
$next_array_line[3];
						$next_subject_length          =
$next_array_line[4];
```

```
                                $next_subject_evalue          =
$next_array_line[5];
                                $next_subject_hit_range_start =
$next_array_line[7];
                                $next_subject_hit_range_end   =
$next_array_line[8];
                                # > 50 should be user selectable limit for now
it is hard-coded
                                # to exclude all sequences with <=50 bases.
                                # Another limit is Matched Length (end - start)
this should also be
                                # more than 50.
                                $next_match_length =
$next_subject_hit_range_end - $next_subject_hit_range_start;

                                if ( $next_line_number ne 0 && $query_accession
eq $next_query_accession && $next_subject_length > 50 && $next_match_length >
50) {
                                        print "\n$next_line_number ne 0 &&
$query_accession eq $next_query_accession && $next_subject_length > 50 &&
$next_match_length > 50\n";
                                        push( @subject_hits,

"$next_subject_accession,$next_subject_hit_range_start,$next_subject_hit_rang
e_end,$next_subject_length,$next_subject_evalue"
                                        );
                                }
                        }
                        else {
                           last;
                        }
                    }
                    &rank_sort;
                    undef(@subject_hits);
                }
            }
            close(FUSION);
        }
}

sub rank_sort {
     $subject_hit_length = @subject_hits;

     # Get fusion ORF length and half
     $fusionORF = $query_length;
     $half_fusion_length = sprintf( "%.0f", $query_length / 2 );
     print "Fusion Length - $fusionORF / 2 = $half_fusion_length\n";

     for ( $q = 0 ; $q < $subject_hit_length ; $q++ ) {
          @unfused = split( /,/, $subject_hits[$q] );

          #print "Q = $q\n";
          if ( $unfused[1] <= $half_fusion_length ) {
               $pos1 = "L";
          }
          elsif ( $unfused[1] >= $half_fusion_length ) {
               $pos1 = "R";
          }
          if ( $unfused[2] <= $half_fusion_length ) {
               $pos2 = "L";
          }
```

```
        elsif ( $unfused[2] >= $half_fusion_length ) {
            $pos2 = "R";
        }

        #print "$pos1 - $unfused[1] , $unfused[2] - $pos2\n";
        if ( "$pos1$pos2" eq "LL" ) {

            #print "$pos1$pos2\t$unfused[0] ORF is Leftmost\n";
            push( @leftmost, "$subject_hits[$q]" );
        }
        elsif ( "$pos1$pos2" eq "RR" ) {

            #print "$pos1$pos2\t$unfused[0] ORF is Rightmost\n";
            push( @rightmost, "$subject_hits[$q]" );
        }
        elsif ( "$pos1$pos2" eq "LR" ) {

            #print "$pos1$pos2\t$unfused[0] ORF is more left\n";
            push( @middles, "$subject_hits[$q]" );
        }
        elsif ( "$pos1$pos2" eq "RL" ) {

            #print "$pos1$pos2\t$unfused[0] ORF is more right\n";
            push( @middles, "$subject_hits[$q]" );
        }
    }

    $left_num   = @leftmost;
    $right_num  = @rightmost;
    $middle_num = @middles;
    print "L:$left_num\tR:$right_num\tM:$middle_num\n";

    if ( $left_num == 0 && $right_num == 0 && $middle_num == 0 ) {

        # Discard
    }
    elsif (    $left_num == 0 && $right_num == 0
            or $right_num == 0 && $middle_num == 0
            or $left_num == 0  && $middle_num == 0 ) {

        # Discard
    }
    elsif ( $left_num == 0 && $right_num >= 1 and $middle_num >= 1) {

        &ignore_orthologues;
        print "\t\tXX $continue XX\n";
        if ($continue eq "yes") {
            &middle_and_right;
        }
    }
    elsif ( $right_num == 0 && $left_num >= 1 and $middle_num >= 1) {

        &ignore_orthologues;
        print "\t\tXX $continue XX\n";
        if ($continue eq "yes") {
            &middle_and_left;
        }

    }
    elsif ( $middle_num == 0 && $left_num >= 1 and $right_num >= 1) {
```

```
            print "011\n";
            &left_and_right;

      }
      elsif ( $left_num >= 1 && $right_num >= 1 and $middle_num >= 1 ) {

            #print "\nXX - All - XX\n";
            &ignore_orthologues;
            print "\t\tXX $continue XX\n";
            if ($continue eq "yes") {
                  print "111\n";
                  &left_and_right;
                  &middle_and_left;
                  &middle_and_right;
               }
      }
      undef(@middles);
      undef(@leftmost);
      undef(@rightmost);
}

sub left_and_right {
      print "\nLR\n";
      for ( my $a = 0 ; $a < $left_num ; $a++ ) {
            print "A:$a\n";
            @unfused_one = split( /,/, $leftmost[$a] );
            my $one_end = $unfused_one[2];

            for ( my $b = 0 ; $b < $right_num ; $b++ ) {
                  print "\tB:$b\n";
                  @unfused_two = split( /,/, $rightmost[$b] );
                  my $two_start = $unfused_two[1];

                  print "\t$query_accession -> $unfused_one[0] +
$unfused_two[0]\n";

                  $ratio = $one_end / $two_start;
                  $ratio = sprintf( "%.2f", $ratio );
                  print "\tRlr: $one_end / $two_start = $ratio\n";

                  if ( $ratio >= $lower_ratio && $ratio <= $higher_ratio ) {
                        undef(@subject_hits);
                        push( @subject_hits,
"$unfused_one[0],$unfused_one[1],$unfused_one[2],$unfused_one[3]" );
                        push( @subject_hits,
"$unfused_two[0],$unfused_two[1],$unfused_two[2],$unfused_two[3]" );
                        $subject_hit_length = @subject_hits;
                        &generate_image;
                  }
            }
      }
}

sub ignore_orthologues {

print "Ignoring Potential Orthologues\n";
        $continue = "yes";

        for ( my $a = 0 ; $a < $middle_num ; $a++ ) {
                @unfused_middles = split( /,/, $middles[$a] );
                my $middle_length = $unfused_middles[3];
```

```
                print "\t$query_accession - $middle_evalue\n";
                $length_ratio = $middle_length / $query_length;

                if ($length_ratio <= "0.95") {

                        #$continue = "yes";
                        push (@cont, "yes");
                        print "\t\t$unfused_middles[0] - $middle_length /
$query_length = $length_ratio - yes\n";
                }
                elsif ($length_ratio > "0.95") {

                        push (@cont, "no");
                        print "\t\t$unfused_middles[0] - $middle_length /
$query_length = $length_ratio - no\n";
                }

        }
        foreach $item (@cont) {
                print $item . ",";
        }
        print "\n";

        $continue = "no" if (grep /^no$/, @cont);

        undef(@cont);
        return $continue;
}

sub middle_and_left {
     print "\nML\n";
     for ( my $a = 0 ; $a < $left_num ; $a++ ) {
          print "A:$a\n";
          @unfused_one = split( /,/, $leftmost[$a] );

          my $one_end = $unfused_one[2];

          for ( my $b = 0 ; $b < $middle_num ; $b++ ) {
               print "\tB:$b\n";
               @unfused_two = split( /,/, $middles[$b] );

               my $two_start = $unfused_two[1];
               $two_match_length = $unfused_two[2] - $unfused_two[1];
               print "$query_accession -> $unfused_one[0] +
$unfused_two[0]\n";

               $ratio = $one_end / $two_start;
               $ratio = sprintf( "%.2f", $ratio );

               if ( $two_start <= $one_end ) {
                    print "\t$two_start <= $one_end Overlap!\n";
               }
               else {
                    print "\tRml: $one_end / $two_start = $ratio\n";
                    if ( $ratio >= $lower_ratio && $ratio <= $higher_ratio )
{
                         undef(@subject_hits);
                         push( @subject_hits,

"$unfused_one[0],$unfused_one[1],$unfused_one[2],$unfused_one[3]" );
                         push( @subject_hits,
```

```
"$unfused_two[0],$unfused_two[1],$unfused_two[2],$unfused_two[3]" );
                        $subject_hit_length = @subject_hits;
                        &generate_image;
                    }
                }
            }
        }
}

sub middle_and_right {
     print "\nMR\n";
     for ( my $a = 0 ; $a < $middle_num ; $a++ ) {
          print "A:$a\n";
          @unfused_one = split( /,/, $middles[$a] );

          my $one_end = $unfused_one[2];

          for ( my $b = 0 ; $b < $right_num ; $b++ ) {
               print "\tB:$a\n";
               @unfused_two = split( /,/, $rightmost[$b] );

               my $two_start = $unfused_two[1];
               $two_match_length = $unfused_two[2] - $unfused_two[1];
               print "$query_accession -> $unfused_one[0] +
$unfused_two[0]\n";

               $ratio = $one_end / $two_start;
               $ratio = sprintf( "%.2f", $ratio );

               if ( $one_end >= $two_start ) {
                    print "\t$one_end >= $two_start Overlap!\n";
               }
               else {
                    print "\tRmr: $one_end / $two_start = $ratio\n";
                    if ( $ratio >= $lower_ratio && $ratio <= $higher_ratio )
{
                        undef(@subject_hits);
                        push( @subject_hits,

"$unfused_one[0],$unfused_one[1],$unfused_one[2],$unfused_one[3]" );
                        push( @subject_hits,

"$unfused_two[0],$unfused_two[1],$unfused_two[2],$unfused_two[3]" );
                        $subject_hit_length = @subject_hits;
                        &generate_image;
                    }
               }
          }
     }
}

sub rank_sort_old {
     $subject_hit_length = @subject_hits;
     if ( $subject_hit_length != 0 ) {
          $lowest          = 0;
          $highest         = 0;
          $lowest_element  = 0;
          $highest_element = 0;
          for ( $q = 0 ; $q < $subject_hit_length ; $q++ ) {
```

```
                @temp = split( /,/, $subject_hits[$q] );

                $start = $temp[1];
                $end   = $temp[2];

                # This is for the first case where $lowest = 0 and
                # so will always be lower as no value has been assigned
                # to it....
                if ( $q == 0 ) {
                     $lowest          = $temp[2];
                     $highest         = $temp[1];
                     $lowest_element  = $q;
                     $highest_element = $q;
                }

                #####print "S=$start\tE=$end\tL=$lowest\tH=$highest\n";

                if ( $lowest >= $end ) {
                     $lowest         = $end;
                     $lowest_element = $q;
                }

                if ( $highest <= $start ) {
                     $highest         = $start;
                     $highest_element = $q;
                }

                #####print
"L=$lowest\tH=$highest\tLE=$lowest_element\tHE=$highest_element\n";
           }

           # Here I need to re populate $subject_hits with only
           # the lowest and highest element groups...
           # if the lowest / highest score is greater than user input
           # or user input along with the match for lowest_element and
highest_element
           # this of course limits fusions to those between only 2 genes...

           $ratio = $lowest / $highest;
           $ratio = sprintf( "%.1f", $ratio );
           if ( $ratio >= $lower_ratio && $ratio <= $higher_ratio ) {

                if ( $lowest_element != $highest_element ) {
                     $temp1 = $subject_hits[$lowest_element];
                     $temp2 = $subject_hits[$highest_element];
                     undef(@subject_hits);
                     push( @subject_hits, $temp1 );
                     push( @subject_hits, $temp2 );
                     $subject_hit_length = @subject_hits;

                     # Output text of hits here...
                     print
"$j,$query_accession,$subject_hits[0],$subject_hits[1]\n" if $verbose == 1;
                }
                &generate_image;
           }
      }
}

## These could be call to print_log with a filename variable...
sub print_comp_list {
```

```
	my $message = shift;
	open LOG, ">>$run_directory/composite_list.csv";
	print LOG "$message";
	close(LOG);
}

sub print_split_list {
	my $message = shift;
	open LOG, ">>$run_directory/split_list.csv";
	print LOG "$message";
	close(LOG);
}

sub read_domain_file {
	print "Name of all-domains file?\n>:";
	chomp( $domain_in = <STDIN> );
	open DOMAINS, "$run_directory\/$domain_in";
	while (<DOMAINS>) {
		$line = $_;
		push (@domains, $line);
	}
	close(DOMAINS);
}

sub generate_image {

	$query_acc_size = ( length $query_accession );
	push( @names_length, $query_acc_size );

	$number_of_hits = $subject_hit_length;

	$unfused_one_length = ( length $unfused_one[0] );
	push( @names_length, $unfused_one_length );
	$unfused_two_length = ( length $unfused_two[0] );
	push( @names_length, $unfused_two_length );

	# Get longest gene accession and * 6 for gene accession length in
'pixels'
	my $highest;
	for (@names_length) {
		$highest = $_ if $_ > $highest;
	}
	$name_length = $highest * 6;

	# Image specific
	$padding_left  = 100;
	$padding_right = 100;
	$padding       = $padding_left + $padding_right;

	# place to start drawing sequences from
	$left_pos = $padding_left + $name_length;

	# Image Width(length?)
	$image_length = $query_length + $padding + $name_length;
	if ( $image_length < 500 ) {
		$image_length += 250;
	}

	# number of hits * 50 + 50 for query + 20 for padding
	$image_height = ( $number_of_hits * 50 ) + 50 + 20;
```

```
	# set positions
	$font_vpos      = 2;
	$bar_vpos       = 20;
	$accession_vpos = 15;
	$start          = 0;
	$end            = 0;

	# create a new image
	$im = new GD::Image( $image_length, $image_height, 1 ); # Not paletted -
allows for alpha transparency of domains

	# allocate some specific colors
	$white = $im->colorAllocate( 255, 255, 255 );
	$black = $im->colorAllocate( 0,   0,   0 );
	$red   = $im->colorAllocate( 255, 0,   0 );
	$blue  = $im->colorAllocate( 0,   21,  181 );

	# make the background non-transparent and interlaced
	$im->transparent(-1);    # no transparency
	$im->interlaced('true');
	$im->alphaBlending(1);

	# White background
	$im->filledRectangle(
	                      0,
	                      0,
	                      $image_length,
	                      $image_height,
	                      $white
	);
	# Put a black frame around the picture
	$im->rectangle( 0, 0, $image_length - 1, $image_height - 1, $black );

	$actual_length = "";
	&draw_query;
	&draw_subjects;
	&watermark;

	@unfused_one = split( /,/, $subject_hits[0] );
	@unfused_two = split( /,/, $subject_hits[1] );

	# We only really need 1dp for the folders, 2dp ratio is printed in
image...
	$ratio = sprintf( "%.1f", $ratio );

	$query_accession_dir = "$gene_hits_differentials/$query_accession";

	&print_comp_list("$query_accession,\n");
	&print_split_list("$unfused_one[0],\n$unfused_two[0],\n");

	if ( -e $query_accession_dir && -d $query_accession_dir ) {

	    $ratio_dir = "$query_accession_dir/$ratio";
	    if ( -e $ratio_dir && -d $ratio_dir ) {

	        open( OUT,
">$ratio_dir/$query_accession\_\_$unfused_one[0]\_\_$unfused_two[0].png" );
	        binmode OUT;
	        print OUT $im->png(9);
	        close(OUT);
	    }
```

```
            else {
                 mkdir( $ratio_dir, 0755 );
                 open( OUT,
">$ratio_dir/$query_accession\_\_$unfused_one[0]\_\_$unfused_two[0].png" );
                 binmode OUT;
                 print OUT $im->png(9);
                 close(OUT);
            }
       }
       else {

            mkdir( $query_accession_dir, 0755 );
            $ratio_dir = "$query_accession_dir/$ratio";
            if ( -e $ratio_dir && -d $ratio_dir ) {

                 open( OUT,
">$ratio_dir/$query_accession\_\_$unfused_one[0]\_\_$unfused_two[0].png" );
                 binmode OUT;
                 print OUT $im->png(9);
                 close(OUT);
            }
            else {
                 mkdir( $ratio_dir, 0755 );
                 open( OUT,
">$ratio_dir/$query_accession\_\_$unfused_one[0]\_\_$unfused_two[0].png" );
                 binmode OUT;
                 print OUT $im->png(9);
                 close(OUT);
            }
       }
}

sub draw_query {
       $bar_length = $query_length;
       $accession  = $query_accession;
       &scale_bars;

       #
       $end       = $query_length;
       $end_scale = $query_length;
       &length_bars;

       #
       &bar($blue);
       &accession_name($blue);
       &draw_domain;
}

sub draw_subjects {

       for ( $c = 0 ; $c < $number_of_hits ; $c++ ) {

            @temp          = split( /,/, $subject_hits[$c] );
            $left_pos      = $left_pos + $temp[1];
            $bar_vpos      = $bar_vpos + 50;
            $bar_length    = ( $temp[2] - $temp[1] );
            $actual_length = $temp[3];
            $font_vpos     = $font_vpos + 50;

            #
            $colour_ratio = $bar_length / $actual_length;
```

```
            $colour_ratio = sprintf( "%.2f", $colour_ratio );

            &get_colour;

            #
            &bar($ratio_colour);

            #
            $start     = $temp[1];
            $end       = $bar_length;
            $end_scale = $end + $start;
            &scale_bars;

            #
            &length_bars;
            #
            $accession      = $temp[0];
            $left_pos       = $left_pos - $temp[1];
            ##$left_pos = $padding_left + $name_length;
            $accession_vpos = $accession_vpos + 50;
            &accession_name($ratio_colour);
            #if ($c = "0") {
            #     &draw_domain;
            #}
            #elsif ($c = "1") {
            #     #$left_pos = $padding_left + $name_length;
            &draw_domain;
            #}

      }
}

sub draw_domain {

      for ( my $i = 0 ; $i <= $#domains; $i++ ) {

            my @temp = split( /\t/, $domains[$i] );

            if ( $accession eq $temp[0] ) {
                  print "$accession = $temp[0]\n";

                  my @temp2 = split( /\~/, $temp[2] );
                  for ( my $j = 0 ; $j <= $#temp2; $j++ ) {

                        $var = $j * 3 + 3;

                        $domain = $temp2[$j];
                        $domain_start = $temp[$var];
                        $domain_end   = $temp[ $var + 1 ];
                        $domain_type  = $temp[ $var + 2 ];
                        chomp($domain_type); # Remove end of line, causing last
domain to be ignored.

                        print "\t$#temp2 - $j -
$domain\t$domain_start\t$domain_end\t$domain_type\n";

                        &domain_colour($domain);
                        if ( $domain_type eq ".." ) {
                              &domain_start_open;
                              &domain_middle;
                              &domain_end_open;
```

```
                    }
                    elsif ( $domain_type eq "[." ) {
                         &domain_start_closed;
                         &domain_middle;
                         &domain_end_open;
                    }
                    elsif ( $domain_type eq ".]" ) {
                         &domain_start_open;
                         &domain_middle;
                         &domain_end_closed;
                    }
                    elsif ( $domain_type eq "[]" ) {
                         &domain_start_closed;
                         &domain_middle;
                         &domain_end_closed;
                    }
               }
          }
     }
}


sub domain_end_open {

     # make a polygon
     $hpos = $left_pos + $domain_end - 12;

     $poly = new GD::Polygon;
     $poly->addPt( $hpos,     $bar_vpos);
     $poly->addPt( $hpos + 4, $bar_vpos - 4 );
     $poly->addPt( $hpos,     $bar_vpos - 4 );
     $poly->addPt( $hpos,     $bar_vpos);
     $poly->addPt( $hpos,     $bar_vpos + 8 );
     $poly->addPt( $hpos + 4, $bar_vpos + 4 );
     $poly->addPt( $hpos,     $bar_vpos);

     # draw the polygon, filling it with a color
     $im->filledPolygon( $poly, $dom_colour );
}

sub domain_start_open {

     # make a polygon
     $hpos = $left_pos + $domain_start + 9;

     $poly = new GD::Polygon;
     $poly->addPt( $hpos + 4, $bar_vpos - 4 );    #40
     $poly->addPt( $hpos,     $bar_vpos - 4 );    #00
     $poly->addPt( $hpos + 4, $bar_vpos - 4 );    #44
     $poly->addPt( $hpos,     $bar_vpos );        #08
     $poly->addPt( $hpos + 4, $bar_vpos + 4 );    #412
     $poly->addPt( $hpos,     $bar_vpos + 8 );    #016
     $poly->addPt( $hpos + 4, $bar_vpos + 8 );    #416

     # draw the polygon, filling it with a color
     $im->filledPolygon( $poly, $dom_colour );

}

sub domain_end_closed {
```

```
     $im->setAntiAliased($dom_colour);
     $im->filledArc( $left_pos + $domain_end - 13, $bar_vpos + 2, 13, 13,
270, 90, $dom_colour, gdArc );
}

sub domain_start_closed {

     $im->filledArc( $left_pos + $domain_start + 14, $bar_vpos + 2, 13, 13,
90, 270, $dom_colour, gdArc );
     $im->setAntiAliased($dom_colour);
}

sub domain_middle {
     $im->filledRectangle( $left_pos + $domain_start + 14,
                           $bar_vpos - 4,
                           $left_pos + $domain_end - 13,
                           $bar_vpos + 8, $dom_colour ); # + 14/-13 for caps
     $domain_name_length = length ($domain) * 3;
     $dark_grey  = $im->colorAllocate( 83,  83,  83 );
     if ( $domain_name_length < ($domain_end - $domain_start) ) {
     $im->string( gdSmallFont, $left_pos + $domain_start + ((($domain_end -
$domain_start) / 2) - $domain_name_length) - 1, $bar_vpos - 4 - 1, "$domain",
$dark_grey ); # shadow
     $im->string( gdSmallFont, $left_pos + $domain_start + ((($domain_end -
$domain_start) / 2) - $domain_name_length), $bar_vpos - 4, "$domain", $white
);
     }
}

sub domain_colour {
          my $temp = shift;
          # Add the ascii values of the domain text to generate
          # colours specific to each name...
          my @domain = unpack ("C*", "$temp");
          $sum = eval join '+', @domain;
          $sum = $sum / 2;

          $r = int ((($sum * 10) - 40) - 240);
          $g = int ((($sum * 10) - 40) - 120);
          $b = int ((($sum * 10) - 40) - 80);
          $a = "45";

          #$dom_colour = $im->colorAllocate( $r, $g, $b );
          $dom_colour = $im->colorAllocateAlpha( $r, $g, $b, $a );

        return $dom_colour;
}

sub scale_bars {
     $scale_quotient = int( $bar_length / 100 );

     #Large Scale Bars
     for ( $a = 0 ; $a <= $scale_quotient ; $a++ ) {
          $scale = $a * 100;
          $im->string( gdSmallFont, $scale + $left_pos, $font_vpos, "$scale",
$black );
          $im->rectangle( $scale + $left_pos, $bar_vpos - 5, $scale +
$left_pos + 1, $bar_vpos - 1, $black );
     }
}
```

```
sub length_bars {

     $end_padded = $end + $left_pos;

     # Start
     $im->string( gdSmallFont, $left_pos, $font_vpos + 30, "$start", $red );
     $im->rectangle( $left_pos, $bar_vpos + 6, $left_pos + 1, $bar_vpos + 10,
$red );

     #End
     $im->string( gdSmallFont, $end_padded, $font_vpos + 30, "$end_scale",
$red );
     $im->rectangle( $end_padded - 1, $bar_vpos + 6, $end_padded, $bar_vpos +
10, $red );

     #Length End
     #$im->string( gdSmallFont, $end_padded, $font_vpos, "$bar_length",
$black );
     #$im->rectangle( $end_padded - 1, $bar_vpos - 5, $end_padded, $bar_vpos
- 1, $black );

}

sub bar {
     $colour = shift;
     $im->filledRectangle( $left_pos, $bar_vpos, $bar_length + $left_pos,
$bar_vpos + 5, $colour );
}

sub accession_name {
     my $colour = shift;
     if ( $actual_length == "" ) {
          $im->string( gdSmallFont, $padding_left - 50, $accession_vpos,
"$accession",    $colour );
          $im->string( gdTinyFont,  $padding_left - 50, $accession_vpos + 12,
"L=$bar_length", $colour );
     }
     else {
          $im->string( gdSmallFont, $padding_left - 50, $accession_vpos,
"$accession",        $colour );
          $im->string( gdTinyFont,  $padding_left - 50, $accession_vpos + 20,
"ML=$bar_length",    $colour );
          $im->string( gdTinyFont,  $padding_left - 50, $accession_vpos + 12,
"L =$actual_length", $colour );
     }
}

sub watermark {

     $dark_grey  = $im->colorAllocate( 83,  83,  83 );
     $light_grey = $im->colorAllocate( 192, 192, 192 );
     $dark_red   = $im->colorAllocate( 146, 84,  83 );
     $dark_green = $im->colorAllocate( 129, 158, 107 );

     $darker_grey  = $im->colorAllocate( 52,  50,  51 );
     $darker_red   = $im->colorAllocate( 123, 59,  59 );
     $purple       = $im->colorAllocate( 96,  84,  112 );
     $dark_blue    = $im->colorAllocate( 33,  135, 204 );
     $darker_green = $im->colorAllocate( 43,  166, 22 );
     $pink         = $im->colorAllocate( 255, 0,   255 );
```

```
	$logo_left_x = $image_length - $padding_right;
	$logo_left_y = $image_height - 20;

	$fdfBLAST = "fdfBLAST v$VERSION";

	$im->string( gdSmallFont, $image_length - 100, $logo_left_y + 7,
"$fdfBLAST", $light_grey );

	#$im->filledRectangle(
	#			$image_length - 10,
	#			$logo_left_y + 1,
	#			$image_length - 3,
	#			$logo_left_y + 8,
	#			$dark_grey
	#);
	#$im->filledRectangle(
	#			$image_length - 19,
	#			$logo_left_y + 10,
	#			$image_length - 12,
	#			$logo_left_y + 17,
	#			$dark_grey
	#);

	#$im->filledRectangle(
	#			$image_length - 19,
	#			$logo_left_y + 1,
	#			$image_length - 12,
	#			$logo_left_y + 8,
	#			$dark_green
	#);
	#$im->filledRectangle(
	#			$image_length - 10,
	#			$logo_left_y + 10,
	#			$image_length - 3,
	#			$logo_left_y + 17,
	#			$dark_red
	#);

	#$im->string( gdTinyFont, $image_length - 17, $logo_left_y + 1,  "C",
$light_grey );
	#$im->string( gdTinyFont, $image_length - 8,  $logo_left_y + 1,  "E",
$light_grey );
	#$im->string( gdTinyFont, $image_length - 17, $logo_left_y + 10, "E",
$light_grey );
	#$im->string( gdTinyFont, $image_length - 8,  $logo_left_y + 10, "M",
$light_grey );

	$im->string( gdSmallFont, $image_length - 180, $logo_left_y + 7, "Ratio
= $ratio", $light_grey );

	$im->string( gdTinyFont, $image_length - 425, $logo_left_y + 10, "Length
Match %", $light_grey );

	$im->filledRectangle(
			$image_length - 350,
			$logo_left_y + 10,
			$image_length - 320,
			$logo_left_y + 17,
			$darker_grey
	);
	$im->filledRectangle(
```

```
                        $image_length - 320,
                        $logo_left_y + 10,
                        $image_length - 290,
                        $logo_left_y + 17,
                        $darker_red
    );
    $im->filledRectangle(
                        $image_length - 290,
                        $logo_left_y + 10,
                        $image_length - 260,
                        $logo_left_y + 17,
                        $purple
    );
    $im->filledRectangle(
                        $image_length - 260,
                        $logo_left_y + 10,
                        $image_length - 230,
                        $logo_left_y + 17,
                        $dark_blue
    );
    $im->filledRectangle(
                        $image_length - 230,
                        $logo_left_y + 10,
                        $image_length - 200,
                        $logo_left_y + 17,
                        $darker_green
    );
    $im->filledRectangle( $image_length - 200,
                        $logo_left_y + 10,
                        $image_length - 190,
                        $logo_left_y + 17, $pink );

    $im->string( gdTinyFont, $image_length - 343, $logo_left_y + 10, "<40",
$light_grey );
    $im->string( gdTinyFont, $image_length - 317, $logo_left_y + 10, "40-
60",  $light_grey );
    $im->string( gdTinyFont, $image_length - 287, $logo_left_y + 10, "60-
70",  $light_grey );
    $im->string( gdTinyFont, $image_length - 257, $logo_left_y + 10, "70-
80",  $light_grey );
    $im->string( gdTinyFont, $image_length - 229, $logo_left_y + 10, "80-
100", $light_grey );
    $im->string( gdTinyFont, $image_length - 197, $logo_left_y + 10, "!",
$light_grey );
}

sub get_colour {

    if ( $colour_ratio le "0.4" ) {

        # Black
        $r = int( 42 - ( 40 - ( 10 * ( 10 * 1 ) ) ) );
        $g = int( 40 - ( 40 - ( 10 * ( 10 * 1 ) ) ) );
        $b = int( 41 - ( 40 - ( 10 * ( 10 * 1 ) ) ) );

        #print "Black $r, $g, $b\n";
        $ratio_colour = $im->colorAllocate( $r, $g, $b );
    }
    elsif ( $colour_ratio gt "0.4" && $colour_ratio le "0.6" ) {

        # Red
```

```
            $r = int( 123 - ( 60 - ( 10 * ( 10 * 1 ) ) ) );
            $g = int( 59 -  ( 60 - ( 10 * ( 10 * 1 ) ) ) );
            $b = int( 59 -  ( 60 - ( 10 * ( 10 * 1 ) ) ) );

            #print "Red $r, $g, $b\n";
            $ratio_colour = $im->colorAllocate( $r, $g, $b );
        }
        elsif ( $colour_ratio gt "0.6" && $colour_ratio le "0.7" ) {

            # Purple
            $r = int( 96 -  ( 70 - ( 10 * ( 10 * 1 ) ) ) );
            $g = int( 84 -  ( 70 - ( 10 * ( 10 * 1 ) ) ) );
            $b = int( 112 - ( 70 - ( 10 * ( 10 * 1 ) ) ) );

            #print "Purple $r, $g, $b\n";
            $ratio_colour = $im->colorAllocate( $r, $g, $b );
        }
        elsif ( $colour_ratio gt "0.7" && $colour_ratio le "0.8" ) {

            # Blue
            $r = int( 33 -  ( 80 - ( 10 * ( 10 * 1 ) ) ) );
            $g = int( 135 - ( 80 - ( 10 * ( 10 * 1 ) ) ) );
            $b = int( 204 - ( 80 - ( 10 * ( 10 * 1 ) ) ) );

            #print "Blue $r, $g, $b\n";
            $ratio_colour = $im->colorAllocate( $r, $g, $b );
        }
        elsif ( $colour_ratio gt "0.8" && $colour_ratio le "1.0" ) {

            # Green
            $r = int( 43 -  ( 100 - ( 10 * ( 10 * 1 ) ) ) );
            $g = int( 166 - ( 100 - ( 10 * ( 10 * 1 ) ) ) );
            $b = int( 22 -  ( 100 - ( 10 * ( 10 * 1 ) ) ) );

            #print "Green $r, $g, $b\n";
            $ratio_colour = $im->colorAllocate( $r, $g, $b );
        }
        else {

            # Bright Pink
            $ratio_colour = $im->colorAllocate( 255, 0, 255 );
        }
        return $ratio_colour;
}

sub duplication_scan {

        $verbose = 0;
        #####
        print "\nPerforming Duplication Scans\n";
        #####

        @duplication_filenames = <$gene_hits_duplications/*fixed.csv>;
        $duplication_number    = @duplication_filenames;

        for ( $i = 0 ; $i < $duplication_number ; $i++ ) {
            open DUPLICATION, "<$duplication_filenames[$i]";
            #####
            #print "Opening $duplication_filenames[$i]...\n" if $verbose == 1;
            #####
            $input_file = fileparse( $duplication_filenames[$i] );
```

```
print "$input_file";
$pass = 0;
while (<DUPLICATION>) {
     my ($line)        = $_;
     my ($line_number) = $.;

     #print "\nLN\:$line_number\n";
     chomp($line);

     ##$pass = 0;
     if ( $line ne "" ) {

          push( @duplication_array, "$line" );
     }
     elsif ( $line eq "" && $pass == 0 ) {
          $duplicated_genes = $line_number;
          $pass++;
     }
}
close(DUPLICATION);
$duplication_array_length = @duplication_array;

for ( $k = 0 ; $k < $duplicated_genes - 1 ; $k++ ) {
     $count        = 0;
     @temp         = split( /\;/, $duplication_array[$k] );
     $query_gene   = $temp[1];
     $query_length = $temp[2];
     #####
     print "$k\t$query_gene\t" if $verbose == 1;
     #####
     for ( $j = 0 ; $j < $duplication_array_length ; $j++ ) {
          @temp2          = split( /\;/, $duplication_array[$j] );
          $subject_gene   = $temp2[3];
          $subject_length = $temp2[4];
          $subject_evalue = $temp[5];
          $subject_ident  = $temp[6];
          $query_count    = $temp2[1];
          #####
          print "$subject_gene\n" if $verbose == 1;
          #####
          if ( $query_gene eq $query_count ) {
               $count++;
               push( @subject,
"$subject_gene,$subject_length,$subject_evalue,$subject_ident," );
          }
     }
     $output_file = fileparse( $duplication_filenames[$i] );
     #####
     #print "Output to
$gene_hits_duplications/$count\_hits_$output_file\n" if $verbose == 1;
     #####
     $hits = $count + 1;
     open OUTPUT,
">>$gene_hits_duplications/$hits\_hits_$output_file";
     print OUTPUT "$query_gene,$query_length,";
     foreach $value (@subject) {
          print OUTPUT "$value";
     }
     print OUTPUT "\n";
     #####
     print "Hits = $count\n" if $verbose == 1;
```

```
                #####
                @subject = ();
                print ".";
           }

           @duplication_array = ();
           print "\n";
           close(OUTPUT);
      }
}

sub scan_subject_initial {
      $verbose       = 0;
      @scan_array    = ();
      @variables     = @_;
      $query_hits    = $variables[0];
      $init_file     = $variables[1];
      $init_file_two = $variables[2];

      #####$init_file =~ m/(.*?\/lookup\/)(.*?\..*?)(\.csv)/;
      #####$init_file = $2;
      ( $init_file, $dir, $ext ) = fileparse( $init_file, '\..*' );
      #####$init_file_two =~ m/(.*?\/lookup\/)(.*?\..*?)(\.csv)/;
      #####$init_file_two     = $2;
      ( $init_file_two, $dir, $ext ) = fileparse( $init_file_two, '\..*' );
      $subject_accession = $variables[3];
      $loop_position     = $variables[4];

      #####
      # print "Opening $gene_hits_initial/$init_file_two\_$init_file\.txt\n";
      #####

      open IN, "<$gene_hits_initial/$init_file_two\_$init_file\.csv";

      while (<IN>) {
           my ($line) = $_;
           chomp($line);
           @temp = split( /,/, $line );
           #####
           print "if $temp[1] == $subject_accession\n" if $verbose == 1;
           #####
           if ( $temp[1] eq $subject_accession ) {
                #####
                # print "$query_hits != $temp[0]\n";
                print "\# Reciprocal Hits = $temp[0]\n" if $verbose == 1;
                ######
                for ( $r = 0 ; $r < $temp[0] ; $r++ ) {
                     $loop_position = $r + 1;
                     $location      = ( 6 * $loop_position ) - 3;
                     #####
                     print "Loc = $location\tQH = $query_hits \<\-\>
$temp[$location]\n" if $verbose == 1;
                     #####
                     if ( $query_hits eq $temp[$location] ) {

                          push( @scan_array, "recip" );
                          push( @scan_array,
                                "$temp[(6 * $loop_position) - 3]",
                                "$temp[(6 * $loop_position) - 2]",
                                "$temp[(6 * $loop_position) - 1]",
                                "$temp[(6 * $loop_position)]",
```

```
                                        "$temp[(6 * $loop_position) + 1]",
                                        "$temp[(6 * $loop_position) + 2]" );
                        }
                        else {
                              push( @scan_array, "norecip" );

                        }
                  }
            }
      }
   return @scan_array;
}

sub get_line_number_start {

      # This method opens a file and gets the last
      # line number and returns the value
      # This will probably be exponentially slow. Hrmmm.
      $last_line_number = 0;

      @variables    = @_;
      $file_handle = $variables[0];

      open FILE, "<$file_handle";

      while (<FILE>) {

            #$last_line_number = $. if eof;
            #####$last_line        = $_ if eof;
            $last_accession = $_ if eof;
            @temp             = split( /;/, $last_accession );
            $last_accession   = $temp[1];
            $last_line_number = $temp[0];
      }
      close(FILE);
      #####chomp($last_line);
      ######$last_line_number = $last_line_number + 1;
   return "$last_line_number";
   return "$last_accession";
}

sub get_accession {
      $verbose = 0;

      # First we need to open the corresponding file in the initial dir
      # To do that we need the file name (no ext or dir) of the lookup file
      @variables          = @_;
      $init_file          = $variables[0];
      $init_file_two      = $variables[1];
      $lookup_line_number = $variables[2];

      #####$init_file =~ m/(.*?\/lookup\/)(.*?\..*?)(\.csv)/;
      #####$init_file = $2;
      ( $init_file, $dir, $ext ) = fileparse( $init_file, '\..*' );

      #####$init_file_two =~ m/(.*?\/lookup\/)(.*?\..*?)(\.csv)/;
      #####$init_file_two = $2;
      ( $init_file_two, $dir, $ext ) = fileparse( $init_file_two, '\..*' );

      # Set initial directory and open file
```

```
     open ACCESS, "<$gene_hits_initial\/$init_file\_$init_file_two\.csv";
     #####
     # print "Opening $gene_hits_initial\/$init_file\_$init_file_two\.txt\n"
if $verbose == 1;
     #####

     # While file, read each line and compare line numbers
     #
     while (<ACCESS>) {
          my ($line) = $_;
          chomp($line);
          my ($initial_line_number) = $.;

          # When line numbers match, assign accession and then return
          if ( $lookup_line_number == $initial_line_number ) {

               # Split the line on , and retrieve accession (pos 1)
               @temp             = split( /,/, $line );
               $accession        = $temp[1];
               $accession_length = $temp[2];
               @accession_array  = ( "$accession", "$accession_length" );
          }

          # Else do nothing
     }
     close(ACCESS);
   return $accession_array;
}

sub get_hit_info {

     $verbose      = 0;
     @return_array = ();

     # First we need to open the corresponding file in the initial dir
     # To do that we need the file name (no ext or dir) of the lookup file
     @variables          = @_;
     $init_file          = $variables[0];
     $init_file_two      = $variables[1];
     $lookup_line_number = $variables[2];
     $location           = $variables[3];
     $location           = $location + 1;

     #####$init_file =~ m/(.*?\/lookup\/)(.*?\..*?)(\.csv)/;
     #####$init_file = $2;
     ( $init_file, $dir, $ext ) = fileparse( $init_file, '\..*' );

     #####$init_file_two =~ m/(.*?\/lookup\/)(.*?\..*?)(\.csv)/;
     #####$init_file_two = $2;
     ( $init_file_two, $dir, $ext ) = fileparse( $init_file_two, '\..*' );

     #####
     # print "\nFile = $init_file\t\# = $lookup_line_number\tLoc =
$location\n" if $verbose == 1;
     #####

     # Set initial directory and open file

     open ACCESS, "<$gene_hits_initial\/$init_file\_$init_file_two\.csv";

     # While file, read each line and compare line numbers
```

```
     #
     while (<ACCESS>) {
          my ($line) = $_;
          chomp($line);
          my ($initial_line_number) = $.;

          # When line numbers match, assign accession and then return
          if ( $lookup_line_number == $initial_line_number ) {
               #####
               # print "The line is $line\n" if $verbose == 1;
               #####
               if ( $location == 1 ) {
                    #####
                    # print "One\t$line\n" if $verbose == 1;
                    #####
                    # Split the line on ',' and retrieve accession (pos 1)
                    @temp                   = split( /,/, $line );
                    $hit_accession          = $temp[3];
                    $length                 = $temp[4];
                    $evalue                 = $temp[5];
                    $percent_identity       = $temp[6];
                    $query_hit_range_start = $temp[7];
                    $query_hit_range_end    = $temp[8];
                    @return_array = (
                                        "$hit_accession",          "$length",
                                        "$evalue",
"$percent_identity",
                                        "$query_hit_range_start",
"$query_hit_range_end"
                    );
               }
               else {
                    #####
                    # print "Else\t$line\n" if $verbose == 1;
                    #####
                    $location               = ( $location * 6 );
                    @temp                   = split( /,/, $line );
                    $hit_accession          = $temp[ $location - 3 ];
                    $length                 = $temp[ $location - 2 ];
                    $evalue                 = $temp[ $location - 1 ];
                    $percent_identity       = $temp[$location];
                    $query_hit_range_start = $temp[ $location + 1 ];
                    $query_hit_range_end    = $temp[ $location + 2 ];
                    @return_array = (
                                        "$hit_accession",          "$length",
                                        "$evalue",
"$percent_identity",
                                        "$query_hit_range_start",
"$query_hit_range_end"
                    );
               }
          }
     }
     close(ACCESS);
   return @return_array;
}
```

10 Appendix: Putative Gene Fusions and Fissions as Predicted by fdfBLAST

Below are a set of tables detailing the putative differentially distributed gene fusion and fission events, as predicted by my program fdfBLAST; each table includes: the PFAM domains present in the putative gene fusion/fission event and the accession number for each ORF identified. Where cells are left empty, fdfBLAST did not predict a putative fusion event at the e-values selected (all tables reflect 1e-10), this may be because the gene fusion event is absent from that proteome or at the e-value selected it was not predicted; for example, a reciprocal hit may not be able to be produced. Cells that contain two accessions indicate the predicted split ORFs/domains identified by fdfBLAST, those with one represent a putative gene fusion event (whereby with phylogenetic analysis the SDC may be polarised). The fusion number column corresponds to the phylogeny (tree topology) numbers in the figures of Chapter 6 and the rows that are shaded green indicate the topologies included Chapter 6 are phylogenetically informative.

10.1 Extended Viridiplantae

Fusion #	PFAM Domain 1	PFAM Domain 2	*Arabidopsis thaliana*	*Oryza sativa*	*Physcomitrella patens*	*Selaginella moellendorffii*
1	PH	Oxysterol_BP	AT1G13170.1		Phys77864, Phys127103	
2	PMD	DUF716	AT1G321	LOC_Os03g36		

			20.1	760.1, LOC_Os11g40570.1		
3	TPP_enzyme_N and M and C	MR_MLE BAAT_C	AT1G68890.1	Phys118361, Phys118305		
4	Yae1_N	FTCD_N FTCD	AT2G20830.2	LOC_Os07g34610.1, LOC_Os06g03740.1	Phys146163, Phys60782	Sela843379, Sela128752
5	PTEN_C2 DUF2457	FH2	AT2G25050.1	LOC_Os07g40520.1, LOC_Os02g55150.1	Phys159530	Sela167047
6	Cu_amine_oxidN2 Cu_amine_oxidN3	Cu_amine_oxid	AT4G12290.1	LOC_Os04g20164.2, LOC_Os04g20164.1		
7	SufE	BolA	AT4G26500.1	LOC_Os09g09790.1	Phys89618, Phys38913	
8	ANF_receptor	(SBP_bac_3) Lig_chan	AT5G11210.1	LOC_Os06g08880.1, LOC_Os06g09090.1	Phys196025	
9	3_5_exonuc	DUF82	AT5G24340.1		Phys189851	Sela425875
10	Glyco_hydro_17	X8	AT5G56590.1		Phys25096, Phys8710	
11	DNA_pol_alpha_N	DNA_pol_B_exo DNA_pol_B zf-DNA-Pol	AT5G67100.1	LOC_Os01g64820.1	Phys94238, Phys60547	Sela232193
12	COX2_TM	COX2	ATMG00160.1			Sela388815, Sela139130
13	UPF0054	Hydrolase_3		LOC_Os01g53720.2	Phys77717, Phys77716	
14	B3 Auxin_resp	AUX_IAA		LOC_Os02g06910.1		Sela424114, Sela18617
15	B3	AUX_IAA		LOC_Os12g41		Sela1172

	Auxin_resp			950.1		17, Sela115320
16	PH	Oxysterol_BP		LOC_Os03g63074.2	Phys232824, Phys148099	
17	TIMELESS	TIMELESS_C		LOC_Os05g11980.1	Phys231531	Sela432638, Sela416994
18	Porin_3	Chorismate_bind	AT3G01280.1, AT1G74710.1	LOC_Os09g19734.1		
19	PP2C PP2C	Pkinase_Tyr	AT3G63340.1, AT3G63330.2	LOC_Os11g37540.1		
20	RF-1	Terpene_synth Terpene_synth_c	AT1G62850.2, AT5G48110.1			Sela406214
21	Response_reg	Myb_DNA-binding		LOC_Os04g28120.1, LOC_Os01g13740.2		Sela428906
22	Dynamin_N Dynamin_MGED	NdhO	AT2G14120.1, AT1G74880.1	LOC_Os04g31190.1, LOC_Os01g72950.1		Sela437242
23	eIF-3c_N PCI	Peptidase_C48		LOC_Os02g03080.1, LOC_Os11g01180.1		Sela437467
24	PNP_UDP_1	DNA_pol_delta_4		LOC_Os06g02210.1, LOC_Os09g34850.1		Sela441751

10.2 Fungi

Fusion #	PFAM Domain 1	PFAM Domain 2	*Neurospora crassa*	*Saccharomyces cerevisiae*	*Schizosaccharomyces pombe*	*Ustilago maydis*
1	PRA-CH PRA-PH	Histidinol_dh	NCU03139T0	1076975Sc	SPBC29A302c, SPBC171113	UM03974

2a	Epimerase	Aldose_epim	NCU05133T0, NCU08516T0	6319493Sc	SPBPB2B212c	UM06057, UM01772
2b	Epimerase	Aldose_epim	NCU04442T0, NCU09705T0	62738905Sc	SPBPB2B212c	
3	SurE	TTL		6319570Sc		UM05266, UM03366
4	Hydant_A_N Hydantoinase_A	Hydantoinase_B	NCU04569T0	6322634Sc		UM03902, UM00080
5	GATase IGPS	PRAI	NCU00200T0	173045Sc, 136348Sc	SPBC153909c	UM02376
6	DNA_ligase_A_N DNA_ligase_A_M	DNA_ligase_A_C BRCT BRCT	NCU06264T0	1151001Sc, 1151000Sc		
Fi1	eIF-3c_N	eIF-3c_N		984682Sc, 798951Sc	SPAC4A816c	UM06323
7	Hydrolase	CDP-OH_P_transf		6322923Sc, 6320059Sc	SPAC22A1208c	
8	COX15-CtaA	Fer2	NCU04817T0, NCU07794T0	6320989Sc, 73530106Sc	SPAC22E1210c	
9	Palm_thioest	PAP2	NCU06530T0, NCU03718T0		SPBC53012c	
10	Aconitase Aconitase_C	Ribosomal_L21p	NCU04280T0, NCU00977T0	6322261Sc, 37362666Sc	SPBP4H1015	
11	FSH1	DHFR_1	NCU08094T0, NCU11825T0	6324854Sc, 118999Sc	SPCC122308c	UM06281, UM03183
12	adh_short	PIG-F	NCU00302T0, NCU06663T0		SPCC145015	
13	Peptidase_M	IPPT	NCU0673	6324283Sc	SPCC132205c,	UM001

	1 Leik-A4-hydro-C		2T0, NCU10185T0	, 171963Sc	SPAC34315	70
14	ACT Formyl_trans_N	Pyr_redox_2 FMO-like	NCU09911T0, NCU03755T0			UM00231
15	Glyco_transf_22	XAP5	NCU07472T0, NCU07604T0	SPBC173412c, SPCC102012c		UM00341
16	DUF3448 AMP-binding	ADH_N ADH_zinc_N	NCU01417T0, NCU07042T0			UM00363
17	DUF3448 AMP-binding	ADH_N ADH_zinc_N	NCU01417T0, NCU07042T0			UM00363
18	Acyltransferase	Peptidase_M18 Peptidase_M42	NCU11282T0, NCU00122T0			UM00818
19	ATP-grasp-2	CoA_binding Ligase_CoA Citrate_synt x2	NCU06783T0, NCU06785T0		SPAC22A1216, SPBC170307	UM01005
20	DUF579	FAD_binding_4 FAD_oxidase_C	NCU09442T0, NCU00904T0	6325031Sc, 297922Sc		UM01069
21	Allantoicase Allantoicase	Ureidogly_hydro	NCU01816T0, NCU03850T0	666107Sc, 6322223Sc	SPAC1F709c, SPAC19G1204	UM01926
22	PGAM	Thymidylat _synt	NCU11201T0 NCU10053T0	6322896Sc 83578104 Sc		UM02407
23	DEAD_2	Biotin_lipoyl	NCU11409T0 NCU02704T0			UM03198

		E3_binding 2-oxoacid_dh				
24	Synaptobrevin	Ferrochelatase	NCU00566T0 NCU08291T0	6319289Sc 27065373Sc	SPAC6G911 SPCC32009	UM03444
25	BRAP2 zf-C3HC4 zf-UBP PHAGE_GP20	Aldo_ket_red	NCU11215T0 NCU08384T0			UM03993
26	DUF540	UreD	NCU01989T0 NCU01246T0			UM04139
27	TBC	Elongin_A			SPBC177809 SPBC29A308	UM04222
28	RRM_1	RPN7 PCI	NCU03226T0 NCU03972T0		SPAC23A109 SPBC58207c	UM04564
29	Nol1_Nop2_Fmu	Pribosyltran	NCU05301T0 NCU05290T0	6319447Sc 4757Sc	SPAC23C417 SPBC72515	UM05126
30	Acetyltransf_1	Bromodomain		5822444Sc 11513447Sc		UM05168
31	EBP	Glyco_transf_1	NCU00346T0 NCU06779T0			UM05209
32	RmlD_sub_bind	PX Vps5	NCU01177T0 NCU04137T0			UM05287
33	Amidase	Pkinase	NCU10409T0 NCU00978T0	3364Sc 999051Sc	SPCC55007 SPBC10601	UM05360

34	Cys_Met_Meta_PP	GHMP_kinase_N GHMP_kinase_C	NCU07987T0 NCU03633T0	6321254Sc 6323864Sc	SPCC11E1001 SPAC13G611c	UM05496
35	Spermine_synth	Saccharop_dh	NCU06727T0 NCU03748T0	6323175Sc 6324378Sc	SPBC12C207c SPBC3B803	UM05818
36	UPF0160	Argk	NCU03859T0 NCU01511T0		SPAC69404c SPCPB16A405c	UM06421

10.3 Deuterostomia

Fusion #	PFAM Domain 1	PFAM Domain 2	*Homo sapiens*	*Ciona intestinallis*	*Branchiostoma floridae*	*Strongylocentrotus purpuratus*
1	Zf-MYND	Peptidase_M24	NP_0559582Hs		59647Bf	XP_7914362Sp XP_7843822Sp
2	PGM_PMM_I PGM_PMM_II	PGM_PMM_IV		224588Ci	63910Bf	XP_7824662Sp XP_0011935341Sp
3	E1-E2_ATPase	Cation_ATPase_C	NP_0555533Hs		64297Bf	XP_0011830901Sp XP_7907252Sp
4	eIF3_N	PCI	NP_0015591Hs	230409Ci	114473Bf	XP_7823972Sp XP_0011923891Sp
5	MBOAT	SUI1	NP_0606642Hs NP_0036682Hs		121897Bf	
6	SH3	Guanylate_kin		399614Ci 393787Ci	123382Bf	XP_0011955941Sp
7	IBN_N	Cse1		294803	123574Bf	XP_0011962

		CAS_CSE1		Ci		011Sp XP_0011875 941Sp
8	ENTH	WRW	NP_6532801 Hs NP_0048801 Hs		126066Bf	
9	DWNN	eIF3g U-Hy??		277422 Ci 255082 Ci	199855Bf	
10	PUA	SUI1	NP_0088242 Hs		205375Bf	XP_7822142 Sp XP_0011813 851Sp
10	SH2 SH3 SH2 PH C2	RasGAP			206850Bf	XP_7818592 Sp XP_7905142 Sp
11	tRNA-synt_1_g	Anticodon_ 1	NP_6124041 Hs	286786 Ci	207903Bf	XP_7824912 Sp XP_7802991 Sp
11	STAT_int	STA_alpha STA_bind SH2		210623 Ci	233041Bf 68429Bf	
12	UCH	Exonuc_X-T	NP_0011209 321Hs	210807 Ci		XP_7905872 Sp XP_7922802 Sp
13	PDZ	PH PH		278836 Ci 272168 Ci	212007Bf	
14	Epimerase	DUF1731	NP_0645802 Hs	212042 Ci		XP_7976241 Sp XP_7891741 Sp
15	Peptidase_ M16 Peptidase_ M16_C	Peptidase_ M16_C	NP_0049602 Hs	213097 Ci		XP_7959752 Sp XP_7799642 Sp
16	Peptidase_ M24	SPT16 Rtt106	NP_0091231 Hs	213191 Ci		XP_7903201 Sp XP_7882981 Sp
17	HATPase_c DNA_mis_r	MutL_C		226297 Ci	69265Bf 69263Bf	

	epair					
Fi1	CTP_synth_N	CTP_Synth_N Peptidase_C26		228188Ci		XP_7911541Sp XP_7805972Sp
18	RasGEF_N	RasGEF				XP_0011844861Sp XP_7844452Sp
19	Drf_GBD Drf_FH3	FH2		263772Ci 253174Ci	230981Bf	XP_7850942Sp
20	Ski_Snn	c-SKI_SMAD_bind		266751Ci 268371Ci	231363Bf	
21	Exostosin	Glyco_transf_64	NP_0001182Hs	237549Ci	198850Bf 198883Bf	
22	RRM_1 20G-FeII_Oxy	Methytransf_12		241199Ci		XP_7975762Sp XP_7852242Sp
23	TRNA-synt_1 tRNA-synt_1	TRNA-synt_1 tRNA-synt_1g Anticodon_1	NP_0561551Hs	250065Ci		XP_7918541Sp XP_7935001Sp
24	Snase x4	TUDOR Snase		269813Ci	275079Bf 117294Bf	
25	MIF4G	MA3 W2	NP_0037512Hs	264801Ci 262656Ci	277494Bf	XP_0011925291Sp
26	CD20	CD20		285152Ci		XP_8007592Sp XP_8002101Sp
28	Glyco_hydro_64 Alpha-mann_mid	Glyco_hydro_38C		292293Ci	216531Bf 216484Bf	
29	GFO_IDH_MocA	GFO_IDH_MocA_C		292965Ci		XP_0011762881Sp XP_0011904191Sp
30	EHN	Abhydrolase_1	NP_0001111Hs			XP_7921651Sp

						XP_7929562 Sp
31	dsrm dsrm	A_deamin	NP_0011031 Hs		227842Bf 227803Bf	XP_7818321 Sp
32	Adaptin_N	Alpha_adaptinC2	NP_0011193 Hs	270145 Ci 277488 Ci		XP_7927732 Sp
33	DUF3452	RB_A RB_B	NP_0028862 Hs			XP_0011814 791Sp XP_0011940 301Sp
34	Rep-A_N tRNA_anti	Rep_fac-A_C	NP_0029361 Hs		89409Bf 89408Bf	
35	THF_DHG_CYH THF_DHG_CYH_C	FTHFS	NP_0059473 Hs			XP_7854042 Sp XP_7946542 Sp
36	Not3	NOT2_3_5	NP_0553311 Hs		118862Bf 68754Bf	XP_0012039 951Sp
37	La RRM_1	RRM_1	NP_0562691 Hs		271338Bf 130684Bf	
38	DOMON Cu2_monooxygen	Cu2_monoox_C	NP_0563442 Hs	275868 Ci 277622 Ci		
39	Dak1	Dak2	NP_0563482 Hs			XP_7812501 Sp XP_7871102 Sp
40	FtsJ DUF3381	Spb1_C	NP_0601173 Hs		279878Bf 121764Bf	XP_0011769 161Sp XP_7911782 Sp
41	tRNA-synt_1d	DALR_1	NP_0647162 Hs			XP_7921252 Sp XP_7811151 Sp
42	MoCE_biosynth	MoeA_N MoCE_biosynth MoeA_C	NP_0658571 Hs		202600Bf 202591Bf	
43	Macro	CRAL_TRIO	NP_0011290 611Hs	296424 Ci 266122 Ci		
44	TIG PSI TIG	Plexin_cytopl			90981Bf 60237Bf	XP_7856982 Sp
45	VHS	GAT			68113Bf	XP_7867802

		Alpha_adaptinC2			57346Bf	Sp
46	Glyco_hydro_1	Zona_pellucida		246935 Ci 286397 Ci		XP_7871052 Sp
47	CSD	Arf ArgK			197924Bf 200709Bf	XP_7884222 Sp
48	DUF1352	SIMPL		379864 Ci 228290 Ci		XP_0011763 621Sp
49	Lipin_N	LNS2		212084 Ci		XP_0011773 301Sp
50	Sin_N	Cluap		380222 Ci 232059 Ci		XP_0011864 471Sp
51	SH3_1 MIP-T3	DUF2013			66473Bf 203537Bf	XP_0011889 211Sp
51	Glyco_hydro_31	Glyco_hydro_31		250973 Ci 271363 Ci		XP_0011907 651Sp

10.4 Vertebrata

Fusion #	**PFAM Domain 1**	**PFAM Domain 2**	***Homo sapiens***	***Gallus gallus***	***Mus musclus***	***Danio rerio***
1	Ion_trans	KCNQ_channel KCNQC3-Ank-G_bd	NP_0045101		NP_6908871	NP_0011035101 XP_0019217381
2	C2 C2	RasGAP PH BTK	NP_0046491		NP_0388602	XP_6995663 XP_0019207371
3	ABC_membrane_2	ABC_tran	NP_0051551	XP_4159382	NP_0361242	XP_7005563 XP_0019227471
4	SPC7 DUF342	Kinesin	NP_0055413	XP_00123 60121 XP_41399	NP_0347612	

				62		
5	PSI PSI TIG TIG	Plexin_cytopl	NP_00575 21	XP_41614 32 XP_41614 42	NP_06126 71	XP_69075 93
6	RGS	RBD RBD Gol?	NP_00647 12	XP_00123 48041 XP_00123 19181	NP_05803 82	
7	cNMP_binding	Patatin	NP_00669 33	XP_42316 12 XP_00123 60551	NP_05661 61	XP_00192 12081
8	FtsJ DUF3381	Spb1_C	NP_06011 73	NP_00102 60961	NP_07958 61	XP_00192 17001 XP_00192 04971
9	Wbl1_Borealin_N	Borealin	NP_06057 11	XP_00123 60761 XP_42287 62	NP_08083 62	
10	KH_1	DEAD Helicase_C	NP_06113 52	XP_41988 21 XP_42619 52	XP_00148 09731	
11	ANF_recept	SBP_bac_3	NP_06877 51			XP_00134 41982

	or	Lig_chan				XP_00192 39771
12	N1221	DUF3402	NP_14907 92	XP_41794 12 XP_42354 22	NP_70579 11	
13	Sp100	SAND PHD Bromodom ain	NP_61241 14		XP_91011 91 XP_00147 35741	
14	Fucokinase	GHMP_kina se_N GHMP_kina se_C	NP_65949 62	XP_41404 92 XP_00123 33641	NP_75848 72	XP_00134 42721
15	SH3 ww	FCH DUF1664	NP_95883 91	XP_41557 72		XP_00133 74492 XP_00133 35072
16	Drf_GBD	FH2		NP_00101 27921		XP_00192 26181 XP_69352 53
17	CUB Pentaxin	GPS 7tm_2		NP_00102 62421		XP_00192 08271 XP_68965 43
18	Brix	7tm_1	NP_00103 57541			NP_95887 11 XP_00192 37501
19	Propep_M1 4 Peptidase_	Propep_M1 4 Peptidase_	NP_00112 09141 NP_00186	XP_41497 82		

	M14	M14	02			
20	Exo_endo_p hos CALC0C01	P2X_recept or x2	NP_00100 28371 NP_05740 21	XP_41528 72		
21	Kua	UFV1		XP_41751 42	NP_66351 31 XP_00147 84941	
22	Filament	Aldedh		XP_41762 42		XP_00191 99261 NP_95745 21
23	Extosin	Glyco_tran sf_64		XP_41839 62		XP_00191 90581 XP_00191 84961
24	Kelch_2 x6	DUF947		XP_41932 32		XP_68370 71 NP_00107 34131
25	FEZ	7tm_1		XP_41952 12		NP_00107 09001 XP_68512 12
26	Ank Ank Ank TRP_2	PKD_chann el		XP_42632 21		XP_00133 53132 XP_00192 02241
27	DUF3456	Methyltran sf_11	NP_00657 72 NP_06183 31	XP_00123 28291		
28	Dor1	Pep_defor		XP_00123		NP_00101 35241

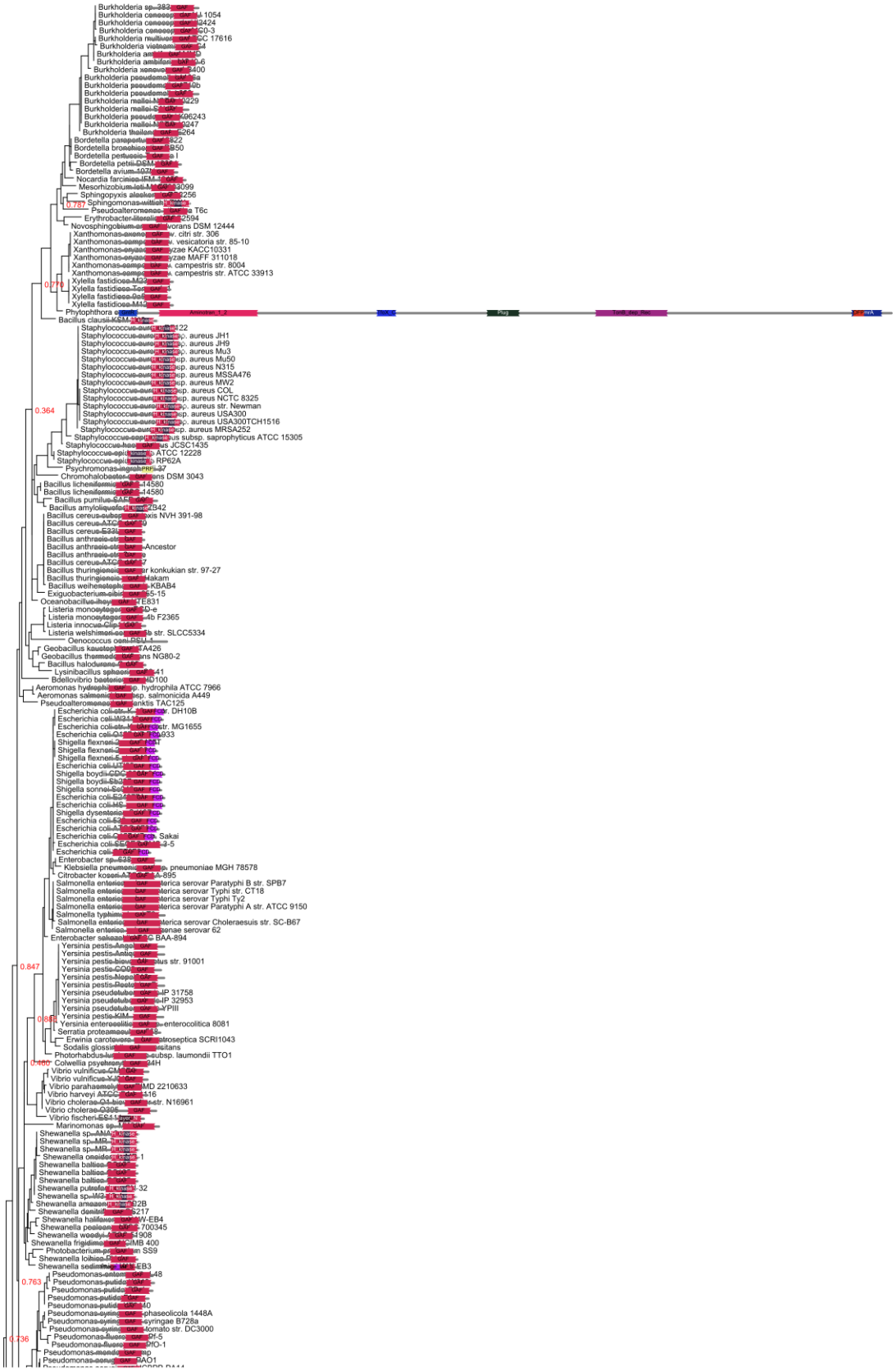
		mylase		29751		NP_00102 89021
30	IP_trans	Exo_endo_ phos CALC0C01	NP_00621 51 NP_57012 21	XP_00123 48131		

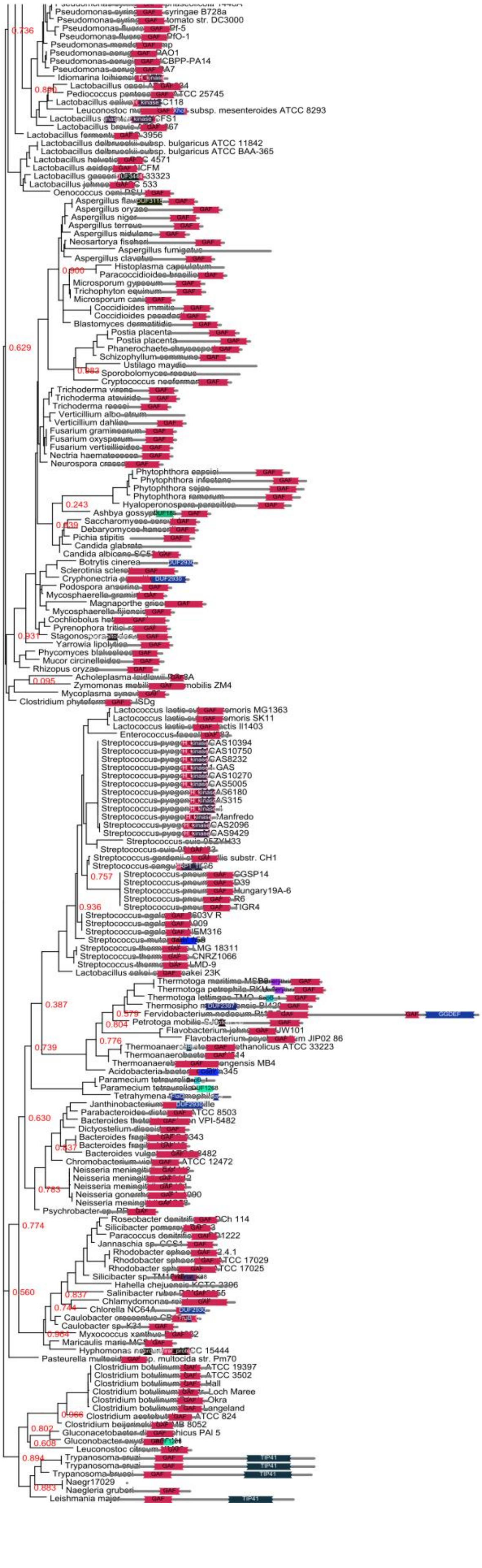
10.5 Discicristata

Fusion #	**PFAM Domain 1**	**PFAM Domain 2**	***Trypanosoma brucei***	***Trypanosoma cruzi***	***Leishmania major***	***Naegleria gruberi***
1	crypto_DASH	crypt_chrom_pln		Tc00.1047053509025.29 Tc00.1047053509027.10	LmjF09.0360	
2	MutS_I	MUTSd		Tc00.1047053507915.19 Tc00.1047053416511.9	LmjF15.1420	
3	Fucokinase	COG2605		Tc00.1047053507091.4 Tc00.1047053508377.9	LmjF16.0480	
4	CTGs	PRK06186	Tb927.1.1240	Tc00.1047053507949.4 Tc00.1047053506479.130	LmjF20.0560	Naegr1_estExt_fgeneshHS_pg.C_590022
5	COG1956	TIP41	Tb927.5.1250	Tc00.1047053509767.120	LmjF23.1460	Naegr1_fgeneshNG_pg.scaffold_39000108 Naegr1_estExt_gwp_gw1.C_190147
6	FYVE Fab1_TCP	MSS4	Tb11.47.0002	Tc00.1047053503793.20 Tc00.1047053506719.10	LmjF27.0890	
7	Alg14	COG5017	Tb927.6.1960	Tc00.1047053504071.90	LmjF30.0530	Naegr1_e_gw1.30.102.1 Naegr1_e_gw1.2.312.1
8	FSH1	Obg_like COG09602 MMR_HS		Tc00.1047053511021.99 Tc00.1047053508709.10	LmjF32.3110	

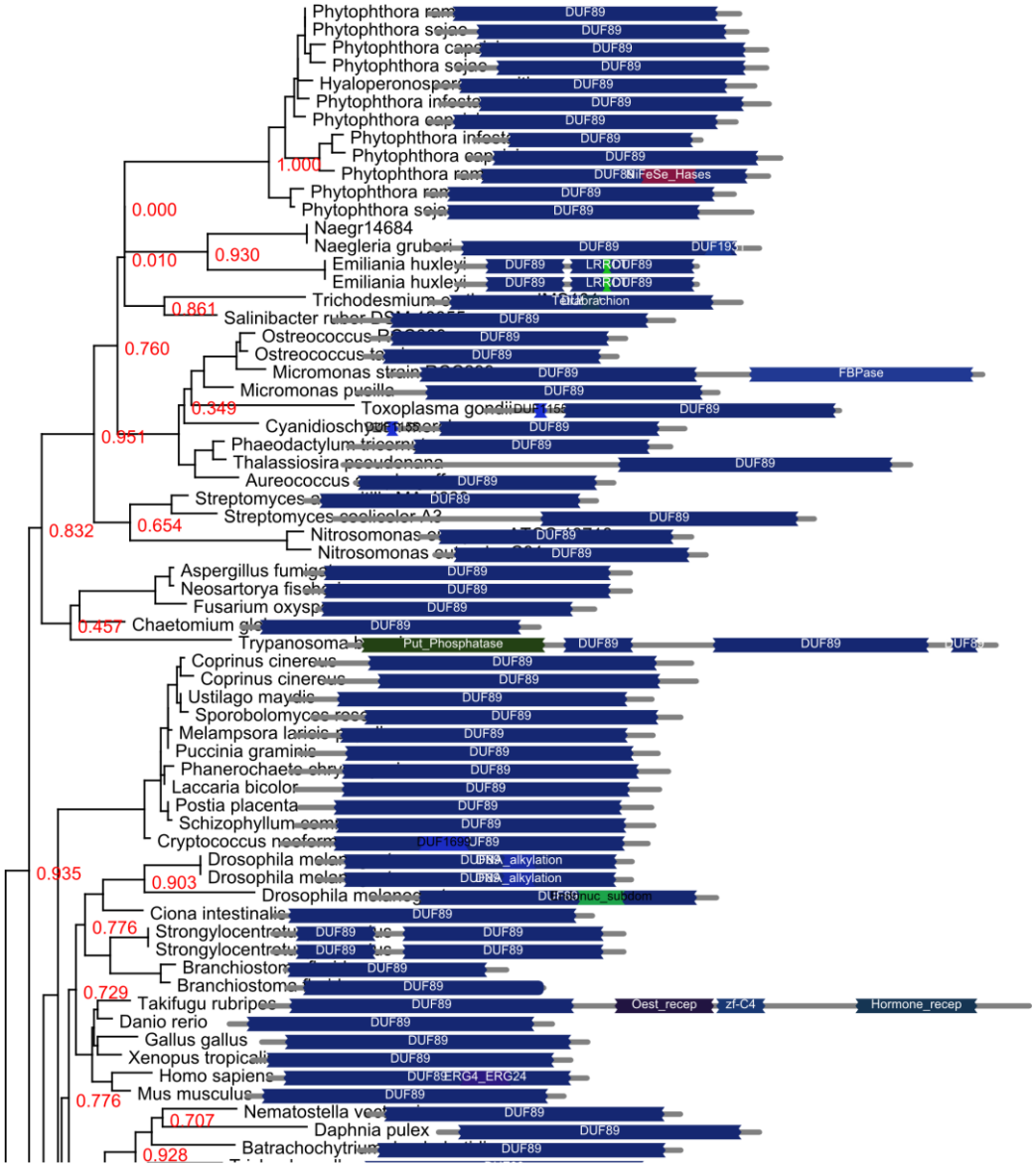
		R1_c				
9	WD40	Coatomer_WDAD	Tb927.2.6050		LmjF33.3210	Naegr1_estExt_fgeneshNG_pg.C_350117 Naegr1_estExt_fgeneshHS_pg.C_4830001
9	COG2319	Coatomer_WDAD COPI_C	Tb927.4.450	Tc00.1047053510687.149 Tc00.1047053510689.10	LmjF34.4310	Naegr1_estExt_fgeneshNG_pm.C_150014
10	BMS1 AARP2CN	COG5177 DUF663	Tb11.01.0820	Tc00.1047053510899.59 Tc00.1047053510901.10	LmjF28.1880	Naegr1_estExt_gwp_gw1.C_70074
11			Tb11.01.4660	Tc00.1047053508261.140 Tc00.1047053457979.5		
12	DKMT PPase-SF	DUF89	Tb927.7.5210			Naegr1_estExt_gwp_gw1.C_70121 Naegr1_fgeneshHS_pg.scaffold_9000018

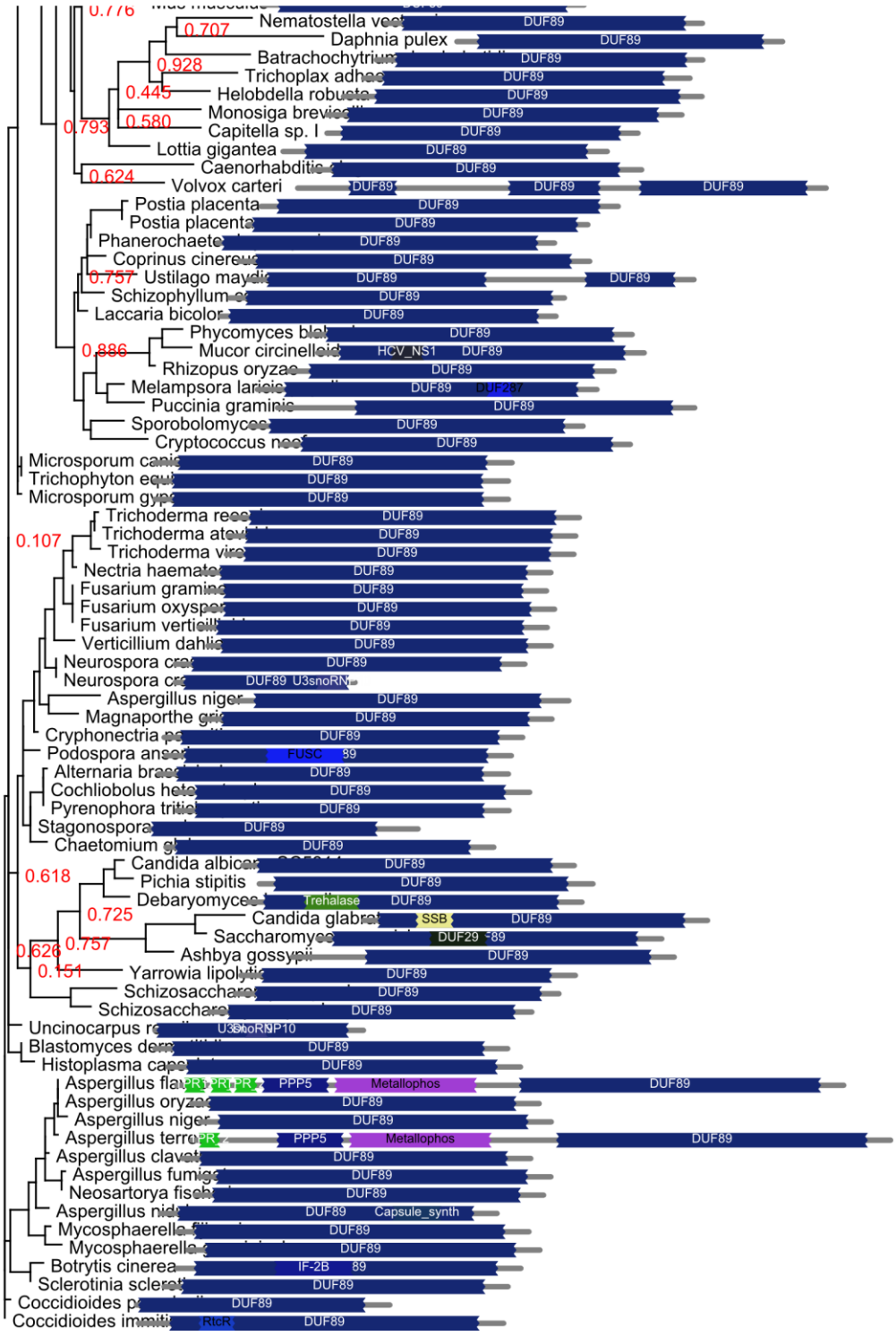
11 Phylogenetically Informative Tree Topologies





Pseudomonas syringae B728a
Pseudomonas tomato str. DC3000
Pseudomonas Pf-5
Pseudomonas PfO-1
Pseudomonas mendocina ymp
Pseudomonas PAO1
Pseudomonas UCBPP-PA14
Pseudomonas PA7
Idiomarina loihiensis
0.736
Lactobacillus
Pediococcus ATCC 25745
Lactobacillus C118
Leuconostoc subsp. mesenteroides ATCC 8293
Lactobacillus
Lactobacillus brevis
Lactobacillus fermentum 3956
Lactobacillus delbrueckii subsp. bulgaricus ATCC 11842
Lactobacillus delbrueckii subsp. bulgaricus ATCC BAA-365
Lactobacillus helveticus 4571
Lactobacillus acidophilus NCFM
Lactobacillus gasseri 33323
Lactobacillus johnsonii 533
Oenococcus oeni PSU
Aspergillus flavus
Aspergillus oryzae
Aspergillus niger
Aspergillus terreus
Aspergillus nidulans
Neosartorya fischeri
Aspergillus fumigatus
Aspergillus clavatus
Histoplasma capsulatum
Paracoccidioides brasiliensis
Microsporum gypseum
Trichophyton equinum
Microsporum canis
Coccidioides immitis
Coccidioides posadasii
Blastomyces dermatitidis
Postia placenta
Postia placenta
Phanerochaete chrysosporium
Schizophyllum commune
Ustilago maydis
Sporobolomyces roseus
Cryptococcus neoformans
0.629
Trichoderma virens
Trichoderma atroviride
Trichoderma reesei
Verticillium albo-atrum
Verticillium dahliae
Fusarium graminearum
Fusarium oxysporum
Fusarium verticillioides
Nectria haematococca
Neurospora crassa
Phytophthora capsici
Phytophthora infestans
Phytophthora sojae
Phytophthora ramorum
Hyaloperonospora parasitica
0.243
Ashbya gossypii
Saccharomyces cerevisiae
Debaryomyces hansenii
Pichia stipitis
Candida glabrata
Candida albicans SC5314
Botrytis cinerea
Sclerotinia sclerotiorum
Cryphonectria parasitica
Podospora anserina
Mycosphaerella graminicola
Magnaporthe grisea
Mycosphaerella fijiensis
Cochliobolus heterostrophus
Pyrenophora tritici-repentis
0.931
Stagonospora nodorum
Yarrowia lipolytica
Phycomyces blakesleeanus
Mucor circinelloides
Rhizopus oryzae
Acholeplasma laidlawii PG-8A
0.095
Zymomonas mobilis subsp. mobilis ZM4
Mycoplasma synoviae
Clostridium phytofermentans ISDg
Lactococcus lactis subsp. cremoris MG1363
Lactococcus lactis subsp. cremoris SK11
Lactococcus lactis subsp. lactis Il1403
Enterococcus faecalis
Streptococcus pyogenes MGAS10394
Streptococcus pyogenes MGAS10750
Streptococcus pyogenes MGAS8232
Streptococcus pyogenes M1 GAS
Streptococcus pyogenes MGAS10270
Streptococcus pyogenes MGAS5005
Streptococcus pyogenes MGAS6180
Streptococcus pyogenes MGAS315
Streptococcus pyogenes
Streptococcus pyogenes Manfredo
Streptococcus pyogenes MGAS2096
Streptococcus pyogenes MGAS9429
Streptococcus suis 05ZYH33
Streptococcus suis
Streptococcus gordonii str. Challis substr. CH1
Streptococcus sanguinis
0.757
Streptococcus pneumoniae CGSP14
Streptococcus pneumoniae D39
Streptococcus pneumoniae Hungary19A-6
Streptococcus pneumoniae R6
Streptococcus pneumoniae TIGR4
0.936
Streptococcus agalactiae 2603V R
Streptococcus agalactiae A909
Streptococcus agalactiae NEM316
Streptococcus mutans
Streptococcus thermophilus LMG 18311
Streptococcus thermophilus CNRZ1066
Streptococcus thermophilus LMD-9
Lactobacillus sakei subsp. sakei 23K
Thermotoga maritima MSB8
Thermotoga petrophila RKU-1
Thermotoga lettingae TMO
Thermosipho melanesiensis BI429
0.387
0.579
Fervidobacterium nodosum Rt17-B1
0.804
Petrotoga mobilis SJ95
Flavobacterium johnsoniae UW101
Flavobacterium psychrophilum JIP02 86
0.776
Thermoanaerobacter ethanolicus ATCC 33223
Thermoanaerobacter X514
0.739
Thermoanaerobacter tengcongensis MB4
Acidobacteria bacterium Ellin345
Paramecium tetraurelia
Paramecium tetraurelia
Tetrahymena thermophila
Janthinobacterium
Parabacteroides distasonis ATCC 8503
0.630
Bacteroides thetaiotaomicron VPI-5482
Dictyostelium discoideum
Bacteroides fragilis 9343
Bacteroides fragilis
Bacteroides vulgatus ATCC 8482
Chromobacterium violaceum ATCC 12472
Neisseria meningitidis
Neisseria meningitidis
Neisseria meningitidis
0.783
Neisseria gonorrhoeae FA 1090
Neisseria meningitidis
Psychrobacter sp.
0.774
Roseobacter denitrificans OCh 114
Silicibacter pomeroyi
Paracoccus denitrificans PD1222
Jannaschia sp. CCS1
Rhodobacter sphaeroides 2.4.1
Rhodobacter sphaeroides ATCC 17029
Rhodobacter sphaeroides ATCC 17025
Silicibacter sp. TM1040
Hahella chejuensis KCTC 2396
0.560
0.837
Salinibacter ruber
Chlamydomonas reinhardtii
Chlorella NC64A
Caulobacter crescentus CB15
Caulobacter sp. K31
Myxococcus xanthus DK 1622
Maricaulis maris MCS10
Hyphomonas neptunium ATCC 15444
Pasteurella multocida subsp. multocida str. Pm70
Clostridium botulinum ATCC 19397
Clostridium botulinum ATCC 3502
Clostridium botulinum Hall
Clostridium botulinum Loch Maree
Clostridium botulinum Okra
Clostridium botulinum Langeland
Clostridium acetobutylicum ATCC 824
0.802
Clostridium beijerinckii NCIMB 8052
Gluconacetobacter diazotrophicus PAl 5
0.608
Gluconobacter oxydans 621H
Leuconostoc citreum
0.894
Trypanosoma cruzi
Trypanosoma cruzi
Trypanosoma brucei
0.883
Naegr17029
Naegleria gruberi
Leishmania major
GAF
TIP41
GGDEF


Phytophthora ram
Phytophthora sojae
Phytophthora capsi
Phytophthora sojae
Hyaloperonospora
Phytophthora infesta
Phytophthora capsi
Phytophthora infesta
Phytophthora capsi
Phytophthora ram
Phytophthora ram
Phytophthora sojae
Naegr14684
Naegleria gruberi
Emiliana huxleyi
Emiliana huxleyi
Trichodesmium
Salinibacter ruber
Ostreococcus
Ostreococcus
Micromonas strain
Micromonas pusilla
Toxoplasma gondii
Cyanidioschyzon
Phaeodactylum tricornu
Thalassiosira pseudonana
Aureococcus
Streptomyces
Streptomyces coelicolor A3
Nitrosomonas
Nitrosomonas
Aspergillus fumig
Neosartorya fisch
Fusarium oxysp
Chaetomium gl
Trypanosoma
Coprinus cinereus
Coprinus cinereus
Ustilago maydis
Sporobolomyces
Melampsora larici
Puccinia graminis
Phanerochaete chry
Laccaria bicolor
Postia placenta
Schizophyllum com
Cryptococcus neoform
Drosophila melan
Drosophila melan
Drosophila melanog
Ciona intestinalis
Strongylocentrotu
Strongylocentrotu
Branchiostom
Branchiostoma
Takifugu rubripes
Danio rerio
Gallus gallus
Xenopus tropicali
Homo sapiens
Mus musculus
Nematostella vect
Daphnia pulex
Batrachochytrium
DUF89
DUF1935
FBPase
Put_Phosphatase
DUF1699
DNA_alkylation
nuc_subdom
Oest_recep
zf-C4
Hormone_recep
NiFeSe_Hases
ERG4_ERG24
1.000
0.000
0.010
0.930
0.861
0.760
0.349
0.951
0.832
0.654
0.457
0.935
0.903
0.776
0.776
0.729
0.776
0.707
0.928


0.776
0.707
0.928
0.445
0.793
0.580
0.624
0.757
0.886
0.107
0.618
0.725
0.626
0.757
0.151
Nematostella vec
Daphnia pulex
Batrachochytrium
Trichoplax adhae
Helobdella robusta
Monosiga brevic
Capitella sp. I
Lottia gigantea
Caenorhabditis
Volvox carteri
Postia placenta
Postia placenta
Phanerochaete
Coprinus cinere
Ustilago maydi
Schizophyllum
Laccaria bicolor
Phycomyces bla
Mucor circinelloi
Rhizopus oryzae
Melampsora lari
Puccinia graminis
Sporobolomyces
Cryptococcus neo
Microsporum can
Trichophyton equi
Microsporum gyp
Trichoderma ree
Trichoderma atro
Trichoderma vire
Nectria haemato
Fusarium gramin
Fusarium oxyspo
Fusarium vertici
Verticillium dahli
Neurospora cra
Neurospora cr
Aspergillus niger
Magnaporthe gri
Cryphonectria p
Podospora anse
Alternaria brass
Cochliobolus hete
Pyrenophora tritic
Stagonospora
Chaetomium g
Candida albica
Pichia stipitis
Debaryomyces
Candida glabra
Saccharomyces
Ashbya gossypii
Yarrowia lipolyti
Schizosaccharo
Schizosaccharo
Uncinocarpus re
Blastomyces der
Histoplasma cap
Aspergillus fla
Aspergillus oryza
Aspergillus niger
Aspergillus terre
Aspergillus clavat
Aspergillus fumig
Neosartorya fisch
Aspergillus nid
Mycosphaerella fi
Mycosphaerella
Botrytis cinerea
Sclerotinia sclero
Coccidioides p
Coccidioides immiti
DUF89
HCV_NS1
DUF287
U3snoRNP
FUSC
Trehalase
SSB
DUF29
U3snoRNP10
PPR
PPP5
Metallophos
Capsule_synth
IF-2B
RtcR

Bibliography

Abascal, F., Zardoya, R., & Posada, D. ProtTest: selection of best-fit models of protein evolution. *Bioinformatics, 21*(9), 2104-2105. doi: 10.1093/bioinformatics/bti263

Adl, S. M., Simpson, A. G. B., Farmer, M. A., Andersen, R. A., Anderson, O. R., Barta, J. R., et al. (2005). The New Higher Level Classification of Eukaryotes with Emphasis on the Taxonomy of Protists. *Journal of Eukaryotic Microbiology, 52*(5), 399-451. doi: 10.1111/j.1550-7408.2005.00053.x

Akaike, H. (2003). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716-723.

Akhunov, E. D., Goodyear, A. W., Geng, S., Qi, L.-L., Echalier, B., Gill, B. S., et al. (2003). The Organization and Rate of Evolution of Wheat Genomes Are Correlated With Recombination Rates Along Chromosome Arms. *Genome Research, 13*(5), 753-763. doi: 10.1101/gr.808603

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology, 215*(3), 403-410. doi: 10.1006/jmbi.1990.9999

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research, 25*(17), 3389-3402. doi: 10.1093/nar/25.17.3389

Alvarez, F., Cortinas, M. N., & Musto, H. (1996). The analysis of protein coding genes suggests monophyly of Trypanosoma. *Molecular Phylogenetics and Evolution, 5*(2), 333-343. doi: S1055-7903(96)90028-7

Andersson, J. O., Sarchfield, S. W., & Roger, A. J. (2005). Gene transfers from nanoarchaeota to an ancestor of diplomonads and parabasalids. *Molecular Biology and Evolution, 22*(1), 85-90.

Archibald, J., Mort, M., & Crawford, D. (2003). Bayesian Inference of Phylogeny: A Non-Technical Primer. *Taxon, 52*(2), 187-191.

Archibald, J. M., Rogers, M. B., Toop, M., Ishida, K., & Keeling, P. J. (2003). Lateral gene transfer and the evolution of plastid-targeted proteins in the secondary plastid-containing alga Bigelowiella natans. *Proceedings of the National Academy of Sciences, 100*(13), 7678-7683.

Archibald, J. M. (2008). The Eocyte hypothesis and the origin of the eukaryotic cells. *Proceedings of the National Academy of Sciences of the USA.* 105(51) 20049-20050.

Arisue, N., Hasegawa, M., & Hashimoto, T. (2004). Root of the Eukaryota tree as inferred from combined maximum likelihood analyses of multiple molecular sequence data. *Molecular Biology and Evolution, 22*, 409-420.

Ashlock, P. D. (1971). Monophyly and Associated Terms. [Article]. *Systematic Zoology, 20*(1), 63-69.

Aslett, M., Aurrecoechea, C., Berriman, M., Brestelli, J., Brunk, B. P., Carrington, M., et al. (2010). TriTrypDB: a functional genomic resource for the Trypanosomatidae. *Nucleic Acids Research, 38*(suppl 1), D457-D462. doi: 10.1093/nar/gkp851

Bailey, C. D., Koch, M. A., Mayer, M., Mummenhoff, K., O'Kane, S. L., Jr, Warwick, S. I., et al. (2006). Toward a Global Phylogeny of the Brassicaceae. *Molecular Biology and Evolution, 23*(11), 2142-2160.

Baldauf S. L., Palmer J. D., Doolittle W. F. 1996 The root of the universal tree and the origin of eukaryotes based on elongation factor phylogeny. Proc. Natl Acad. Sci. USA 93, 7749–7754.

Bapteste, E., Brinkmann, H., Lee, J. A., Moore, D. V., Sensen, C. W., Gordon, P., et al. (2002). The analysis of 100 genes supports the grouping of three highly divergent amoebae: Dictyostelium, Entamoeba, and Mastigamoeba. *Proceedings of the National Academy of Sciences, 99*(3), 1414-1419.

Bapteste, E., & Philippe, H. (2002). The potential value of indels as phylogenetic markers: position of trichomonads as a case study. *Molecular Biology and Evolution, 19*(6), 972-977.

Bashton, M., & Chothia, C. (2002). The geometry of domain combination in proteins. *Journal of Molecular Biology, 315*(4), 927-939.

Bass, D., Moreira, D., López-García , P., Polet, S., Chao, E. E., & von der Heyden, S. (2005). Polyubiquitin insertions and the phylogeny of Cercozoa and Rhizaria. *Protist, 156*(2), 149-161.

Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., et al. (2004). The Pfam protein families database. *Nucleic Acids Research, 32*(suppl_1), D138-141.

Baum, B. (1992). Combining Trees as a Way of Combining Data Sets for Phylogenetic Inference, and the Desirability of Combining Gene Trees. *Taxon, 41*(1), 3-10.

Berriman, M., Ghedin, E., Hertz-Fowler, C., Blandin, G., Renauld, H., Bartholomeu, D. C., et al. (2005). The Genome of the African Trypanosome Trypanosoma brucei. *Science, 309*(5733), 416-422. doi: 10.1126/science.1112642

Bininda-Emonds, O. R. P. (2004). The evolution of supertrees. *Trends in Ecology & Evolution, 19*(6), 315-322.

Bininda-Emonds, O. R. P., Gittleman, J. L., & Steel, M. A. (2003). THE (SUPER)TREE OF LIFE: Procedures, Problems, and Prospects. *Annual Review of Ecology and Systematics, 33*(1), 265-289.

Blair, J. E., & Hedges, S. B. (2005). Molecular phylogeny and divergence times of deuterostome animals. *Mol Biol Evol, 22*(11), 2275-2284.

Boyer, M., Madoui, M.A., Gimenez, G., La Scola, B., Raoult, D. (2010). Phylogenetic and phyletic studies of informational genes in genomes highlight existence of a 4th domain of life including giant viruses. *PLoS ONE* 5: e15530

Brilli, M., & Fani, R. (2004). Molecular evolution of hisB genes. *Journal of Molecular Evolution, 58*(2), 225-237.

Brinkman, F. S., Blanchard, J. L., Cherkasov, A., Av-Gay, Y., Brunham, R. C., & Fernandez, R. C. (2002). Evidence that plant-like genes in Chlamydia species reflect an ancestral relationship between Chlamydiaceae, Cyanobacteria, and the chloroplast. *Genome Research, 12*(8), 1159-1167.

Brown, J. R., & Doolittle, W. F. (1997). Archaea and the prokaryote-to-eukaryote transition. *Microbiology and Molecular Biology Reviews, 61*(4), 456-502.

Burki, F., & Pawlowski, J. (2006). Monophyly of Rhizaria and multigene phylogeny of unicellular bikonts. *Molecular Biology and Evolution, 23*(10), 1922-1930.

Burki, F., Shalchian-Tabrizi, K., Minge, M., Skjæveland, Å., Nikolaev, S. I., Jakobsen, K. S., et al. (2007). Phylogenomics Reshuffles the Eukaryotic Supergroups. *PLoS ONE, 2*(8), e790.

Burki, F., Shalchian-Tabrizi, K., & Pawlowski, J. (2008). Phylogenomics reveals a new 'megagroup' including most photosynthetic eukaryotes. *Biology Letters, 4*(4), 366-369.

Castresana, J. (2000). Selection of Conserved Blocks from Multiple Alignments for Their Use in Phylogenetic Analysis. *Molecular Biology and Evolution, 17*(4), 540-552.

Cavalier-Smith, T. (1993). Kingdom protozoa and its 18 phyla. *Microbiological Reviews, 57*(4), 953-994.

Cavalier-Smith, T. (1998). A revised six-kingdom system of life. *Biological Reviews of the Cambridge Philosophical Society, 73*(3), 203-266.

Cavalier-Smith, T. (2002). The phagotrophic origin of eukaryotes and phylogenetic classification of Protozoa. *International Journal of Systematic Evolutionary Microbiology, 52*(2), 297-354.

Cavalier-Smith, T. (2003). The excavate protozoan phyla Metamonada Grasse emend. (Anaeromonadea, Parabasalia, Carpediemonas, Eopharyngia) and Loukozoa emend. (*Jakobea, Malawimonas*): their evolutionary affinities and new higher taxa. *International Journal of Systematic and Evolutionary Microbiology, 53*(6), 1741-1758.

Cavalier-Smith, T. (2004). Chromalveolate Diversity and Cell Megaevolution *Organelles, Genomes and Eukaryote Phylogeny*: CRC Press.

Cavalier-Smith T. (2006). Rooting the tree of life by transition analyses. *Biol. Direct* 1, 19.

Chor, B., & Tuller, T. (2005). Maximum likelihood of evolutionary trees: hardness and approximation. *Bioinformatics, 21 Suppl 1*, i97-106.

Conant, G., & Wagner, A. (2005). The rarity of gene shuffling in conserved genes. *Genome Biology, 6*(6), 50-50.

Copeland, H. F. (1938). The Kingdoms of Organisms. *The Quarterly Review of Biology, 13*(4), 383-420.

Copley, R. R., Schultz, J., Ponting, C. P., & Bork, P. (1999). Protein families in multicellular organisms. *Curr Opin Struct Biol, 9*(3), 408-415.

Cotton, J. A., & Wilkinson, M. (2009). Supertrees join the mainstream of phylogenetics. *Trends in Ecology and Evolution, 24*(1), 1-3.

Dacks, J. B., Marinets, A., Ford Doolittle, W., Cavalier-Smith, T., & Logsdon, J. M., Jr. (2002). Analyses of RNA Polymerase II genes from free-living protists: phylogeny, long branch attraction, and the eukaryotic big bang. *Molecular Biology and Evolution, 19*(6), 830-840.

Darwin, C. (1837-1838). Notebook B: [Transmutation of species], 2010, from http://darwin-online.org.uk

Darwin, C. (1876). *The origin of species by means of natural selection, or the preservation of favoured races in the struggle for life* (6th ed.). London: John Murray.

Dayhoff, M. (1965). *Atlas of protein sequence and structure*: Nat Biomed Research Foundation.

de Queiroz, A., & Ashton, K. G. (2004). The phylogeny of a species-level tendency: species heritability and possible deep origins of Bergmann's rule in tetrapods. *Evolution, 58*(8), 1674-1684.

Dehal, P., Satou, Y., Campbell, R. K., Chapman, J., Degnan, B., De Tomaso, A., et al. (2002). The Draft Genome of Ciona intestinalis: Insights into Chordate and Vertebrate Origins. *Science, 298*(5601), 2157-2167.

Delsuc, F., Brinkmann, H., Chourrout, D., & Philippe, H. (2006). Tunicates and not cephalochordates are the closest living relatives of vertebrates. *Nature, 439*(7079), 965-968.

Delsuc, F., Brinkmann, H., & Philippe, H. (2005). Phylogenomics and the reconstruction of the tree of life. *Nature Reviews: Genetics, 6*(5), 361-375.

Dollo, L. (1922). *Les Céphalopodes déroulés et l'irréversibilité de l'évolution.* Amsterdam: Bijdragen tot de Dierkunde.

Doolittle, R. F. (1995). The multiplicity of domains in proteins. *Annual Review of Biochemistry, 64*, 287-314.

Doolittle, W. F., & Brown, J. R. (1994). Tempo, mode, the progenote, and the universal root. *Proceedings of the National Academy of Sciences, 91*(15), 6721-6728.

Driskell, A. C., Ane, C., Burleigh, J. G., McMahon, M. M., O'Meara B, C., & Sanderson, M. J. (2004). Prospects for building the tree of life from large sequence databases. *Science, 306*(5699), 1172-1174.

Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). The theory behind profile HMMs *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acid* (pp. 356): Cambridge University Press.

Durrens, P., Nikolski, M., & Sherman, D. (2008). Fusion and Fission of Genes Define a Metric between Fungal Genomes. *PLoS Computational Biology, 4*(10), e1000200.

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research, 32*(5), 1792-1797.

Edgar, R. C. (2010). Quality measures for protein alignment benchmarks. *Nucleic Acids Research, 38*(7), 2145-2153.

Edgar, R. C., & Batzoglou, S. (2006). Multiple sequence alignment. *Current Opinion in Structual Biology, 16*(3), 368-373.

Eisen, J. A. (1998). Phylogenomics: Improving Functional Predictions for Uncharacterized Genes by Evolutionary Analysis. *Genome Research, 8*(3), 163-167.

El-Sayed, N. M., Myler, P. J., Bartholomeu, D. C., Nilsson, D., Aggarwal, G., Tran, A.-N., et al. (2005). The Genome Sequence of Trypanosoma cruzi, Etiologic Agent of Chagas Disease. *Science, 309*(5733), 409-415.

El-Sayed, N. M., Myler, P. J., Blandin, G., Berriman, M., Crabtree, J., Aggarwal, G., et al. (2005). Comparative Genomics of Trypanosomatid Parasitic Protozoa. *Science, 309*(5733), 404-409.

Embley, T. M., & Hirt, R. P. (1998). Early branching eukaryotes? *Current Opinion in Genetics & Development, 8*(6), 624-629.

Everett, K. D., Kahane, S., Bush, R. M., & Friedman, M. G. (1999). An unspliced group I intron in 23S rRNA links Chlamydiales, chloroplasts, and mitochondria. *Journal of Bacteriology, 181*(16), 4734-4740.

Felsenstein, J. (1978). Cases in which Parsimony or Compatibility Methods will be Positively Misleading. *Systematic Biology, 27*(4), 401-401.

Felsenstein, J. (1985). Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution, 39*(4), 783-791.

Felsenstein, J. (2004). *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates.

Finn, R. D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J. E., et al. (2010). The Pfam protein families database. *Nucleic Acids Research, 38*(suppl_1), D211-222.

Fitch, W. M., & Markowitz, E. (1970). An improved method for determining codon variability in a gene and its application to the rate of fixation of mutations in evolution. *Biochemical Genetics, 4*(5), 579-593.

Fitzpatrick, D., Logue, M., Stajich, J., & Butler, G. (2006). A fungal phylogeny based on 42 complete genomes derived from supertree and combined gene analysis. *BMC Evolutionary Biology, 6*(1), 99.

Forterre, P., & Philippe, H. (1999). Where is the root of the universal tree of life? *BioEssays, 21*(10), 871-879.

Foster, P. G. (2001). *The Idiot's Guide to the Zen of Likelihood in a Nutshell in Seven Days for Dummies, Unleashed*. Informal Explanation. Department of Zoology, The Natural History Museum. London. Retrieved from http://www.bmnh.org/~pf/idiots.pdf

Foster, P. G., & Hickey, D. A. (1999). Compositional Bias May Affect Both DNA-Based and Protein-Based Phylogenetic Reconstructions. *Journal of Molecular Evolution, 48*(3), 284-290.

Fritz-Laylin, L. K., Prochnik, S. E., Ginger, M. L., Dacks, J. B., Carpenter, M. L., Field, M. C., et al. (2010). The Genome of Naegleria gruberi Illuminates Early Eukaryotic Versatility. *Cell, 140*(5), 631-642.

Galagan, J. E., Calvo, S. E., Borkovich, K. A., Selker, E. U., Read, N. D., Jaffe, D., et al. (2003). The genome sequence of the filamentous fungus Neurospora crassa. *Nature, 422*(6934), 859-868.

Galtier, N. (2001). Maximum-likelihood phylogenetic analysis under a covarion-like model. *Molecular Biology and Evolution, 18*(5), 866-873.

Gao, L., Su, Y. J., & Wang, T. (2010). Plastid genome sequencing, comparative genomics, and phylogenomics: Current status and prospects. *Journal of Systematics and Evolution, 48*(2), 77-93. doi: 10.1111/j.1759-6831.2010.00071.x

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*: W. H. Freeman.

Gascuel, O. (1997). BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Molecular Biology and Evolution, 14*(7), 685-695.

Giaever, G., Chu, A. M., Ni, L., Connelly, C., Riles, L., Veronneau, S., et al. (2002). Functional profiling of the Saccharomyces cerevisiae genome. *Nature, 418*(6896), 387-391.

Gillis, W., St John, J., Bowerman, B., & Schneider, S. (2009). Whole genome duplications and expansion of the vertebrate GATA transcription factor gene family. *BMC Evolutionary Biology, 9*(1), 207.

Gordon, A. D. (1986). Consensus supertrees: The synthesis of rooted trees containing overlapping sets of labeled leaves. *Journal of Classification, 3*(2), 335-348.

Gould, S. J. (1970). Dollo on Dollo's law: Irreversibility and the status of evolutionary laws. *Journal of the History of Biology, 3*(2), 189-212.

Gould, S. J. (1990). *Hen's Teeth and Horse's Toes: Further Reflections in Natural History*. London: Penguin.

Gribaldo, S., Poole, A. M., Daubin, V., Forterre, P., & Brochier-Armanet, C. (2010). The origin of eukaryotes and their relationship with the Archaea: are we at a phylogenomic impasse? [10.1038/nrmicro2426]. *Nat Rev Micro, 8*(10), 743-752.

Guindon, S., Dufayard, J.-F., Lefort, V., Anisimova, M., Hordijk, W., & Gascuel, O. (2010). New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Systematic Biology, 59*(3), 307-321.

Guindon, S., & Gascuel, O. (2003). A Simple, Fast, and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood. *Systematic Biology, 52*(5), 696-704.

Haag, J., O'HUigin, C., & Overath, P. (1998). The molecular phylogeny of trypanosomes: evidence for an early divergence of the Salivaria. *Molecular and Biochemical Parasitology, 91*(1), 37-49.

Haeckel, E. (1866). *Generelle Morphologie der Organismen allgemeine Grundzüge der organischen Formen-Wissenschaft, mechanisch begründet durch die von Charles Darwin reformirte Descendenz-Theorie*. Berlin: G. Reimer.

Hamilton, P. B., Stevens, J. R., Gaunt, M. W., Gidley, J., & Gibson, W. C. (2004). Trypanosomes are monophyletic: evidence from genes for glyceraldehyde phosphate dehydrogenase and small subunit ribosomal RNA. *International Journal for Prasitology, 34*(12), 1393-1404.

Hampl, V., Hug, L., Leigh, J. W., Dacks, J. B., Lang, B. F., Simpson, A. G. B., et al. (2009). Phylogenomic analyses support the monophyly of Excavata and resolve relationships among eukaryotic "supergroups". *Proceedings of the National Academy of Sciences, 106*(10), 3859-3864.

Hasegawa, M., & Hashimoto, T. (1993). Ribosomal RNA trees misleading? [10.1038/361023b0]. *Nature, 361*(6407), 23-23.

Hedges, S. B. (2002). The origin and evolution of model organisms. *Nature Review: Genetics, 3*(11), 838-849.

Hibbett, D. S., Binder, M., Bischoff, J. F., Blackwell, M., Cannon, P. F., Eriksson, O. E., et al. (2007). A higher-level phylogenetic classification of the Fungi. *Mycological Research, 111*(5), 509-547.

Hillis, D. M. (1996). Inferring complex phylogenies. *Nature, 383*(6596), 130-131.

Hirt, R. P., Logsdon, J. M., Jr., Healy, B., Dorey, M. W., Doolittle, W. F., & Embley, T. M. (1999). Microsporidia are related to Fungi: evidence from the largest subunit of RNA polymerase II and other proteins. *Proceedings of the National Academy of Sciences, 96*(2), 580-585.

Hordijk, W., & Gascuel, O. (2005). Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. *Bioinformatics, 21*(24), 4338-4347.

Horner, D. S., & Embley, T. M. (2001). Chaperonin 60 phylogeny provides further evidence for secondary loss of mitochondria among putative early-branching eukaryotes. *Molecular Biology and Evolution, 18*(10), 1970-1975.

Huelsenbeck, J. P., Larget, B., Miller, R. E., & Ronquist, F. (2002). Potential applications and pitfalls of Bayesian inference of phylogeny. *Systematic Biology, 51*(5), 673-688.

Hughes, A., & Piontkivska, H. (2003). Molecular phylogenetics of Trypanosomatidae: contrasting results from 18S rRNA and protein phylogenies. *Kinetoplastid Biology and Disease, 2*(1), 15.

Hughes, A. L., & Piontkivska, H. (2003). Phylogeny of Trypanosomatidae and Bodonidae (Kinetoplastida) based on 18S rRNA: evidence for paraphyly of Trypanosoma and six other genera. *Molecular Biology and Evolution, 20*(4), 644-652.

Huson, D., Richter, D., Rausch, C., Dezulian, T., Franz, M., & Rupp, R. (2007). Dendroscope: An interactive viewer for large phylogenetic trees. *BMC Bioinformatics, 8*(1), 460.

ICGSC. (2004). Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature, 432*(7018), 695-716.

Ivens, A. C., Peacock, C. S., Worthey, E. A., Murphy, L., Aggarwal, G., Berriman, M., et al. (2005). The Genome of the Kinetoplastid Parasite, Leishmania major. *Science, 309*(5733), 436-442. doi: 10.1126/science.1112680

James, T. Y., Kauff, F., Schoch, C. L., Matheny, P. B., Hofstetter, V., Cox, C. J., et al. (2006). Reconstructing the early evolution of Fungi using a six-gene phylogeny. [10.1038/nature05110]. *Nature, 443*(7113), 818-822.

Jenkins, C., & Fuerst, J. A. (2001). Phylogenetic analysis of evolutionary relationships of the planctomycete division of the domain bacteria based on amino acid sequences of elongation factor Tu. *Journal of Molecular Evolution, 52*(5), 405-418.

Jones, D. T., Taylor, W. R., & Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences, 8*(3), 275-282. doi: 10.1093/bioinformatics/8.3.275

Jukes, T. H., & Cantor, C. R. (1969). *Evolution of Protein Molecules*: Academy Press.

Kamper, J., Kahmann, R., Bolker, M., Ma, L. J., Brefort, T., Saville, B. J., et al. (2006). Insights from the genome of the biotrophic fungal plant pathogen Ustilago maydis. *Nature, 444*(7115), 97-101.

Katinka, M. D., Duprat, S., Cornillot, E., Metenier, G., Thomarat, F., Prensier, G., et al. (2001). Genome sequence and gene compaction of the eukaryote parasite Encephalitozoon cuniculi. *Nature, 414*(6862), 450-453.

Keane, T., Creevey, C., Pentony, M., Naughton, T., & Mclnerney, J. (2006). Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified. *BMC Evolutionary Biology, 6*(1), 29.

Keeling, P. J., & Inagaki, Y. (2004). A class of eukaryotic GTPase with a punctate distribution suggesting multiple functional replacements of translation elongation factor 1α. *Proceedings of the National Academy of Sciences, 101*(43), 15380-15385.

Keeling, P. J., & Palmer, J. D. (2001). Lateral transfer at the gene and subgenic levels in the evolution of eukaryotic enolase. *Proceedings of the National Academy of Science USA, 98*(19), 10745-10750.

Kellis, M., Birren, B. W., & Lander, E. S. (2004). Proof and evolutionary analysis of ancient genome duplication in the yeast Saccharomyces cerevisiae. *Nature, 428*(6983), 617-624.

Koonin, E. V. (2010). The Incredible Expanding Ancestor of Eukaryotes. *Cell, 140*(5), 606-608.

Kovalchuk, A., & Driessen, A. (2010). Phylogenetic analysis of fungal ABC transporters. *BMC Genomics, 11*(1), 177.

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

Krylov, D. M., Wolf, Y. I., Rogozin, I. B., & Koonin, E. V. (2003). Gene Loss, Protein Sequence Divergence, Gene Dispensability, Expression Level, and Interactivity Are Correlated in Eukaryotic Evolution. *Genome Research, 13*(10), 2229-2235.

Kummerfeld, S. K., & Teichmann, S. A. (2005). Relative rates of gene fusion and fission in multi-domain proteins. *Trends in Genetics, 21*(1), 25-30.

Lake, J.A., Henderson, E., Oakes, M., Clark, M.W. (1984). Eocytes: a new ribosome structure indicates a kingdom with a close relationship to eukaryotes. *Proceedings of the National Academy of Sciences of the United States of America* 81: 3786–90

Lake, J. A., de la Cruz, V. F., Ferreira, P. C., Morel, C., & Simpson, L. (1988). Evolution of parasitism: kinetoplastid protozoan history reconstructed from mitochondrial rRNA gene sequences. *Proceedings of the National Academy of Sciences of the United States of America, 85*(13), 4779-4783.

Lanave, C., Preparata, G., Sacone, C., & Serio, G. (1984). A new method for calculating evolutionary substitution rates. *Journal of Molecular Evolution, 20*(1), 86-93.

Le, S. Q., & Gascuel, O. (2008). An Improved General Amino Acid Replacement Matrix. *Molecular Biology and Evolution, 25*(7), 1307-1320.

Le, S. Q., Lartillot, N., & Gascuel, O. (2008). Phylogenetic mixture models for proteins. *Philosophical Transactions of the Royal Society B: Biological Sciences, 363*(1512), 3965-3976.

Leonard, G., Stevens, J. R., & Richards, T. A. (2009). REFGEN and TREENAMER: automated sequence data handling for phylogenetic analysis in the genomic era. *Evolutionary Bioinformatics Online, 5*, 1-4.

Letunic, I., Doerks, T., & Bork, P. (2009). SMART 6: recent updates and new developments. *Nucleic Acids Res, 37*(Database issue), D229-232.

Liolios, K., Chen, I. M., Mavromatis, K., Tavernarakis, N., Hugenholtz, P., Markowitz, V. M., et al. (2010). The Genomes On Line Database (GOLD) in 2009: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Res, 38*(Database issue), D346-354.

Liu, F. G., Miyamoto, M. M., Freire, N. P., Ong, P. Q., Tennant, M. R., Young, T. S., et al. (2001). Molecular and morphological supertrees for eutherian (placental) mammals. *Science, 291*(5509), 1786-1789.

Liu, Y., Leigh, J. W., Brinkmann, H., Cushion, M. T., Rodriguez-Ezpeleta, N., Philippe, H., et al. (2009). Phylogenomic Analyses Support the Monophyly of Taphrinomycotina, including Schizosaccharomyces Fission Yeasts. *Molecular Biology and Evolution, 26*(1), 27-34.

Lockhart, P., Novis, P., Milligan, B. G., Riden, J., Rambaut, A., & Larkum, T. (2006). Heterotachy and Tree Building: A Case Study with Plastids and Eubacteria. *Molecular Biology and Evolution, 23*(1), 40-45.

Lockhart, P., Steel, M., Hendy, M., & Penny, D. (1994). Recovering Evolutionary Trees under a More Realistic Model of Sequence. *Molecular Biology and Evolution, 11*(4), 605-612.

Lopez, P., Casane, D., & Philippe, H. (2002). Heterotachy, an Important Process of Protein Evolution. *Molecular Biology and Evolution, 19*(1), 1-7.

Lukeš, J., Jirků, M., Doležel, D., Kral'ová, I., Hollar, L., & Maslov, D. A. (1997). Analysis of Ribosomal RNA Genes Suggests That Trypanosomes Are Monophyletic. *Journal of Molecular Evolution, 44*(5), 521-527.

Marchler-Bauer, A., Anderson, J. B., Cherukuri, P. F., DeWeese-Scott, C., Geer, L. Y., Gwadz, M., et al. (2005). CDD: a Conserved Domain Database for protein classification. *Nucleic Acids Research, 33*(Database issue), D192-196.

Marchler-Bauer, A., Anderson, J. B., Chitsaz, F., Derbyshire, M. K., DeWeese-Scott, C., Fong, J. H., et al. (2009). CDD: specific functional annotation with the Conserved Domain Database. *Nucleic Acids Research, 37*(Database issue)

Marchler-Bauer, A., & Bryant, S. H. (2004). CD-Search: protein domain annotations on the fly. *Nucleic Acids Research, 32*(Web Server issue), W327-331.

Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O., & Eisenberg, D. (1999). Detecting protein function and protein-protein interactions from genome sequences. *Science, 285*(5428), 751-753.

Martin, W., & Muller, M. (1998). The hydrogen hypothesis for the first eukaryote. *Nature, 392*(6671), 37-41.

Maslov, D. A., Lukes, J., Jirku, M., & Simpson, L. (1996). Phylogeny of trypanosomes as inferred from the small and large subunit rRNAs: implications for the evolution of parasitism in the trypanosomatid protozoa. *Molecular and Biochemical Parasitology, 75*(2), 197-205.

Massingham, T., & Goldman, N. (2007). Statistics of the Log-Det Estimator. *Molecular Biology and Evolution, 24*(10), 2277-2285.

Mats, E. (2008). On the difference between mono-, holo-, and paraphyletic groups: a consistent distinction of process and pattern. *Biological Journal of the Linnean Society, 94*(1), 217-220.

McLaughlin, D. J., Hibbett, D. S., Lutzoni, F., Spatafora, J. W., & Vilgalys, R. (2009). The search for the fungal tree of life. *Trends in Microbiology, 17*(11), 488-497.

McMahon, M. M., & Sanderson, M. J. (2006). Phylogenetic Supermatrix Analysis of GenBank Sequences from 2228 Papilionoid Legumes. *Systematic Biology, 55*(5), 818-836.

Medina, M. (2005). Genomes, phylogeny, and evolutionary systems biology. *Proceedings of the National Academy of Sciences, 102*(Suppl 1), 6630-6635.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics, 21*(6), 1087-1092.

Minotto, L., Edwards, M. R., & Bagnara, A. S. (2000). Trichomonas vaginalis: characterization, expression, and phylogenetic analysis of a carbamate kinase gene sequence. *Experimental Parasitology, 95*(1), 54-62.

Moreira, D., Lopez-Garcia, P., & Vickerman, K. (2004). An updated view of kinetoplastid phylogeny using environmental sequences and a closer outgroup: proposal for a new classification of the class Kinetoplastea. *International Journal of Systematic and Evolutionary Microbiology, 54*(Pt 5), 1861-1875.

Morris, P. F., Schlosser, L. R., Onasch, K. D., Wittenschlaeger, T., Austin, R., & Provart, N. (2009). Multiple Horizontal Gene Transfer Events and Domain Fusions Have Created Novel Regulatory and Metabolic Networks in the Oomycete Genome. *PLoS ONE, 4*(7).

Nakamura, Y., Itoh, T., & Martin, W. (2006). Rate and polarity of gene fusion and fission in Oryza sativa and Arabidopsis thaliana. *Molecular Biology and Evolution, 24*(1), 110-121.

Olsen, G., & Woese, C. (1993). Ribosomal RNA: a key to phylogeny. *Federation of American Societies for Experimental Biology Journal, 7*(1), 113-123.

Olsen, O. W. (1986). *Animal parasites: their life cycles and ecology* (3rd ed.). Baltimore: University Park Press.

Page, R. D. M., & Holmes, E. C. (1998). Chapter 5: Measuring Genetic Change *Molecular Evolution: A Phylogenetic Approach* (pp. 346pp): Blackwell.

Parfrey, L. W., Grant, J., Tekle, Y. I., Lasek-Nesselquist, E., Morrison, H. G., Sogin, M. L., et al. (2010). Broadly Sampled Multigene Analyses Yield a Well-Resolved Eukaryotic Tree of Life. *Systematic Biology*.

Perrière, G., & Gouy, M. (1996). WWW-query: An on-line retrieval system for biological sequence banks. *Biochimie, 78*(5), 364-369.
Pethica, R., Barker, G., Kovacs, T., & Gough, J. (2010). TreeVector: Scalable, Interactive, Phylogenetic Trees for the Web. *PLoS ONE, 5*(1), e8934.
Philippe, H. (2000). Early–branching or fast–evolving eukaryotes? An answer based on slowly evolving positions. *Proceedings of the Royal Society of London. Series B: Biological Sciences, 267*(1449), 1213-1221.
Philippe, H. (2000). Opinion: long branch attraction and protist phylogeny. *Protist, 151*(4), 307-316.
Philippe, H., & Forterre, P. (1999). The Rooting of the Universal Tree of Life Is Not Reliable. *Journal of Molecular Evolution, 49*(4), 509-523.
Philippe, H., & Germot, A. (2000). Phylogeny of eukaryotes based on ribosomal RNA: long-branch attraction and models of sequence evolution. *Molecular Biology and Evolution, 17*(5), 830-834.
Philippe, H., Lartillot, N., & Brinkmann, H. (2005). Multigene analyses of bilaterian animals corroborate the monophyly of Ecdysozoa, Lophotrochozoa, and Protostomia. *Mol Biol Evol, 22*(5), 1246-1253.
Philippe, H., Snell, E. A., Bapteste, E., Lopez, P., Holland, P. W., & Casane, D. (2004). Phylogenomics of eukaryotes: impact of missing data on large alignments. *Molecular Biology and Evolution, 21*(9), 1740-1752.
Philippe, H., Zhou, Y., Brinkmann, H., Rodrigue, N., & Delsuc, F. (2005). Heterotachy and long-branch attraction in phylogenetics. *BMC Evolutionary Biology, 5*(1), 50.
Piontkivska, H., & Hughes, A. L. (2005). Environmental kinetoplastid-like 18S rRNA sequences and phylogenetic relationships among Trypanosomatidae: paraphyly of the genus Trypanosoma. *Molecular and Biochemical Parasitology, 144*(1), 94-99.
Pisani, D., Cotton, J. A., & McInerney, J. O. (2007). Supertrees Disentangle the Chimeric Origin of Eukaryotic Genomes. *Molecular Biology and Evolution*.
Pisani, D., & Wilkinson, M. (2002). Matrix Representation with Parsimony, Taxonomic Congruence, and Total Evidence. *Systematic Biology, 51*(1), 151-155.
Price, M. N., Dehal, P. S., & Arkin, A. P. (2009). FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix. *Molecular Biology and Evolution, 26*(7), 1641-1641.
Price, M. N., Dehal, P. S., & Arkin, A. P. (2010). FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE, 5*(3).
Pryer, K. M., Schneider, H., Zimmer, E. A., & Ann Banks, J. (2002). Deciding among green plants for whole genome studies. *Trends in Plant Science, 7*(12), 550-554.
Putnam, N. H., Butts, T., Ferrier, D. E., Furlong, R. F., Hellsten, U., Kawashima, T., et al. (2008). The amphioxus genome and the evolution of the chordate karyotype. *Nature, 453*(7198).
Ragan, M. A. (1992). Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution, 1*(1), 53-58.
Rambaut, A. FigTree (Version 1.3.1). Retrieved from http://tree.bio.ed.ac.uk/software/figtree/
Rannala, B., & Yang, Z. (1996). Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *Journal of Molecular Evolution, 43*(3), 304-311.

Rappé, M. S., & Giovannoni, S. J. (2003). The Uncultured Microbial Majority. *Annual Review of Microbiology, 57*(1), 369-394.

Rensing, S. A., Lang, D., Zimmer, A. D., Terry, A., Salamov, A., Shapiro, H., et al. (2008). The Physcomitrella Genome Reveals Evolutionary Insights into the Conquest of Land by Plants. *Science, 319*(5859), 64-69.

Richards, T. A., & Cavalier-Smith, T. (2005). Myosin domain evolution and the primary divergence of eukaryotes. [10.1038/nature03949]. *Nature, 436*(7054), 1113-1118. doi: http://www.nature.com/nature/journal/v436/n7054/suppinfo/nature03949_S1.html

Richards, T. A., Hirt, R. P., Williams, B. A., & Embley, T. M. (2003). Horizontal gene transfer and the evolution of parasitic protozoa. *Protist, 154*(1), 17-32.

Richards, T. A., Soanes, D. M., Foster, P. G., Leonard, G., Thornton, C. R., & Talbot, N. J. (2009). Phylogenomic analysis demonstrates a pattern of rare and ancient horizontal gene transfer between plants and fungi. *Plant Cell, 21*(7), 1897-1911.

Rivera, M. C., & Lake, J. A. (1992). Evidence that eukaryotes and eocyte prokaryotes are immediate relatives. *Science, 257*(5066), 74-76.

Rivera, M. C., & Lake, J. A. (2004). The ring of life provides evidence for a genome fusion origin of eukaryotes. *Nature, 431*(7005), 152-155.

Rodríguez-Ezpeleta, N., Brinkmann, H., Burey, S. C., Roure, B., Burger, G., Löffelhardt, W., et al. (2005). Monophyly of Primary Photosynthetic Eukaryotes: Green Plants, Red Algae, and Glaucophytes. *Current Biology, 15*(14), 1325-1330.

Rodríguez-Ezpeleta, N., Brinkmann, H., Burger, G., Roger, A. J., Gray, M. W., Philippe, H., et al. (2007). Toward Resolving the Eukaryotic Tree: The Phylogenetic Positions of Jakobids and Cercozoans. *Current Biology, 17*(16), 1420-1425.

Rodríguez, F., Oliver, J. L., Marín, A., & Medina, J. R. (1990). The general stochastic model of nucleotide substitution. *Journal of Theoretical Biology, 142*(4), 485-501.

Roger, A. J., & Simpson, A. G. B. (2009). Evolution: Revisiting the Root of the Eukaryote Tree. *Current Biology, 19*(4), R165-R167.

Rokas, A., & Holland, P. W. (2000). Rare genomic changes as a tool for phylogenetics. *Trends in Cell Biology, 15*(11), 454-459.

Ronquist, F., & Huelsenbeck, J. P. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics, 19*(12), 1572-1574.

Satou, Y., & Satoh, N. (2003). [Draft genome sequence of *Ciona intestinalis* and its meaning]. *Tanpakushitsu Kakusan Koso, 48*(9), 1282-1286.

Schoch, C. L., Sung, G.-H., López-Giráldez, F., Townsend, J. P., Miadlikowska, J., Hofstetter, V., et al. (2009). The Ascomycota Tree of Life: A Phylum-wide Phylogeny Clarifies the Origin and Evolution of Fundamental Reproductive and Ecological Traits. *Systematic Biology, 58*(2), 224-239.

Schultz, J., Milpetz, F., Bork, P., & Ponting, C. P. (1998). SMART, a simple modular architecture research tool: Identification of signaling domains. *Proceedings of the National Academy of Sciences of the United States of America, 95*(11), 5857-5864.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics, 6*(2), 461-464.

Searcy, D. G., & Hixon, W. G. (1991). Cytoskeletal origins in sulfur-metabolizing archaebacteria. *Biosystems, 25*(1-2), 1-11.

Shimodaira, H., & Hasegawa, M. (1999). Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference. *Molecular Biology and Evolution, 16*(8), 1114.

Shimodaira, H., & Hasegawa, M. (2001). CONSEL: for assessing the confidence of phylogenetic tree selection. *Bioinformatics, 17*(12), 1246-1247.

Simpson, A. G., Inagaki, Y., & Roger, A. J. (2006). Comprehensive multigene phylogenies of excavate protists reveal the evolutionary positions of "primitive" eukaryotes. *Molecular Biology and Evolution, 23*(3), 615-625.

Simpson, A. G., Lukes, J., & Roger, A. J. (2002). The evolutionary history of kinetoplastids and their kinetoplasts. *Molecular Biology and Evolution, 19*(12), 2071-2083.

Simpson, A. G., & Roger, A. J. (2004). The real 'kingdoms' of eukaryotes. *Current Biology, 14*(17), R693-696.

Simpson, A. G., Stevens, J. R., & Lukes, J. (2006). The evolution and diversity of kinetoplastid flagellates. *Trends in Parasitology, 22*(4), 168-174.

Simpson, A. G. B., Gill, E. E., Callahan, H. A., Litaker, R. W., & Roger, A. J. (2004). Early Evolution within Kinetoplastids (Euglenozoa), and the Late Emergence of Trypanosomatids. *Protist, 155*(4), 407-422.

Simpson, A. G. B., Inagaki, Y., & Roger, A. J. (2006). Comprehensive Multigene Phylogenies of Excavate Protists Reveal the Evolutionary Positions of “Primitive” Eukaryotes. *Molecular Biology and Evolution, 23*(3), 615-625.

Skophammer R. G., Servin J. A., Herbold C. W., Lake J. A. (2007). Evidence for a gram-positive, eubacterial root of the tree of life. *Mol. Biol. Evol.* 24, 1761–1768.

Slot, J. C., & Rokas, A. (2010). Multiple GAL pathway gene clusters evolved independently and by different mechanisms in fungi. *Proceedings of the National Academy of Sciences, 107*(22), 10136-10141.

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology, 147*(1), 195-197.

Snel, B., Bork, P., & Huynen, M. (2000). Genome evolution: gene fusion versus gene fission. *Trends in Genetics, 16*(1), 9-11.

Sodergren, E., Shen, Y., Song, X., Zhang, L., Gibbs, R. A., & Weinstock, G. M. (2006). Shedding genomic light on Aristotle's lantern. *Developmental Biology, 300*(1), 2-8.

Sodergren, E., Weinstock, G. M., Davidson, E. H., Cameron, R. A., Gibbs, R. A., Angerer, R. C., et al. (2006). The genome of the sea urchin Strongylocentrotus purpuratus. *Science, 314*(5801), 941-952.

Sogin, M. L. (1991). Early evolution and the origin of eukaryotes. *Current Opinion in Genetics & Development, 1*(4), 457-463.

Sonnhammer, E. L., Eddy, S. R., & Durbin, R. (1997). Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins, 28*(3), 405-420.

Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., et al. (2002). The Bioperl toolkit: Perl modules for the life sciences. *Genome Research, 12*(10), 1611-1618.

Stamatakis, A. (2006). RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics, 22*(21), 2688-2690.

Stamatakis, A., Ludwig, T., & Meier, H. (2005). RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics, 21*(4), 456-463.

Stechmann, A., & Cavalier-Smith, T. (2002). Rooting the eukaryote tree by using a derived gene fusion. *Science, 297*(5578), 89-91.
Stechmann, A., & Cavalier-Smith, T. (2003). The root of the eukaryote tree pinpointed. *Current Biology, 13*(17), R665-666.
Steel, M. (1994). Recovering a tree from the leaf colourations it generates under a Markov model. *Applied Mathematics Letters, 7*(2), 19-23.
Steel, M., Huson, D., & Lockhart, P. J. (2000). Invariable Sites Models and Their Use in Phylogeny Reconstruction. *Systematic Biology, 49*(2), 225-232.
Sterelny, K., & Griffiths, P. E. (1999). *Sex and death: an introduction to philosophy of biology*. Chicago: The University of Chicago Press.
Stevens, J. R. (2008). Kinetoplastid phylogenetics, with special reference to the evolution of parasitic trypanosomes. *Parasite, 15*(3), 226-232.
Stevens, J. R., & Gibson, W. (1999). The molecular evolution of trypanosomes. *Parasitology Today, 15*(11), 432-437.
Stevens, J. R., Noyes, H. A., Dover, G. A., & Gibson, W. C. (1999). The ancient and divergent origins of the human pathogenic trypanosomes, Trypanosoma brucei and T. cruzi. *Parasitology, 118 (Pt 1)*, 107-116.
Stevens, J. R., Noyes, H. A., Schofield, C. J., & Gibson, W. (2001). The molecular evolution of Trypanosomatidae. *Parasitology, 48*, 1-56.
Stiller, J. W., & Hall, B. D. (1999). Long-branch attraction and the rDNA model of early eukaryotic evolution. *Molecular Biology and Evolution, 16*(9), 1270-1279.
Stover, B., & Muller, K. (2010). TreeGraph 2: Combining and visualizing evidence from different phylogenetic analyses. *BMC Bioinformatics, 11*(1), 7.
Stuart, G. W., Moffett, K., & Leader, J. J. (2002). A Comprehensive Vertebrate Phylogeny Using Vector Representations of Protein Sequences from Whole Genomes. *Molecular Biology and Evolution, 19*(4), 554-562.
Swofford, D. L. (2003). *PAUP* Phylogenetic Analysis Using Parsimony (*and Other Methods) Version 4.04beta*: Sinauer Associates, Sunderland, Massachusetts.
Talavera, G., & Castresana, J. (2007). Improvement of Phylogenies after Removing Divergent and Ambiguously Aligned Blocks from Protein Sequence Alignments. *Systematic Biology, 56*(4), 564-577.
Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., et al. (2003). The COG database: an updated version includes eukaryotes. *BMC Bioinformatics, 4*(41), 41.
Tatusov, R. L., Koonin, E. V., & Lipman, D. J. (1997). A genomic perspective on protein families. *Science, 278*(5338), 631-637.
Tavaré, S. (1986). Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences *American Mathematical Society: Lectures on Mathematics in the Life Sciences* (Vol. 17, pp. 57-86): Amer Mathematical Society.
TBestDB. Taxonomically Broad EST Database, 2010, from http://amoebidia.bcm.umontreal.ca/pepdb/
Teichmann, S. A., Chothia, C., & Gerstein, M. (1999). Advances in structural genomics. *Current Opinion in Structural Biology, 9*(3), 390-399.
Teichmann, S. A., & Mitchison, G. (1999). Making family trees from gene families. *Nature Genetics, 21*(1), 66-67.
Teichmann, S. A., Park, J., & Chothia, C. (1998). Structural assignments to the Mycoplasma genitalium proteins show extensive gene duplications and domain rearrangements. *Proceedings of the National Academy of Science USA, 95*(25), 14658-14663.

Townsend, J., López-Giráldez, F., & Friedman, R. (2008). The Phylogenetic Informativeness of Nucleotide and Amino Acid Sequences for Reconstructing the Vertebrate Tree. *Journal of Molecular Evolution, 67*(5), 437-447.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., et al. (2001). The Sequence of the Human Genome. *Science, 291*(5507), 1304-1351.

Walker, G., Dacks, J., & Embley, M. T. (2006). Ultrastructural Description of *Breviata anathema*, N. Gen, N., N. Sp., the Organism Previously Studied as "Mastigamoeba invertens". *The Journal of Eukaryotic Microbiology*. 53(2), 65-78.

Waterston, R. H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J. F., Agarwal, P., et al. (2002). Initial sequencing and comparative analysis of the mouse genome. *Nature, 420*(6915), 520-562.

Webb, C. O., & Donoghue, M. J. (2005). Phylomatic: tree assembly for applied phylogenetics. *Molecular Ecology Notes, 5*(1), 181-183.

Whittaker, R. H. (1969). New Concepts of Kingdoms of Organisms. *Science, 163*(3863), 150-160. doi: 10.1126/science.163.3863.150

Wickstead, B., Gull, K., & Richards, T. (2010). Patterns of kinesin evolution reveal a complex ancestral eukaryote with a multifunctional cytoskeleton. *BMC Evolutionary Biology, 10*(1), 110.

Wilkinson, M., & Thorley, J. L. (1998). Reduced supertrees. *Trends in Ecology & Evolution, 13*(7), 283-283.

Williams, T.A., Embley, T.M., Heinz, E. (2011). Informational Gene Phylogenies Do Not Support a Fourth Domain of Life for Nucleocytoplasmic Large DNA Viruses. *PLoS ONE* 6(6): e21080

Willis, K. J., & McElwain, J. C. (2002). *Evolution of Plants*. Oxford: Oxford University Press.

Woese, C.R., Fox, G.E. (1977). Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proceedings of the National Academy of Sciences of the United States of America* 74: 5088–90

Woese, C. R. (1987). Bacterial evolution. *Microbiological Reviews, 51*(2), 221-271.

Woese, C. R. (1996). Phylogenetic trees: Whither microbiology? *Current Biology, 6*(9), 1060-1063.

Woese, C. R., Kandler, O., & Wheelis, M. L. (1990). Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences, 87*(12), 4576-4579.

Wood, V., Gwilliam, R., Rajandream, M. A., Lyne, M., Lyne, R., Stewart, A., et al. (2002). The genome sequence of Schizosaccharomyces pombe. *Nature, 415*(6874), 871-880.

Wright, A.-D. G., Li, S., Feng, S., Martin, D. S., & Lynn, D. H. (1999). Phylogenetic position of the kinetoplastids, Cryptobia bullocki, Cryptobia catostomi, and Cryptobia salmositica and monophyly of the genus Trypanosoma inferred from small subunit ribosomal RNA sequences. *Molecular and biochemical parasitology, 99*(1), 69-76.

Yang, Z. (1994). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular Evolution, 39*(3), 306-314.

Yang, Z. (1996). Among-site rate variation and its impact on phylogenetic analyses. *Trends in Ecology & Evolution, 11*(9), 367-372.

Yang, Z., & Roberts, D. (1995). On the use of nucleic acid sequences to infer early branchings in the tree of life. *Molecular Biology and Evolution, 12*(3), 451-458.

Zuckerkandl, E., & Pauling, L. (1965). Molecules as documents of evolutionary history. *Journal of Theoretical Biology, 8*(2), 357-366.