2017

Mirage: A Novel Multiple Protein Sequence Alignment Tool

Alex Nord
University of Montana

Recommended Citation

Nord, Alex, "Mirage: A Novel Multiple Protein Sequence Alignment Tool" (2017). *Graduate Student Theses, Dissertations, & Professional Papers*. 11110.
https://scholarworks.umt.edu/etd/11110

MIRAGE: A NOVEL MULTIPLE PROTEIN SEQUENCE ALIGNMENT TOOL

By

Alex Nord

Bachelor of Arts, Reed College, Portland, OR, 2014

Thesis

presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Computer Science

The University of Montana
Missoula, MT

Winter 2018

Approved by:

Scott Whittenburg, Dean
Graduate School

Travis Wheeler Ph.D., Chair
Computer Science

Oliver Serang Ph.D.
Computer Science

Steve Lodmell Ph.D.
Biology

Nord, Alex, M.S., January 2018 Computer Science

Mirage: A Novel Multiple Protein Sequence Alignment Tool

Chairperson: Travis Wheeler

A fundamental problem in computational biology is the organization of many related sequences into a multiple sequence alignment (MSA) [2]. MSAs have a range of research applications, such as inferring phylogeny [22] and identifying regions of conserved sequence that indicate functional similarity [18]. In the case of protein isoforms, MSAs are valuable tools for transitively annotating post-translational modifications (PTMs) by enabling information transfer between known PTM sites and the sites that they align to [11].

For protein MSA tools, one challenging biological phenomenon is alternative splicing, wherein identical genomic sequence will differentially select from a subset of available coding regions (exons), depending on the biochemical environment [21]. Traditional methods struggle to align the islands of non-homologous sequence produced by alternative splicing, and frequently compensate for the penalties incurred from aligning non-identical characters by aligning small pieces of relatively similar sequence from alternative exons in a way that avoids extreme gap penalties but falsely indicates sequence homology.

Presented here is Mirage, a novel protein MSA tool capable of accurately aligning alternatively spliced proteins by first mapping proteins to the genomic sequence that encoded them and then aligning proteins to one another based on the relative positions of their coding DNA. This method of transitive alignment demonstrates an awareness of intron splice site locations and resolves the problems associated with alternative splicing in traditional MSA tools.

ACKNOWLEDGMENTS

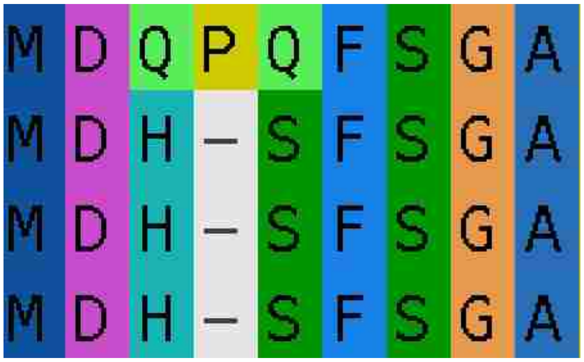
First and foremost, I am immeasurably grateful to Travis Wheeler for the incredible guidance, compelling ideas, and contagious excitement that he has provided at every step along Mirage's development (for more on Travis, refer to the Acknowledgments section in my dissertation (Nord, forthcoming)). I also owe a debt of gratitude to Peter Hornbeck at PhosphoSite, who initially proposed the idea of transitive alignment and by doing so catalyzed this amazing project. Many thanks to my friends in the Wheeler lab group at UM, especially Kaitlin Carey, whose investigations into ARFs produced some of the most exciting figures in my thesis. Thanks to Robyn Berg for sustaining the caffeine high through which Mirage materialized. And, finally, infinite thanks to my family and friends for the incredible support that they have given me over the past couple of years.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

A fundamental problem in computational biology is the organization of many related sequences into a multiple sequence alignment (MSA) [2]. An MSA is a matrix in which each row corresponds to one member of an input set of biologically related sequences and columns display the common ancestry shared by the letters found in the column, using the gap character ('-') to signify an insertion or deletion (collectively, indel) that is required to place other letters in shared columns.



M	D	Q	P	Q	F	S	G	A
M	D	H	–	S	F	S	G	A
M	D	H	–	S	F	S	G	A
M	D	H	–	S	F	S	G	A

Figure 1.1: A multiple sequence alignment.

Multiple sequence alignment is one of the classical problems of computational biology, and some modern multiple sequence alignment tools can trace their lineages as far back as 30 years [10]. Traditionally, MSAs are computed using a scoring scheme that effectively maximizes character identity within columns while minimizing the number of gaps, resulting in compact alignments that prefer occasional mismatches to long gaps [2]. In most contexts, these methods generate accurate and informative alignments, but certain biological phenomena can contradict the logical underpinnings of traditional MSA heuristics and cause tools to produce inaccurate alignments. For protein MSA tools, one such challenging phenomenon is alternative splicing, wherein identical genomic sequence will differentially select protein-coding exons from the available exon pool, depending on the biochemical environment [21]. Alternatively spliced products of the same gene are commonly referred to as "isoforms."

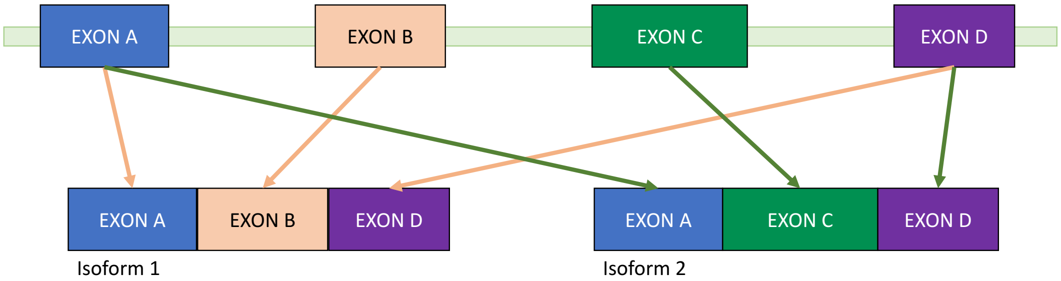


Figure 1.2: Diagram of alternative splicing, where exons B and C are alternatively excluded from the protein isoforms.

The capacity for alternative splicing is biologically advantageous, as it allows cells to dynamically adjust the functional dispositions of certain genes in response to environmental changes, such as the presence of a pathogen or the overexpression of another gene [21]. An estimated 95% of human genes undergo some form of alternative splicing [15], which can range from minor changes in the number of amino acids contributed by a specific exon (alternative splice site usage) to the removal and addition of entire exons. The ubiquity and variety of alternative splicing events make understanding the causes and functional impacts of alternative splicing central to numerous biological research projects, and make the ability to accurately and automatically transfer information across alternatively spliced protein sequences through splice-aware multiple sequence alignments desirable.

PhosphoSite, for example, is a database of protein sequences dedicated to annotating sites where post-translational modifications (PTMs) are known to occur [11]. PTMs are augmentations made to individual amino acids that fine-tune protein behavior, and are difficult to detect because the physical difference between an amino acid that has been modified with a PTM and one that has not is marginal. A valuable feature of PhosphoSite is that it enables researchers to infer possible PTMs through the use of multiple protein sequence alignments. If an amino acid with a known PTM in a mouse protein has a human homolog (or an unannotated mouse homolog), then there is reason to suspect that the other protein may also undergo modification at the homologous site, and an accurate alignment of those proteins' sequences would clearly and effectively communicate that homology. Accurately transferring information about PTMs that occur on alternatively spliced exons is facilitated by the ability to precisely align splice-isoforms both within and across species, but this proves challenging for most MSA software packages.

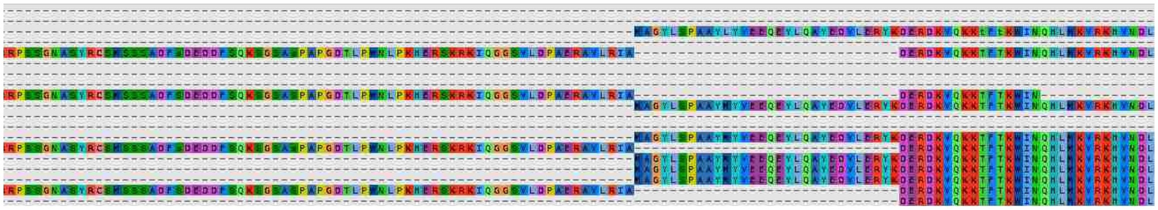
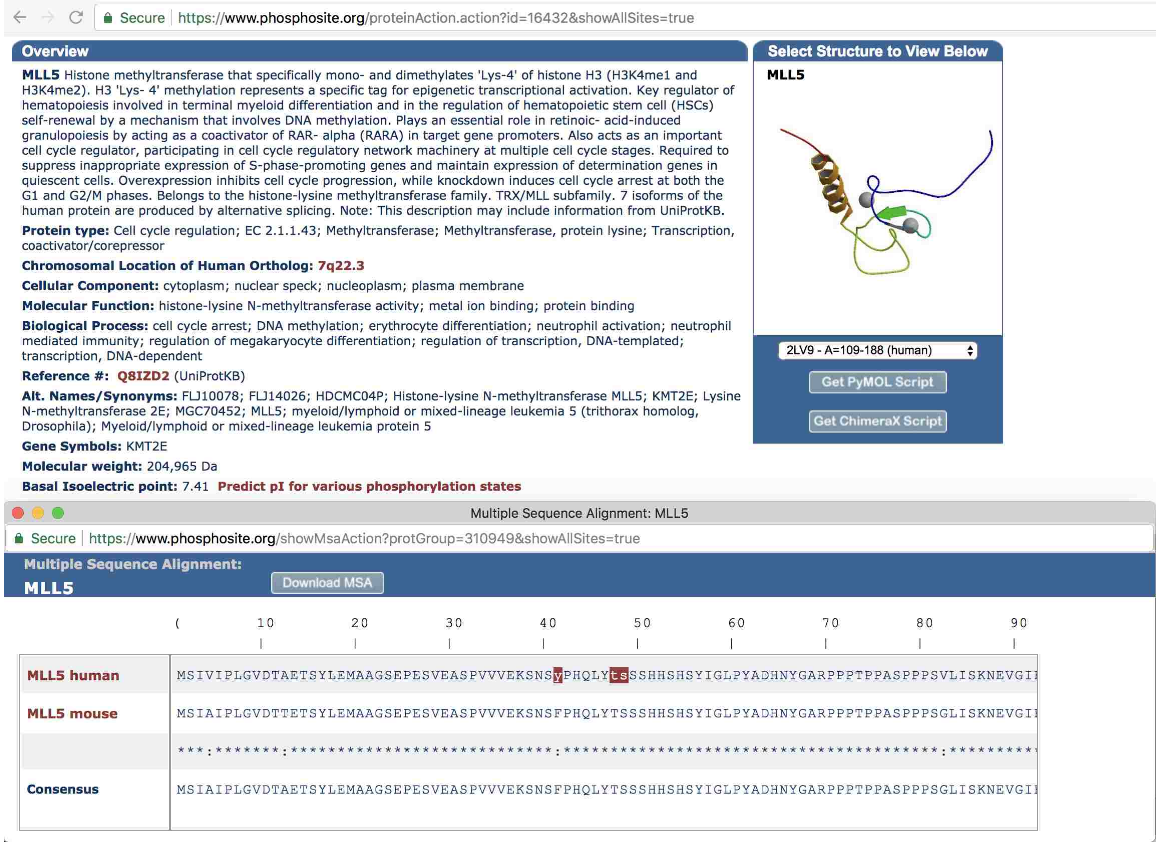


Figure 1.3: A multiple sequence alignment of alternatively spliced BPAG1 isoforms, produced by Mirage.

The reason traditional MSA software tools struggle to align alternatively spliced proteins is that they see mutually exclusive exons as sequences that should be aligned to one another because the bounding exons shared by both isoforms are supposed to be aligned. The benefits of aligning small pieces of relatively similar sequence from alternative exons (and thus avoiding large indel costs) make overlaying alternative exons optimal under the heuristics that guide traditional MSA tools, but produce MSAs that communicate false sequence homology.

Over time, MSA tools have improved with regards to other shortcomings by implementing clever methods for estimating the evolutionary distance between sequences [1], iteratively refining multiple sequence alignments [6], and quantifying alignment quality [19], but without substantial changes to their algorithmic cores they will continually struggle to



https://www.phosphosite.org/proteinAction.action?id=16432&showAllSites=true

Overview

MLL5 Histone methyltransferase that specifically mono- and dimethylates 'Lys-4' of histone H3 (H3K4me1 and H3K4me2). H3 'Lys- 4' methylation represents a specific tag for epigenetic transcriptional activation. Key regulator of hematopoiesis involved in terminal myeloid differentiation and in the regulation of hematopoietic stem cell (HSCs) self-renewal by a mechanism that involves DNA methylation. Plays an essential role in retinoic- acid-induced granulopoiesis by acting as a coactivator of RAR- alpha (RARA) in target gene promoters. Also acts as an important cell cycle regulator, participating in cell cycle regulatory network machinery at multiple cell cycle stages. Required to suppress inappropriate expression of S-phase-promoting genes and maintain expression of determination genes in quiescent cells. Overexpression inhibits cell cycle progression, while knockdown induces cell cycle arrest at both the G1 and G2/M phases. Belongs to the histone-lysine methyltransferase family. TRX/MLL subfamily. 7 isoforms of the human protein are produced by alternative splicing. Note: This description may include information from UniProtKB.

Protein type: Cell cycle regulation; EC 2.1.1.43; Methyltransferase; Methyltransferase, protein lysine; Transcription, coactivator/corepressor

Chromosomal Location of Human Ortholog: 7q22.3

Cellular Component: cytoplasm; nuclear speck; nucleoplasm; plasma membrane

Molecular Function: histone-lysine N-methyltransferase activity; metal ion binding; protein binding

Biological Process: cell cycle arrest; DNA methylation; erythrocyte differentiation; neutrophil activation; neutrophil mediated immunity; regulation of megakaryocyte differentiation; regulation of transcription, DNA-templated; transcription, DNA-dependent

Reference #: Q8IZD2 (UniProtKB)

Alt. Names/Synonyms: FLJ10078; FLJ14026; HDCMC04P; Histone-lysine N-methyltransferase MLL5; KMT2E; Lysine N-methyltransferase 2E; MGC70452; MLL5; myeloid/lymphoid or mixed-lineage leukemia 5 (trithorax homolog, Drosophila); Myeloid/lymphoid or mixed-lineage leukemia protein 5

Gene Symbols: KMT2E

Molecular weight: 204,965 Da

Basal Isoelectric point: 7.41 **Predict pI for various phosphorylation states**

https://www.phosphosite.org/showMsaAction?protGroup=310949&showAllSites=true

Multiple Sequence Alignment:
MLL5

```
          (        10        20        30        40        50        60        70        80        90
                    |         |         |         |         |         |         |         |         |
MLL5 human   MSIVIPLGVDTAETSYLEMAAGSEPESVEASPVVVEKSNSYPHQLYTSSSHHSHSYIGLPYADHNYGARPPPTPPASPPPSVLISKNEVGII
MLL5 mouse   MSIAIPLGVDTTETSYLEMAAGSEPESVEASPVVVEKSNSFPHQLYTSSSHHSHSYIGLPYADHNYGARPPPTPPASPPPSGLISKNEVGII
             ***:*******:****************************:****************************************:**********
Consensus    MSIAIPLGVDTAETSYLEMAAGSEPESVEASPVVVEKSNSFPHQLYTSSSHHSHSYIGLPYADHNYGARPPPTPPASPPPSGLISKNEVGII
```

Figure 1.4: A sample page from the PhosphoSite website, providing information about the MLL5 gene and indicating known PTMs on a human MLL5 isoform.

align alternatively spliced proteins. This is because all modern multiple protein sequence alignment tools treat input sequences simply as strings from a biological alphabet, so that they can apply indel models that roughly agree with insertion and deletion events on an evolutionary timescale while remaining agnostic about the actual biological origins of the sequences. These models are ubiquitously implemented using variations on the Needleman-Wunsch dynamic programming algorithm [14], when generalized to accept multiple sequence alignments as inputs.

Needleman-Wunsch aligns biological sequences by flood-filling a table with the cumulative scores for the best possible alignments of each pair of the inputs' prefixes, as illustrated in Figure 1.6. Diagonal movements on the table represent that the corresponding charac-

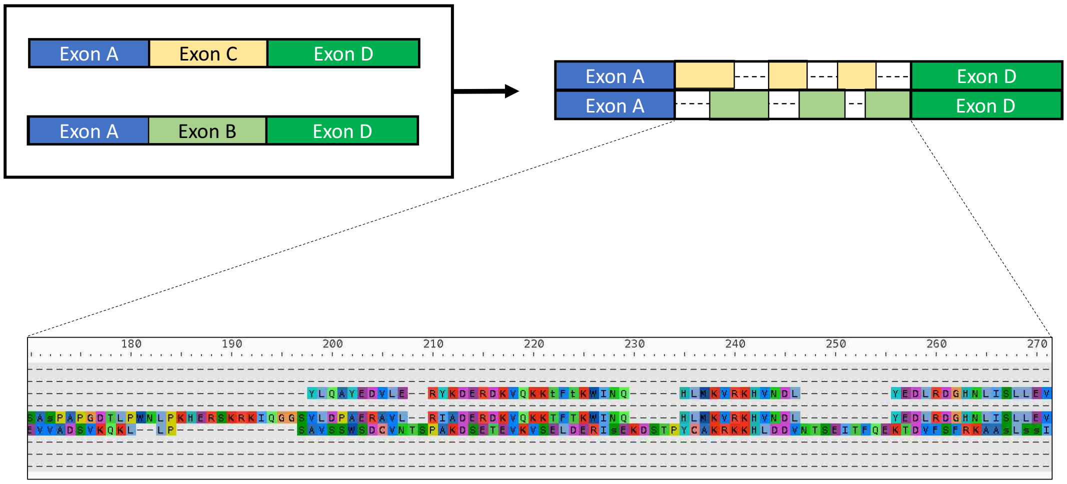
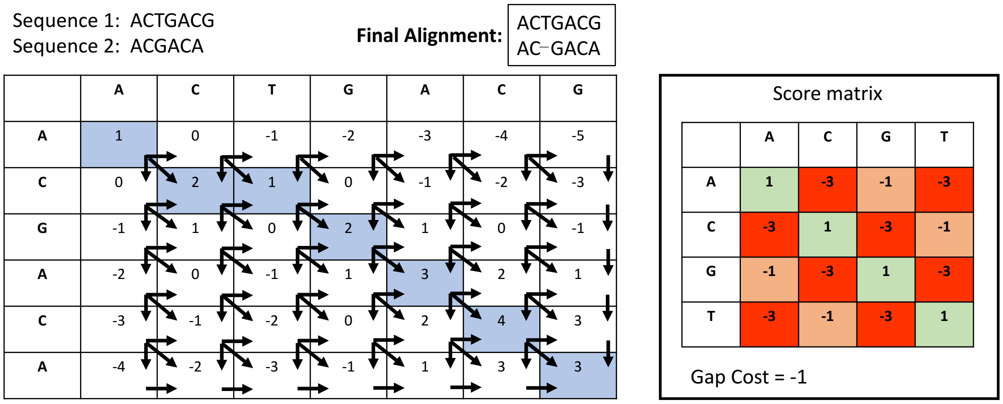

Exon A
Exon C
Exon D
Exon A
Exon B
Exon D
Exon A
Exon D
Exon A
Exon D
180
190
200
210
220
230
240
250
260
270
-------------------------YLQAYEDVLE--RYKDERDKVQKKtFtKWINQ-----HLMKVRKHVNDL--------YEDLRDGHNLISLLEV
SAsPAPGDTLPWNLPKHERSKRKIQGGSVLDPAERAVL--RIADERDKVQKKTFTKWINQ-----HLMKVRKHVNDL--------YEDLRDGHNLISLLEV
EVVADSVKQKL--LP-----------SAVSSWSDCVNTSPAKDSETEVKVSELDERIsEKDSTPYCAKRKKHLDDVNTSEITFQEKTDVFSFRKAASLssI

Figure 1.5: A model displaying how traditional MSA tools align alternatively spliced proteins in a way that falsely communicates alternative exon homology, with a real-world example from an alignment of human BPAG1 sequences produced by MUSCLE.

ters are aligned to one another (matches), whereas vertical movements represent insertions relative to the sequence at the top and horizontal movements represent insertions relative to the sequence on the side (gaps). Once the table has been completely filled, the best possible alignment of the full input sequences is revealed by tracing a path backwards through the table that follows the dependencies that yielded the final cell's score.

Prior to the advent of high-throughput sequencing and gene indexing, when large bodies of biological data were unavailable to researchers, a sort of biological agnosticism about the input sequences was a necessity for any broadly applicable MSA tool. Thanks to tools such as genome annotations [8] and database-integrated sequence similarity search software [16], however, it is now feasible to approach the problem of protein multiple sequence alignment with the assistance of complementary data. Specific to the problem of aligning alternatively spliced proteins, high-quality reference genomes can be combined with gene indexing files or splice-aware single-sequence alignment tools to map protein sequences to their encoding genomic DNA, and produce MSAs based on the relative genomic positions to which indi-



Sequence 1: ACTGACG
Sequence 2: ACGACA

Final Alignment:
ACTGACG
AC–GACA

	A	C	T	G	A	C	G
A	1	0	-1	-2	-3	-4	-5
C	0	2	1	0	-1	-2	-3
G	-1	1	0	2	1	0	-1
A	-2	0	-1	1	3	2	1
C	-3	-1	-2	0	2	4	3
A	-4	-2	-3	-1	1	3	3

Score matrix

	A	C	G	T
A	1	-3	-1	-3
C	-3	1	-3	-1
G	-1	-3	1	-3
T	-3	-1	-3	1

Gap Cost = -1

Figure 1.6: An illustration of the Needleman-Wunsch dynamic programming algorithm for the single DNA sequence inputs ‘ACTGACG’ and ‘ACGACA.’

vidual amino acids mapped. Genomic mapping has previously only been used for protein sequence alignment in the context of gene prediction, and to the best of our knowledge only with the intention of removing uncommon isoforms from consideration when labeling new sequences. Leveraging genomic data for the primary purpose of aligning protein sequences represents a promising new avenue for improving multiple protein sequence alignments of alternatively spliced protein products.

Presented here is Mirage (Multiple Isoform Alignment Tool Guided by Exon Boundaries), a novel MSA software package that transitively aligns protein sequences according to the genomic positions of their constitutive exons (*i.e.*, through protein-to-genome mapping). Protein-to-genome mapping, wherein each amino acid in a given protein sequence is assigned to the codon triple on its species genome that most likely encoded it, allows Mirage to recognize the underlying exonic structure of protein isoforms and use this information to align isoforms in a way that preserves homology on the level of exons. As my results demonstrate, this approach successfully addresses the fundamental challenge to multiple protein sequence alignment that alternatively utilized exons pose, thus facilitating more accurate information transfer between protein isoforms than has been possible with existing alignment tools. Improved annotation of post-translational modifications and other protein-level biological phenomena will prove an exciting and direct impact of integrating Mirage into annotation pipelines.

Mirage's novel approach to protein sequence alignment provides the benefits of a theoretical guarantee that intra-species MSAs are correct (not just heuristically optimal) and a marked improvement in inter-species MSA quality, attributable to Mirage's ability to use splice site data as an additional source of information during inter-species MSA production. In addition to improving the quality of protein isoform alignments, Mirage is uniquely able to detect and annotate "alternative reading frames," a special category of alternative splice site usage whereby different reading frames of the same genomic sequence are used to encode substantially different peptides.

Mirage is designed to process large protein datasets in bulk batches, and is thus expected to appeal to groups who host and curate databases (such as PhosphoSite) moreso than small research labs. Of course, the information made available to small groups through large data resources is often invaluable for their research, so the improvements that Mirage will make to multiple protein sequence alignments provided through datacenters should interest any researchers who interact with larger databases. Moreover, one of the eventual aims of Mirage development is the addition of functionality for integrating new sequences into existing Mirage MSAs or rapidly constructing splice-aware MSAs for small sets of protein sequences, thus making Mirage more immediately helpful for groups researching specific gene families. Mirage is algorithmically tailored to improve the quality of isoform alignments, and, given the far-reaching benefits of improved isoform alignments, there is every reason to make Mirage as widely useful as possible.

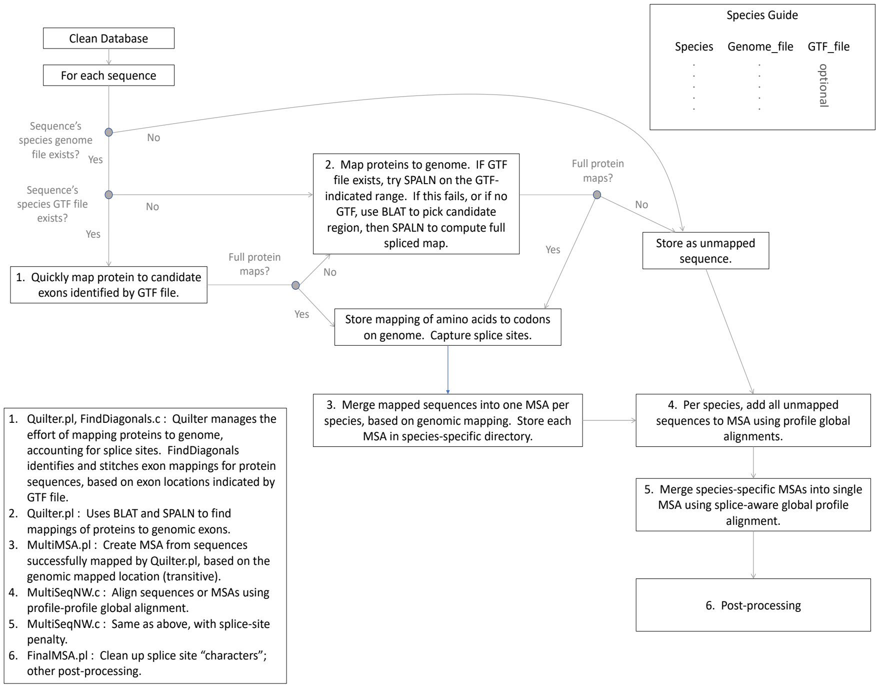
CHAPTER 2 METHODS AND MATERIALS

2.1 Program Input

The input to Mirage consists of a FASTA-formatted database and a mapping file that associates individual species with a FASTA-formatted genome, along with an optional GTF-formatted gene index file. Mirage imposes a specific naming convention to facilitate the recognition of the species and gene family to which a protein belongs, whereby the name of each protein sequence is a '|'-separated list with the third element being the species and the final element being the gene family (*e.g.*, "GN:KMT2E|MLL5|human|Q8IZD2|310949|MLL5" names a human isoform of the MLL5 gene). Mirage also enforces several minor character restrictions required for encoding lists as strings, and a "CleanMirageDB" script is included in the Mirage package to confirm properly formatted protein database and highlight naming problems. Each line of the mapping file is a whitespace-separated triple where the first element is a species name, the second element is a path to that species' genome file, and the third element is a path to that species' gene index file (or a '-' if no gene index is provided). Sequences whose species are not listed in the mapping file will still be aligned, but using a traditional dynamic programming method instead of Mirage's transitive alignment approach.

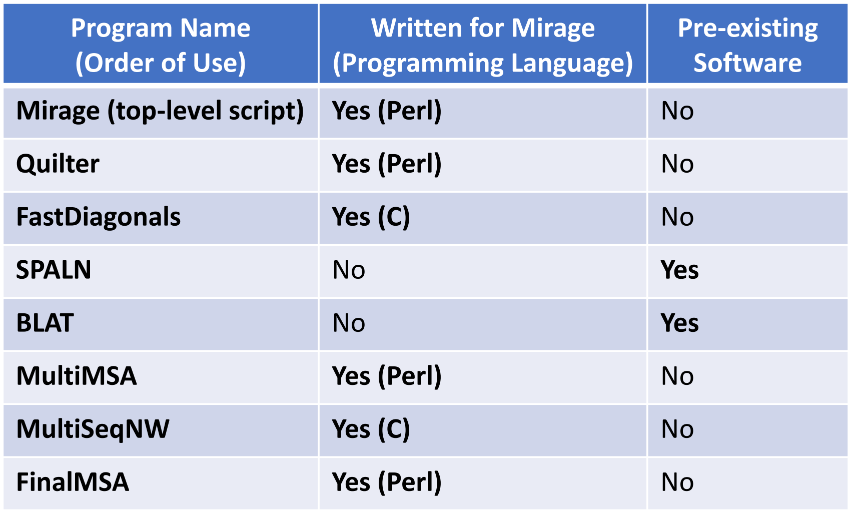
Mirage's naming constraints for protein sequences are partly used to ensure that it can accurately recognize corresponding entries for each sequence's gene family in the GTF-formatted gene index file. GTF files identify, for a given species, where on that species' reference genome particular proteins are believed to be encoded, along with a variety of other data. These entries indicate, for a multitude of exons, the gene family to which an exon belongs, the chromosome on which that exon resides, and the range of nucleotide positions on that chromosome that consititute the exon. While the exons enumerated in a GTF file are oftentimes computationally predicted and thus potentially attributable to computational errors, Mirage's use of annotated exons involves a type of cross-validation whereby only exons that can be perfectly incorporated into a full-protein mapping can inform the final MSAs, reducing the potential for poorly predicted exons to affect Mirage's accuracy.

Below, we describe the steps taken by Mirage to turn these three input files into a set of MSAs representing each gene family, following the pipeline illustrated in Figure 2.1. Additionally, a breakdown of which Mirage components were developed specifically for Mirage and which are existing tools integrated into the Mirage pipeline is provided in Figure 2.2.


Species Guide
Species
Genome_file
GTF_file
optional
Clean Database
For each sequence
Sequence's species genome file exists?
No
Yes
Sequence's species GTF file exists?
No
Yes
2. Map proteins to genome. IF GTF file exists, try SPALN on the GTF-indicated range. If this fails, or if no GTF, use BLAT to pick candidate region, then SPALN to compute full spliced map.
Full protein maps?
No
Yes
Store as unmapped sequence.
1. Quickly map protein to candidate exons identified by GTF file.
Full protein maps?
No
Yes
Store mapping of amino acids to codons on genome. Capture splice sites.
3. Merge mapped sequences into one MSA per species, based on genomic mapping. Store each MSA in species-specific directory.
4. Per species, add all unmapped sequences to MSA using profile global alignments.
5. Merge species-specific MSAs into single MSA using splice-aware global profile alignment.
6. Post-processing

1. Quilter.pl, FindDiagonals.c : Quilter manages the effort of mapping proteins to genome, accounting for splice sites. FindDiagonals identifies and stitches exon mappings for protein sequences, based on exon locations indicated by GTF file.
2. Quilter.pl : Uses BLAT and SPALN to find mappings of proteins to genomic exons.
3. MultiMSA.pl : Create MSA from sequences successfully mapped by Quilter.pl, based on the genomic mapped location (transitive).
4. MultiSeqNW.c : Align sequences or MSAs using profile-profile global alignment.
5. MultiSeqNW.c : Same as above, with splice-site penalty.
6. FinalMSA.pl : Clean up splice site "characters"; other post-processing.

Figure 2.1: Wire-frame diagram of the Mirage pipeline.



Program Name (Order of Use)	Written for Mirage (Programming Language)	Pre-existing Software
Mirage (top-level script)	**Yes (Perl)**	No
Quilter	**Yes (Perl)**	No
FastDiagonals	**Yes (C)**	No
SPALN	No	**Yes**
BLAT	No	**Yes**
MultiMSA	**Yes (Perl)**	No
MultiSeqNW	**Yes (C)**	No
FinalMSA	**Yes (Perl)**	No

Figure 2.2: List of Mirage components, identifying whether they were written for Mirage or are pre-existing tools.

2.2 Translated Mapping

Mirage's first task is to map each protein sequence to its genome, which is handled by the Perl script "Quilter." Quilter iterates over the species listed in the mapping file, considering each species independently in order to avoid the large memory overhead of simultaneously storing the contents of every species' GTF file. Quilter begins by scanning all of the protein names for the given species and compiling a list of present gene families. Using this list, Quilter then uses the "exon" entries of the GTF file to construct a hash table mapping each gene family to a set of coding regions on the genome. These coding regions are organized by chromosome and strand direction, so only biologically consistent mappings are possible.

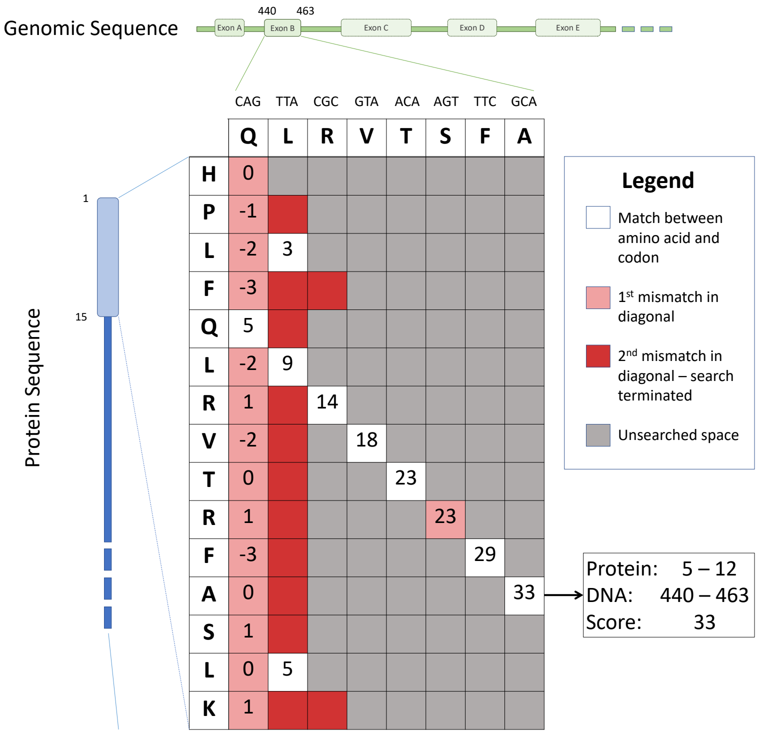
Following construction of the hash table, Quilter iterates over the protein sequences and maps each to the genome using one of three methods:

2.2.1 Fast GTF-based Mapping

The preferred method relies on a C program named “FastDiagonals” that rapidly aligns protein peptides to the specific exons indicated by the GTF file. Quilter provides the protein and coding DNA sequences in single-sequence FASTA files as inputs to FastDiagonals, which immediately reads them into memory. Each of the three forward reading frames (with strand orientation based on GTF annotation) are iteratively translated into amino acid sequences and searched against the protein for full-exon alignments with no more than 1 mismatch. FastDiagonals requires each partial mapping of the protein to cover the full length of a GTF-indicated exon because we assume that if an annotated exon is genuinely one of the exons used to encode that protein, then the splicing machinery would have conformed to that exon’s splice signalling and, by doing so, provide the full exon for translation. This also greatly simplifies the combinatorics associated with identifying an optimal full-protein mapping to the genome by significantly reducing the number of codons in consideration for possibly encoding each amino acid.

The FastDiagonals mapping procedure begins with a “seeding” step where the first two translated amino acids are searched against the full protein using a gapless dynamic programming method and only those parts of the protein with at least one matching amino acid are preserved as seeds. Each seed is stored as a tuple consisting of a starting position in the protein sequence and a score, and a global counter tracks the length of all extending seeds. Seeds are extended through the length of the translated exon until they either accumulate 2 mismatches and are discarded, or else successfully align to the full length of the exon. All successful mappings are scored with the half-bit BLOSUM62 match scores for the protein-translated DNA alignment, and this score, along with the corresponding protein and genome positions, are returned to Quilter. Figure 2.3 illustrates the FastDiagonals algorithm.

After all of the indexed exons have been examined by FastDiagonals, Quilter will have a set of partial translated mappings characterized by their protein ranges, DNA ranges,



	CAG	TTA	CGC	GTA	ACA	AGT	TTC	GCA
	Q	**L**	**R**	**V**	**T**	**S**	**F**	**A**
H	0	▨	▨	▨	▨	▨	▨	▨
P	-1	🟥	▨	▨	▨	▨	▨	▨
L	-2	3	▨	▨	▨	▨	▨	▨
F	-3	🟥	🟥	▨	▨	▨	▨	▨
Q	5	🟥	▨	▨	▨	▨	▨	▨
L	-2	9	▨	▨	▨	▨	▨	▨
R	1	🟥	14	▨	▨	▨	▨	▨
V	-2	🟥	▨	18	▨	▨	▨	▨
T	0	🟥	▨	▨	23	▨	▨	▨
R	1	🟥	▨	▨	▨	23	▨	▨
F	-3	🟥	▨	▨	▨	▨	29	▨
A	0	🟥	▨	▨	▨	▨	▨	33
S	1	🟥	▨	▨	▨	▨	▨	▨
L	0	5	▨	▨	▨	▨	▨	▨
K	1	🟥	🟥	▨	▨	▨	▨	▨

Legend

□ Match between amino acid and codon

1st mismatch in diagonal

🟥 2nd mismatch in diagonal – search terminated

▨ Unsearched space

Protein: 5 – 12
DNA: 440 – 463
Score: 33

Figure 2.3: An illustration of how FastDiagonals identifies candidate exons for Quilter's protein-to-genome mapping.

and scores. Using a straightforward depth-first search algorithm, Quilter attempts to find a set of partial mappings that can be stitched together to cover the full length of the protein in a biologically consistent way (*i.e.*, the mapping uses a collection of mapped exons to completely cover the protein sequence, the mapped exons are sourced from the same chromosome, in the same direction on the chromosome, such that the relative positions of the exons on the genome correspond to the relative positions of the peptides that they encode on the protein). This algorithm (modeled in Figure 2.4) conceptually treats each peptide-to-exon alignment as a node in a graph, and draws a directed edge from every node that ends with the ith amino acid of the protein to every node that begins with the $i + 1$th amino acid, with edges being weighted by the score of the partial mapping. Quilter then traverses the graph in a depth-first manner, looking for the highest scoring path that covers the full length of the protein and recording the best observed scores at each visited node to avoid unnecessary computation.

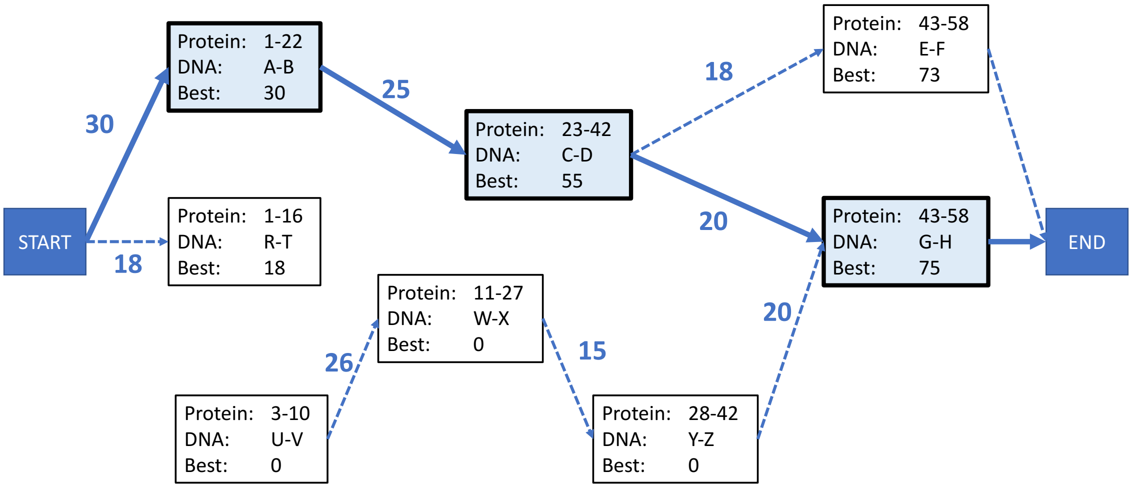


Figure 2.4: Conceptual diagram of the graph algorithm used by Quilter to identify an optimal splice-aware protein-to-genome mapping.

Once Quilter finds an optimal set of partial mappings that can be stitched together to cover the full length of the protein, it writes that mapping out to a file that includes the sequence name, the method by which the hit was found, the chromosome and direction of the hit, and a list of nucleotide positions that index the centers of each codon used in the full-protein mapping. Figure 2.5 broadly illustrates the path from FastDiagonals output to recording a mapping.

Because the full-length mapping exists as an ordered series of exon mappings, introns are implied to exist between each partial mapping. The mapping file makes these introns explicit by placing a ‘∗’ (Mirage’s splice-site character) at each break between two partial mappings in the nucleotide list, enabling Mirage to take splice-sites into consideration during subsequent stages of the alignment pipeline.

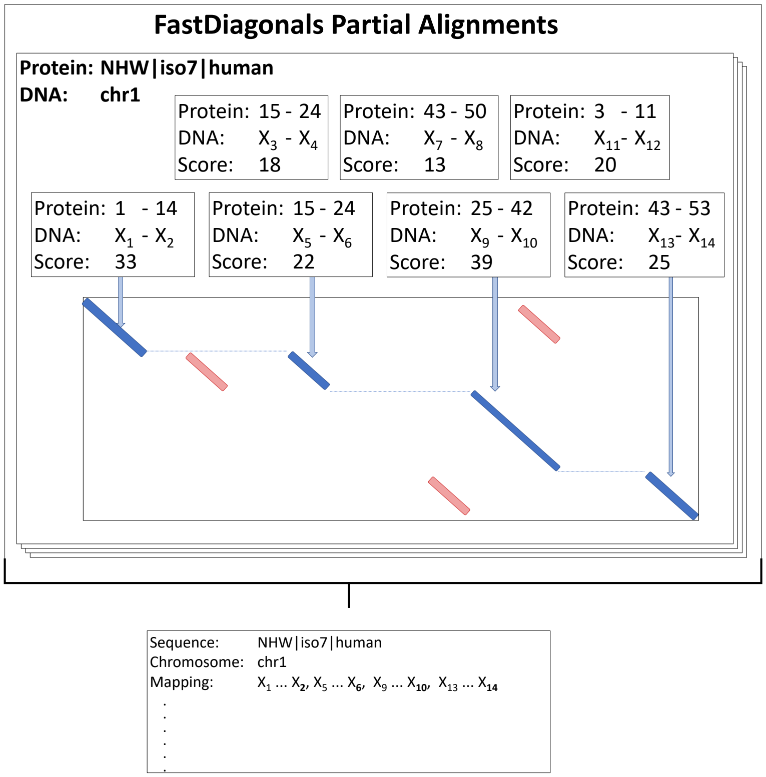

FastDiagonals Partial Alignments
Protein: NHW|iso7|human
DNA: chr1
Protein: 15 - 24 DNA: X_3 - X_4 Score: 18
Protein: 43 - 50 DNA: X_7 - X_8 Score: 13
Protein: 3 - 11 DNA: X_{11}- X_{12} Score: 20
Protein: 1 - 14 DNA: X_1 - X_2 Score: 33
Protein: 15 - 24 DNA: X_5 - X_6 Score: 22
Protein: 25 - 42 DNA: X_9 - X_{10} Score: 39
Protein: 43 - 53 DNA: X_{13}- X_{14} Score: 25
Sequence: NHW|iso7|human
Chromosome: chr1
Mapping: X_1 … X_2, X_5 … X_6, X_9 … X_{10}, X_{13} … X_{14}

Figure 2.5: Illustration of how Quilter uses FastDiagonals output to identify full protein-to-genome mappings.

2.2.2 GTF-based Mapping Using SPALN

In the event that FastDiagonals cannot produce a set of partial mappings that can be stitched together to form a full-protein mapping, Quilter falls back on an external tool called SPALN [5] to generate a translated alignment of the protein to the genome. SPALN performs splice-aware translated sequence alignment (aligning protein sequence to chromosomal DNA sequence) by incorporating splice-site signal into its alignment algorithm; Quilter provides SPALN the protein sequence and a window (determined from the GTF file) of genomic sequence that contains the coding regions indicated by the GTF file. It then parses SPALN's output to extract a full-protein mapping to the genome. In the event that the SPALN does not succeed on its first mapping attempt, the cause is often that a very long intron separates one or more exons from the region suggested by the GTF file. To overcome this concern, the window of genomic sequence is extended 400Kb in each direction and SPALN search is repeated with this larger input.

2.2.3 Mapping Using BLAT+SPALN

Quilter's final fallback, and the method used for species without an associated GTF file, is to use the fast sequence similarity search tool BLAT [7] to identify the location of an alignment seed that suggests a region of the genome that contains the DNA encoding the protein sequence. SPALN is then used to search for a splice-aware alignment of the protein to that region of the genome, and, similarly to the GTF-assisted SPALN runs, the indicated genomic region is extracted with successively larger windows of surrounding the seed location (100Kb, 1Mb, and 10Mb in each direction) until a high-quality full-protein mapping is identified. Figure 2.6 illustrates how this combination of BLAT and SPALN is used to identify a coding region of the genome for a given protein and produce a splice-aware mapping of the protein to that region.

If a protein still fails to map to the genome after pulling in a 10Mb window around the BLAT-indicated region, it is designated as a "miss" and its name is added to a file listing all sequences that failed to map and will be incorporated into their gene family alignments using a dynamic programming approach. Thus, for every protein sequence belonging a species with an associated genome, Quilter will either "hit" by mapping each amino acid in the protein to a codon's central nucleotide or else identify that sequence as a "miss."

2.2.4 Quilter Parallelism

Because the input sequences to Quilter are considered independently from one another, Quilter employs a straightforward "embarrassingly parallel" parallelism whereby each process is given an equal fraction of the input database and is responsible for mapping all sequences beginning in its fraction. Each of Quilter's parallel processes write their results to private "hit" and "miss" files, which is concatenated into final hit and miss files by

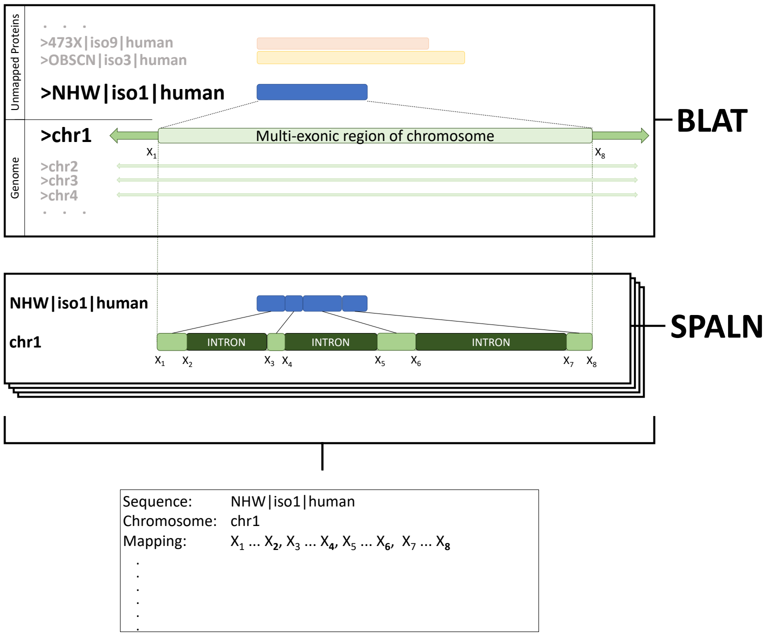


Figure 2.6: BLAT identifies probable coding regions for unmapped proteins, which SPALN searches to identify a protein-to-genome mapping.

the master process once all of the child processes have terminated. To ensure that each process has a roughly equal workload, Mirage generates temporary species-specific protein databases prior to running Quilter so that an uneven distribution of species in the full protein database will not negate the efficiency of using parallel processes.

2.3 Transitive Alignment

Once Quilter has finished mapping proteins to their genomes, Mirage uses the Perl script “MultiMSA” to construct an intra-species MSA for every gene family with at least one member that successfully mapped to the genome. The inputs to MultiMSA are a mapping file produced by Quilter and the protein sequence database. MultiMSA begins by reading the full mapping file and placing each entry (consisting of a sequence name, chromosome and direction, and the mapping itself) in a hash table using gene family names as keys. This

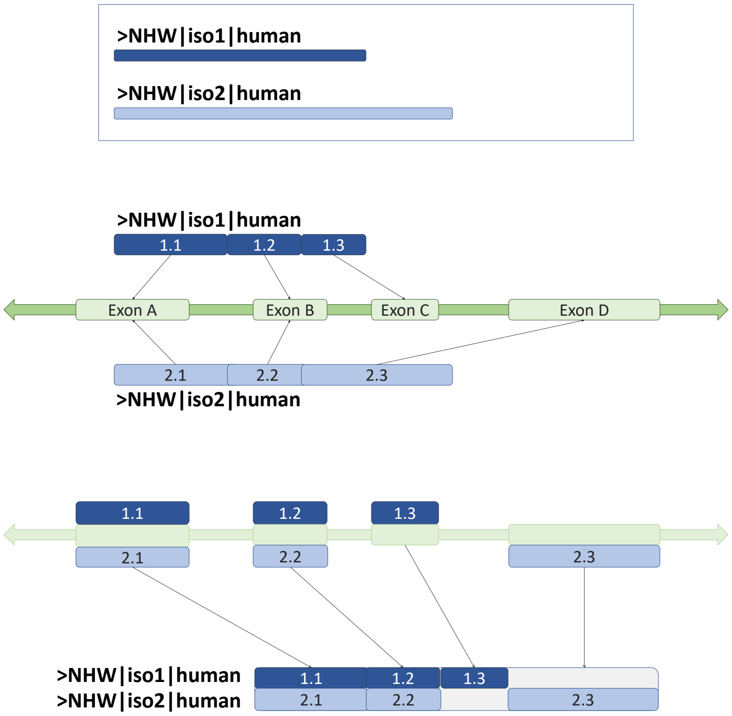
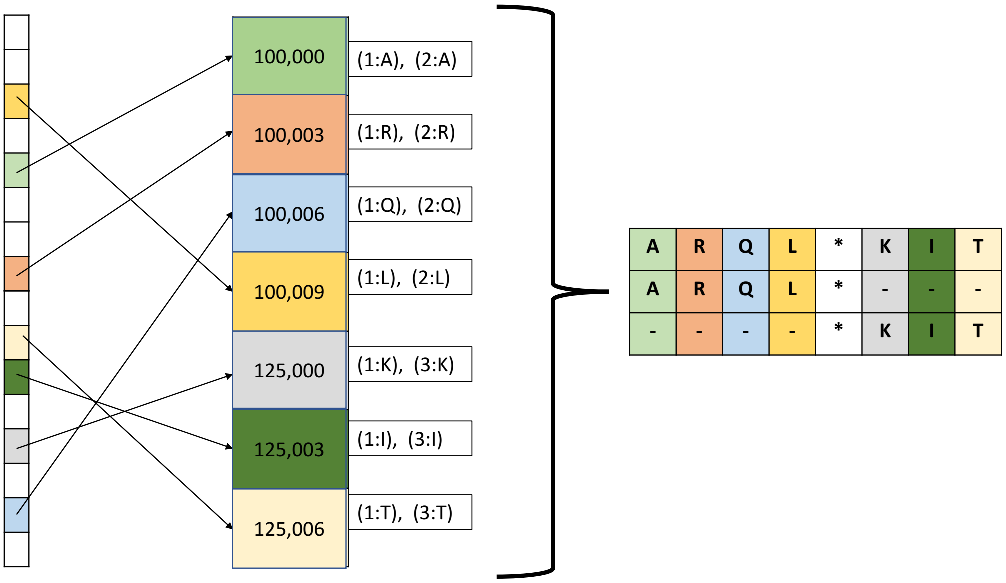


Figure 2.7: A conceptual illustration of transitive alignment, where two protein sequences are individually aligned to the genome and subsequently aligned to one another based on their genome mappings.

hash table places a moderately large memory overhead on MultiMSA but pays for itself by avoiding the massive speed penalty that would be incurred by scanning the full mapping file once per gene family. Once this guiding hash table has been constructed, MultiMSA iterates over the gene families transitively aligning the genome-mapped sequences.

Each of MultiMSA's transitive alignments are built using a hash table where the keys are nucleotide indices and each entry in the table is a list of pairs that consists of a numeric sequence identifier (*i.e.*, the sequence's row number in the MSA) and the amino acid character from that sequence associated with the key nucleotide. Once every mapped sequence from a gene family has been added to this hash table, the keys are sorted (ascending for forward strand, descending for reverse complement) and traversed in order. Each entry in the hash table converts naturally into an MSA column (as depicted in figure 13)—the identifier-character pairs communicate which characters will be aligned in which rows, and gap-characters are placed in every row that is not represented—and thus this ordered traversal of the hash table allows for quick MSA construction. Whenever the difference between adjacent keys (*i.e.*, nucleotide indices) is larger than 3 (the length of a codon), a splice junction is inferred and a column consisting entirely of splice-site characters is appended to the MSA to represent an intron. Finally, terminal splice-site columns are added to both ends of the final MSA, so that every exon is flanked by a column of splice-site characters, and the MSA is written to a file in a species-specific directory.



A	R	Q	L	*	K	I	T
A	R	Q	L	*	-	-	-
-	-	-	-	*	K	I	T

Figure 2.8: Illustration of how sorted hash keys allow MultiMSA to naturally construct a transitive multiple protein sequence alignment.

The data independence between the individual gene families allows for straightforward process parallelization similar to Quilter by evenly dividing the gene families across the processes.

2.4 Aligning Sequences that Did Not Map to a Genome

After MultiMSA has finished constructing the transitive MSAs for a species, Mirage iteratively aligns any sequences that failed to map to their genomes to their species' gene family MSAs through a sequence-to-profile alignment method based on the classic Needleman-Wunsch sequence alignment algorithm [14]. The C program "MultiSeqNW" is used at this stage of the Mirage pipeline, and takes as input two FASTA-formatted files, of which one is the gene family alignment produced by MultiMSA and the other is the unmapped sequence. Matches between profile columns are scored using half-bit BLOSUM62 scores and heterogeneous profile columns are scored proportionally to their non-gap character compositions (*e.g.*, aligning a profile column "R-RK-" to a single sequence character "R" scores 2/3(RR) + 1/3(KR)). Figure 2.9 depicts one iteration of sequence-to-profile alignment using a dynamic programming method.

Specific to Mirage's transitive alignment method, splice-characters have a match score of zero when aligned to one another and a match score of negative infinity when aligned to non-splice-characters, guaranteeing that an amino acid can never be aligned to an intron. Affine gap penalties are used so that the penalty for beginning a gap (-11) is greater than the penalty for extending a gap (-1 per extension). Starting a gap at a splice site incurs a gap-start penalty that scales with the distance to the closest splice site in the other profile (-5 times 2 to the power of the distance to the closest splice site, with a maximum penalty of -200), unless the other profile is a single sequence with no splice markers (*i.e.*, failed to map to the genome), in which case there is no additional penalty for starting a gap at a splice site column. These special splice site considerations preserve the splice-awareness of MultiMSA's transitive alignments by encouraging MultiSeqNW to align introns to one another and maintain strong exon delineation, while enjoying from the greater generality that comes with taking a dynamic programming approach to multiple protein sequence alignment.

In addition to aligning all unmapped proteins to their species-specific gene family MSAs,

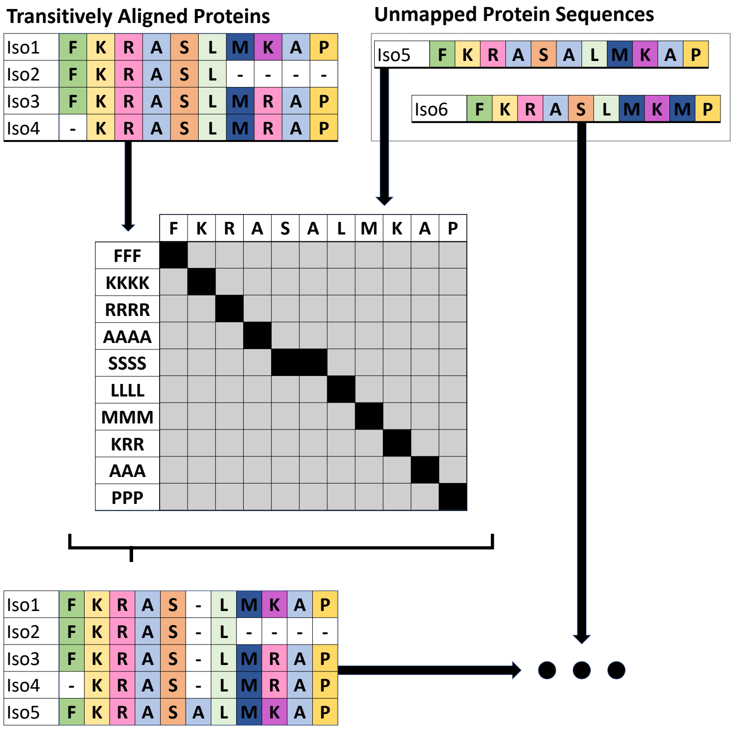
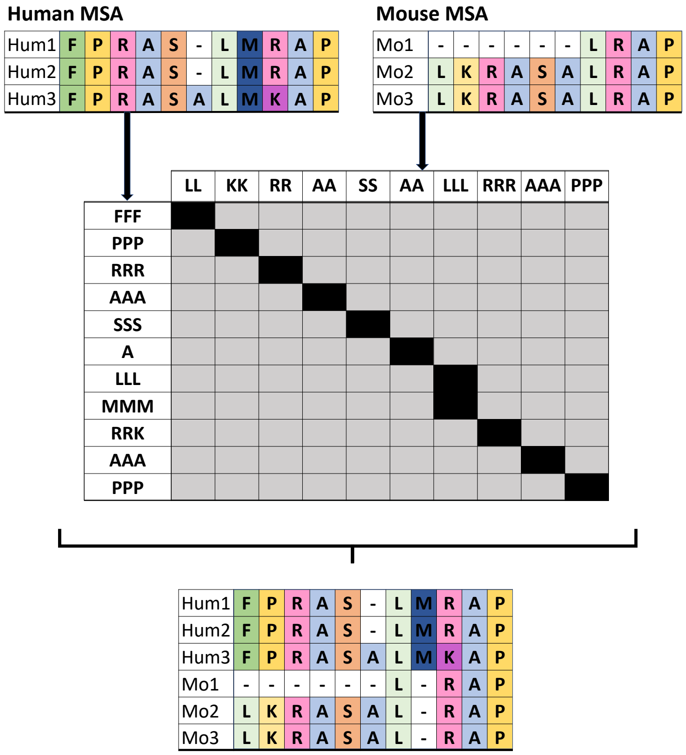

Transitively Aligned Proteins
Iso1 F K R A S L M K A P
Iso2 F K R A S L - - - -
Iso3 F K R A S L M R A P
Iso4 - K R A S L M R A P
Unmapped Protein Sequences
Iso5 F K R A S A L M K A P
Iso6 F K R A S L M K M P
F K R A S A L M K A P
FFF
KKKK
RRRR
AAAA
SSSS
LLLL
MMM
KRR
AAA
PPP
Iso1 F K R A S - L M K A P
Iso2 F K R A S - L - - - -
Iso3 F K R A S - L M R A P
Iso4 - K R A S - L M R A P
Iso5 F K R A S A L M K A P
• • •

Figure 2.9: The iterative dynamic programming procedure Mirage uses to incorporate unmapped protein sequences into their gene families' MSAs.

MultiSeqNW is also used to generate gene family MSAs for every species that was not provided a genome. Through the use of MultiSeqNW, Mirage is thus able to incorporate every protein sequence from the input database into a species-specific gene family MSA, regardless of that protein's ability to map to a genome.

2.5 Inter-Species Alignment and Finalization

Once every sequence from the input database has been incorporated into a species-specific gene family MSA, Mirage's final task is aligning gene family MSAs across species to produce its final splice-aware multiple protein sequence alignments. Once again, process parallelism is employed and gene families are divided equally across processes which iteratively use MultiSeqNW to merge gene families' species-specific MSAs with their final MSAs, as illustrated in Figure 2.10.



Human MSA

Hum1	F	P	R	A	S	-	L	M	R	A	P
Hum2	F	P	R	A	S	-	L	M	R	A	P
Hum3	F	P	R	A	S	A	L	M	K	A	P

Mouse MSA

Mo1	-	-	-	-	-	-	L	R	A	P
Mo2	L	K	R	A	S	A	L	R	A	P
Mo3	L	K	R	A	S	A	L	R	A	P

	LL	KK	RR	AA	SS	AA	LLL	RRR	AAA	PPP
FFF	■									
PPP		■								
RRR			■							
AAA				■						
SSS					■					
A						■				
LLL							■			
MMM							■			
RRK								■		
AAA									■	
PPP										■

Hum1	F	P	R	A	S	-	L	M	R	A	P
Hum2	F	P	R	A	S	-	L	M	R	A	P
Hum3	F	P	R	A	S	A	L	M	K	A	P
Mo1	-	-	-	-	-	-	L	-	R	A	P
Mo2	L	K	R	A	S	A	L	-	R	A	P
Mo3	L	K	R	A	S	A	L	-	R	A	P

Figure 2.10: MultiSeqNW is used to produce inter-species MSAs.

After every species-specific MSA for a gene family has been merged into that family's final inter-species MSA, a cleanup script is used to remove splice site markers and make minor aesthetic corrections to the alignment (*e.g.*, merging complementary amino/gap columns from opposite sides of a splice site). The resulting inter-species MSAs are stored as AFA files in a directory named "FinalMSAs."

2.6 Investigating Alternative Reading Frames

Exons typically consist of a single open uninterrupted run of codon triples (*i.e.*, an open reading frame, ORF) flanked by splice sites, but occasionally a gene will have one or more associated exons that encode amino acids in two or three open reading frames that are shifted off one another by a single nucleotide.

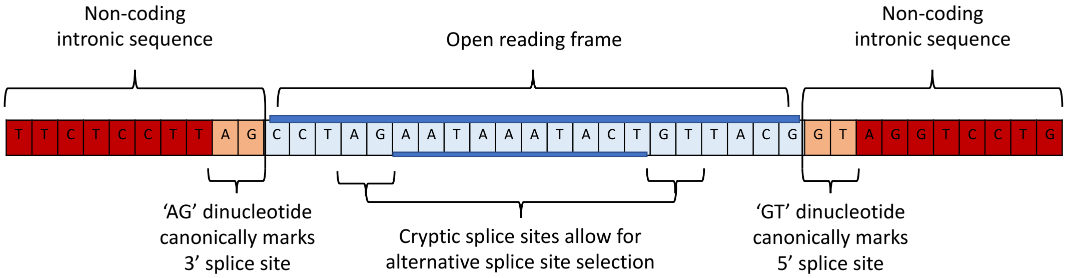

Non-coding intronic sequence
Open reading frame
Non-coding intronic sequence
T T C T C C T T A G C C T A G A A T A A A T A C T G T T A C G G T A G G T C C T G
'AG' dinucleotide canonically marks 3' splice site
Cryptic splice sites allow for alternative splice site selection
'GT' dinucleotide canonically marks 5' splice site

Figure 2.11: Model illustrating a standard open reading frame flanked by canonical 'AG/GT' splice sites and containing cryptic splice sites that would allow for alternative splicing.

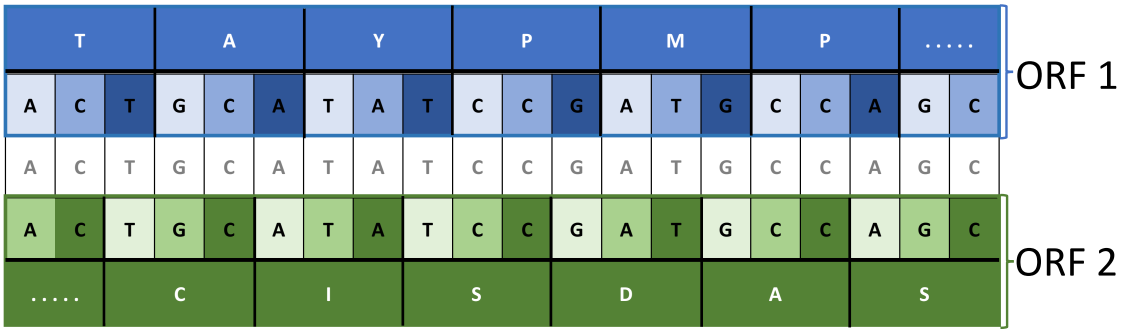

T A Y P M P .....
A C T G C A T A T C C G A T G C C A G C
ORF 1
A C T G C A T A T C C G A T G C C A G C
A C T G C A T A T C C G A T G C C A G C
..... C I S D A S
ORF 2

Figure 2.12: Diagram modeling how multiple open reading frames, and thus different proteins, can be encoded by the same DNA sequence to produce alternative reading frames.

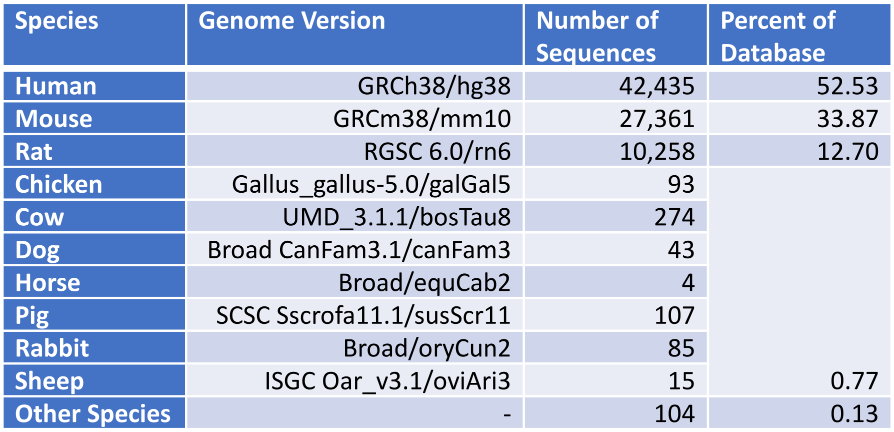
These alternative reading frames (ARFs) are a curious form of biological efficiency and little is understood about them [9], partly due to their difficulty to detect using existing tools. As amino acid sequences, protein isoforms with ARF exons look exactly like the more common alternative splicing phenomenon of mutually-exclusive exons, and thus character-based analyses are incapable of annotating occurrences of ARFs. Because ARFs share coding nucleotides, however, Mirage is able to recognize candidate ARFs as overlapping frame-shifted peptides while constructing its transitive intra-species MSAs. Whenever Mirage detects a candidate ARF it determines which of the overlapping reading frames is less frequently observed (distinguishing between the “alternative” and “standard” reading frames) and records the candidate ARF’s amino acid indices in its sequence’s name field. Mirage thus highlights putative ARF-containing sequence for future analysis, warranted because many protein sequences are generated using predictive software rather than direct observation (*e.g.*, using mass spectrometry data), such that some putative ARFs may simply be software artifacts.

One method for validating putative ARFs is to examine whether the corresponding genomic sequence in a species different from the one in which the ARF was identified also displays overlapping open reading frames. Comparing genomic sequence between divergent species is possible through the use of “lift-over” files from the UCSC genome browser [8], which encode alignments between reference genomes. Collaboration with Kaitlin Carey, a post-baccalaureate researcher in the Wheeler lab group, has led to the development of a pipeline for (1) recording ARF-associated DNA in one species, (2) identifying and extracting the corresponding DNA in a distantly related species, typically seeking putative human ARFs on the mouse genome, (3) translating each of the extracted DNA’s forward reading frames, and (4) reporting whether that DNA also encodes multiple ORFs. While further analysis will be needed to prove that translations of both reading frames occur in nature, the absence of point mutations that preserve only one of the reading frames for 90 million years (the time since the last common ancestor of humans and mice) [8] provides strong support for the notion that these ARFs have some selected function.

CHAPTER 3 RESULTS

3.1 Test Dataset and Hardware

Mirage's performance has been tested using a dataset comprised of 80,779 protein sequences from the UniprotKB database [17]. This dataset includes sequences representing 35 species and 21,980 gene families, of which 18,253 have at least 2 sequences attributed to them.



Species	Genome Version	Number of Sequences	Percent of Database
Human	GRCh38/hg38	42,435	52.53
Mouse	GRCm38/mm10	27,361	33.87
Rat	RGSC 6.0/rn6	10,258	12.70
Chicken	Gallus_gallus-5.0/galGal5	93	0.77
Cow	UMD_3.1.1/bosTau8	274	
Dog	Broad CanFam3.1/canFam3	43	
Horse	Broad/equCab2	4	
Pig	SCSC Sscrofa11.1/susScr11	107	
Rabbit	Broad/oryCun2	85	
Sheep	ISGC Oar_v3.1/oviAri3	15	
Other Species	-	104	0.13

Figure 3.1: Composition of the protein dataset by species.

We downloaded genomes for the 10 most prevalent species from the UCSC Genome Browser (downloaded from http://hgdownload.cse.ucsc.edu/downloads.html), and acquired gene indexing files for human, mouse, and rat from Ensembl (public release 87, located at ftp://ftp.ensembl.org/pub/release-87 and downloaded on 2/1/2017) [24]. Tests were run in a virtual Linux Ubuntu environment on a server housed at the University of Montana with 64 cores and 2 TB of shared RAM.

3.2 Assessing Alignment Quality

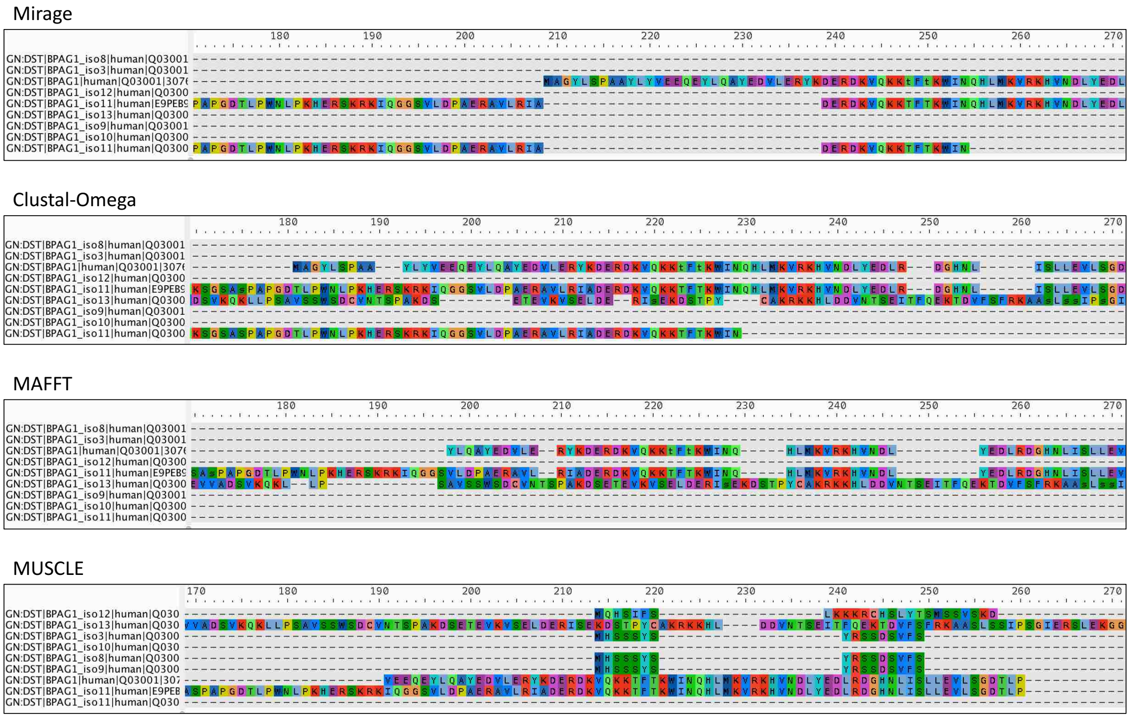
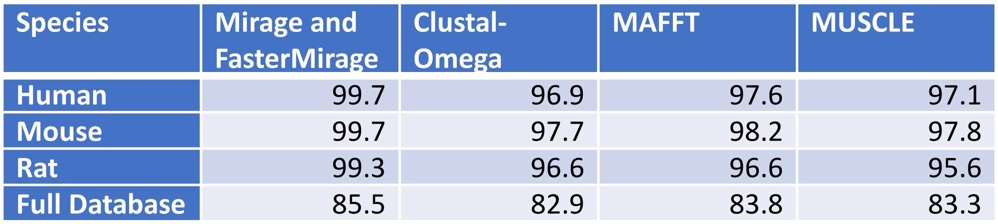


Figure 3.2: Comparable segments of the multiple sequence alignments for the BPAG1 gene family produced (from top to bottom) by Mirage, Clustal-Omega, MAFFT, and MUSCLE (MSA visualizations produced by AliView). Only human sequences are displayed.

Qualitative comparisons of Mirage's MSAs with MSAs produced by Clustal-Omega [19], MAFFT [6], and MUSCLE [1] (three of the most popular protein MSA tools) provide the best illustration that Mirage's splice-aware transitive alignments are far better at characterizing the similarities between protein isoforms than alignments produced using dynamic programming. Mirage clearly illuminates the exonic structures of protein sequences and intelligently recognizes homologous exons from evolutionarily divergent species in a way that cannot be expected from other MSA tools.

Multiple sequence alignments are notoriously difficult to quantitatively benchmark [2]. Percent identity is an intuitive quality metric when comparing MSAs of protein isoforms because all isoforms are derived from the same genomic source, so peptide differences must come from rare post-transcriptional modification (*e.g.*, A-to-I editing [12]), so correct alignments (especially within species) should consist almost entirely of identical columns. Comparing percents identity over all MSA columns, Mirage consistently outperforms its competitors, achieving near-100% column identity within intra-species alignments. Intra-species alignments generally display 100% column identity when transitively aligned by Mirage, so the incorporation of sequences that failed to map to the genome using dynamic programming explains why Mirage does not achieve 100% identity for its intra-species alignments.



Species	Mirage and FasterMirage	Clustal-Omega	MAFFT	MUSCLE
Human	99.7	96.9	97.6	97.1
Mouse	99.7	97.7	98.2	97.8
Rat	99.3	96.6	96.6	95.6
Full Database	85.5	82.9	83.8	83.3

Figure 3.3: Percents identity over all MSA columns for the three main species' intra-species alignments and for the full set of inter-species MSAs.

One measure of alignment accuracy is the density of exons in intra-species alignments. Theoretically, alignments of protein isoforms should exhibit long runs of ungapped amino acids or (where an exon has been excluded) long runs of gap characters, so the relative density of exons can serve as a proxy measurement of isoform alignment accuracy when information about exon boundaries is known, as illustrated in Figure 3.4.

To calculate relative exon density, we identified the amino acid index ranges for mapped

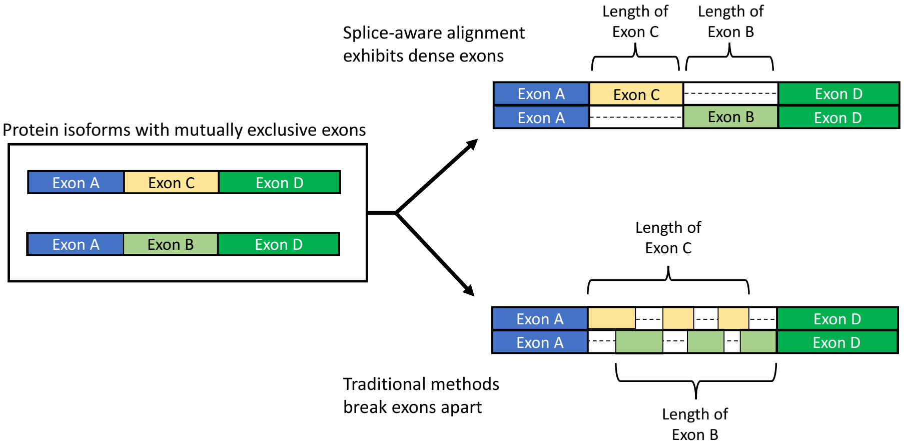
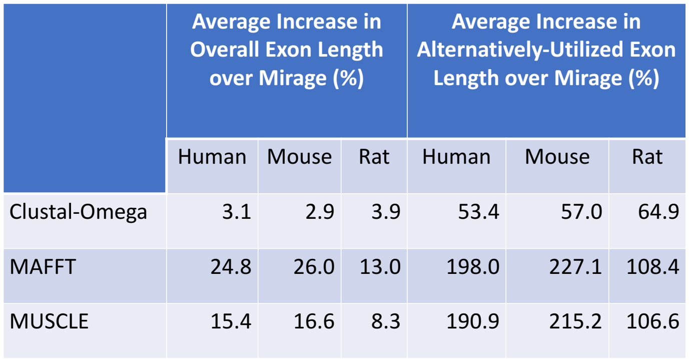


Figure 3.4: Incorrect alignments of protein isoforms will increase the distance between the first and last amino acids in cases of alternatively-utilized exons.

exons and computed the distances between those amino acids in Mirage intra-species alignments and other tools' intra-species alignments. Figure 3.5 displays the average percentage increases in observed exon length in intra-species MSAs with respect to Mirage MSAs across all exons and across alternatively-utilized exons (*i.e.*,, exons in the same gene family that are never incorporated into the same translated protein).



	Average Increase in Overall Exon Length over Mirage (%)			Average Increase in Alternatively-Utilized Exon Length over Mirage (%)		
	Human	Mouse	Rat	Human	Mouse	Rat
Clustal-Omega	3.1	2.9	3.9	53.4	57.0	64.9
MAFFT	24.8	26.0	13.0	198.0	227.1	108.4
MUSCLE	15.4	16.6	8.3	190.9	215.2	106.6

Figure 3.5: Average distance increase (as a percentage) between the first and last amino acids of exons in intra-species MSAs relative to Mirage alignments.

The ability to extract information about intron locations from Mirage intra-species alignments also allows us to examine how frequently traditional tools bleed exons into one another by tracking changes in splicing “pinch points,” where splice sites are flanked on both sides by non-gap characters in the same sequence, as illustrated in Figure 3.6. These sites indicate amino acids between which there is no coding DNA for the given gene family on the genome, and thus MSAs that either fail to align those amino acids or insert other characters between them may be thought of as incorrectly bleeding exons into one another.

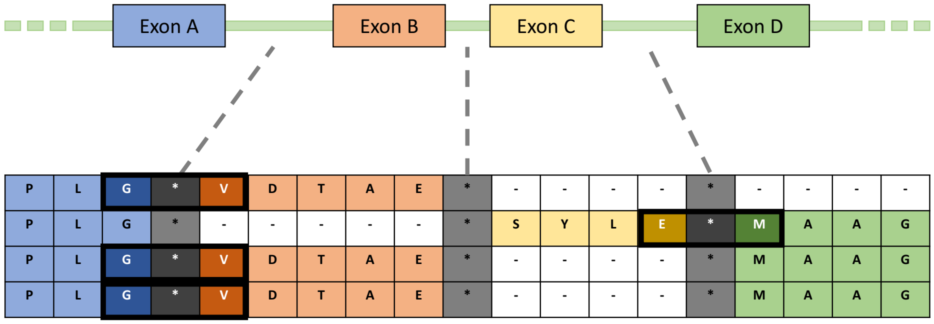
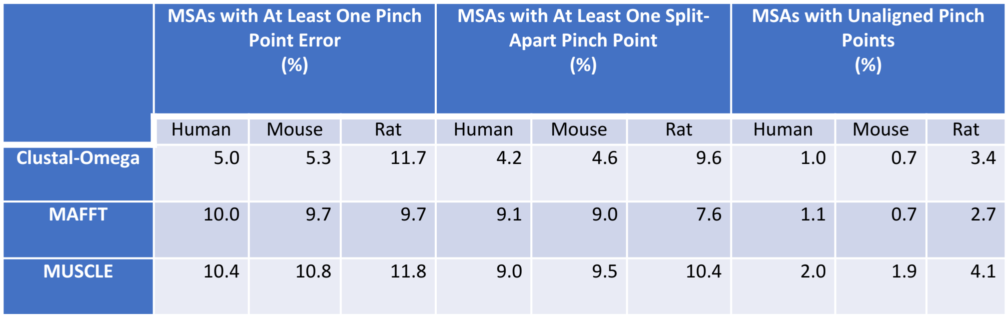

Exon A
Exon B
Exon C
Exon D
P L G * V D T A E * - - - - * - - - -
P L G * - - - - - * S Y L E * M A A G
P L G * V D T A E * - - - - * M A A G
P L G * V D T A E * - - - - * M A A G

Figure 3.6: Model illustrating splicing “pinch points.”

Recording the indices associated with “pinch” amino acids in Mirage intra-species alignments allowed us to track the frequency of errors associated with these sites in the intra-species MSAs generated by other tools, displayed in Figure 3.7. Pinch points that are “split-apart” have gap characters inserted between them, indicating that another sequence has been falsely aligned to the intronic sequence that separates the last and first amino acids of the exons. “Unaligned” pinch points are groups of pinch points from multiple sequences that should be aligned to one another (such as the three “GV” pinch amino acids in Figure 3.6) but were not correctly aligned. Unaligned pinch points, when incorrectly aligned to non-pinch point sequence, effectively place an intron in the middle of the other sequence’s exon.



	MSAs with At Least One Pinch Point Error (%)			MSAs with At Least One Split-Apart Pinch Point (%)			MSAs with Unaligned Pinch Points (%)		
	Human	Mouse	Rat	Human	Mouse	Rat	Human	Mouse	Rat
Clustal-Omega	5.0	5.3	11.7	4.2	4.6	9.6	1.0	0.7	3.4
MAFFT	10.0	9.7	9.7	9.1	9.0	7.6	1.1	0.7	2.7
MUSCLE	10.4	10.8	11.8	9.0	9.5	10.4	2.0	1.9	4.1

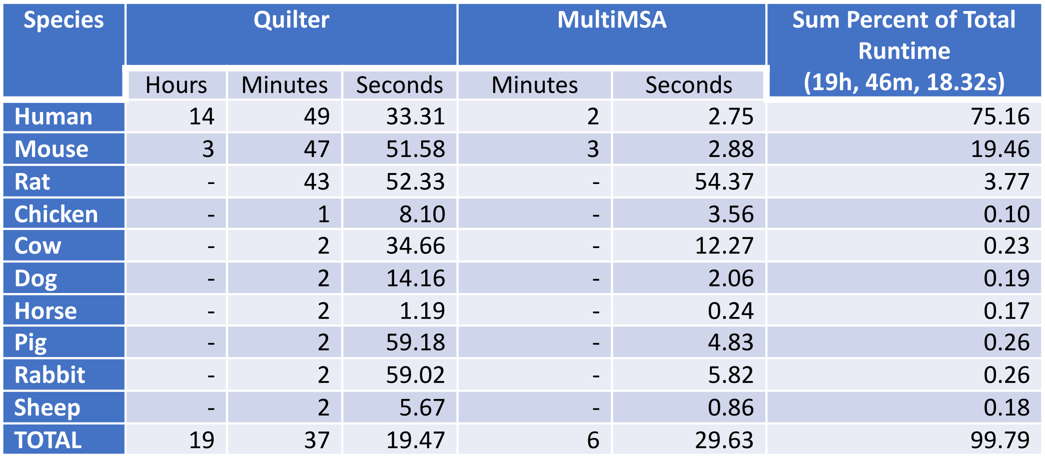
Figure 3.7: MSAs with exon bleeding detected by errors in their alignments of “pinch” amino acids.

3.3 Protein-to-Genome Mapping

Mirage requires just under 20 hours to produce multiple sequence alignments for the full protein database when using 32 processes; the vast majority of this time (over 99%) is spent computing the protein-to-genome mappings that form the basis for Mirage’s transitive intra-species alignments.

Most of Quilter’s runtime is related to running the mapping programs FastDiagonals and SPALN and parsing their output. While aligning the human sequences to their genome, Quilter ran FastDiagonals over 40 million times and SPALN nearly 200 thousand times. The average amount of wall time spent per system call to each program is roughly equal (around 0.01 seconds), although the amount of time spent processing SPALN’s outputs to identify a protein-genome mapping is about 10 times longer than FastDiagonals per program call. I refer to the amount of wall time elapsed between the start and completion of a system call to run a program as the “program time.” Program time is included in the “associated time,” which is the total amount of wall time elapsed between Quilter preparing the input files for the program and having completed its analysis of the program output.

The computational overhead associated with producing splice-aware protein-to-genome



Species	Quilter			MultiMSA		Sum Percent of Total Runtime (19h, 46m, 18.32s)
	Hours	Minutes	Seconds	Minutes	Seconds	
Human	14	49	33.31	2	2.75	75.16
Mouse	3	47	51.58	3	2.88	19.46
Rat	-	43	52.33	-	54.37	3.77
Chicken	-	1	8.10	-	3.56	0.10
Cow	-	2	34.66	-	12.27	0.23
Dog	-	2	14.16	-	2.06	0.19
Horse	-	2	1.19	-	0.24	0.17
Pig	-	2	59.18	-	4.83	0.26
Rabbit	-	2	59.02	-	5.82	0.26
Sheep	-	2	5.67	-	0.86	0.18
TOTAL	19	37	19.47	6	29.63	99.79

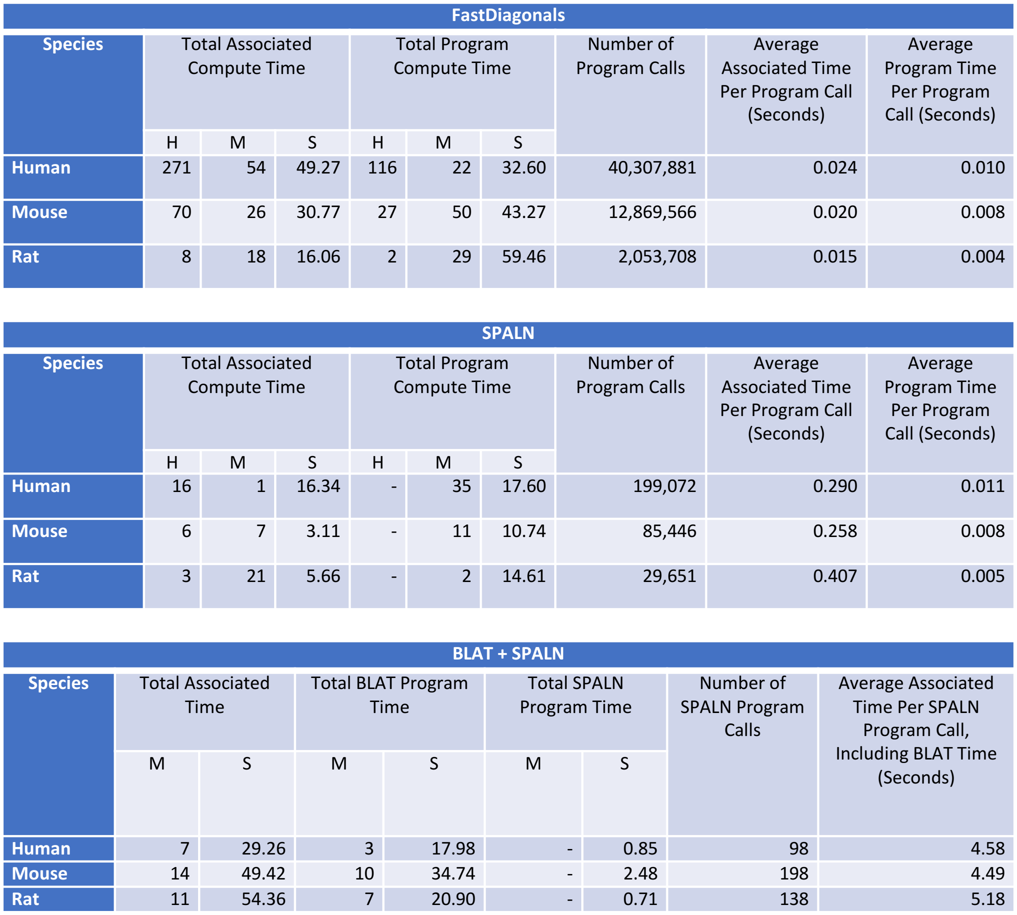
Figure 3.8: Breakdown of wall-clock time Mirage uses to generate intra-species transitive protein sequence alignments.

mappings causes Mirage to run noticeably slower than other MSA tools, as seen in Figure 3.10. To help address this weakness, Mirage has a “fast” option that prevents it from using FastDiagonals, so that all of its protein-to-genome mappings are generated by SPALN.

Mirage is highly successful at mapping the three species with gene indexing files, with 97.7% of human proteins, 93.3% of mouse proteins, and 90.9% of rat proteins mapping to their respective genomes, but struggles to identify high-quality protein-to-genome mappings for other species.

3.4 Addressing SPALN Errors

Given that excising FastDiagonals from its pipeline reduces Quilter’s runtime by nearly 2/3, it is worth considering whether Quilter should generally default to using SPALN and abandon FastDiagonals altogether. Further complicating Quilter’s primary reliance on FastDiagonals is the fact that GTF exons are often computationally predicted, such that FastDiagonals mappings cannot always claim to have greater scientific credibility than SPALN mappings. The primary issue with SPALN, and the reason why FastDiagonals will remain



FastDiagonals

Species	Total Associated Compute Time			Total Program Compute Time			Number of Program Calls	Average Associated Time Per Program Call (Seconds)	Average Program Time Per Program Call (Seconds)
	H	M	S	H	M	S			
Human	271	54	49.27	116	22	32.60	40,307,881	0.024	0.010
Mouse	70	26	30.77	27	50	43.27	12,869,566	0.020	0.008
Rat	8	18	16.06	2	29	59.46	2,053,708	0.015	0.004

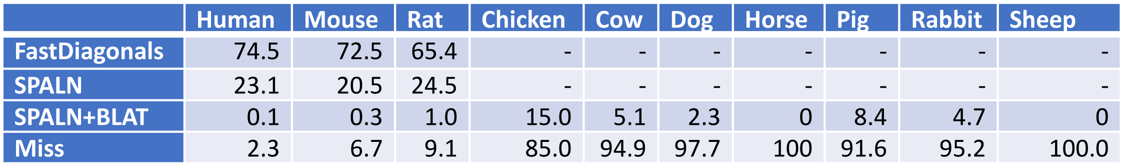
SPALN

Species	Total Associated Compute Time			Total Program Compute Time			Number of Program Calls	Average Associated Time Per Program Call (Seconds)	Average Program Time Per Program Call (Seconds)
	H	M	S	H	M	S			
Human	16	1	16.34	-	35	17.60	199,072	0.290	0.011
Mouse	6	7	3.11	-	11	10.74	85,446	0.258	0.008
Rat	3	21	5.66	-	2	14.61	29,651	0.407	0.005

BLAT + SPALN

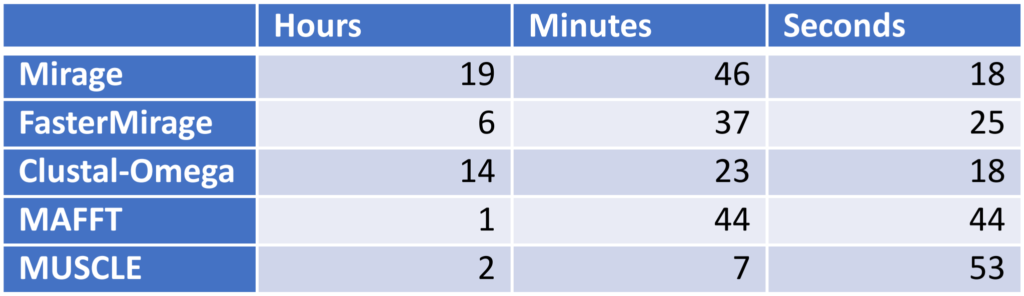
Species	Total Associated Time		Total BLAT Program Time		Total SPALN Program Time		Number of SPALN Program Calls	Average Associated Time Per SPALN Program Call, Including BLAT Time (Seconds)
	M	S	M	S	M	S		
Human	7	29.26	3	17.98	-	0.85	98	4.58
Mouse	14	49.42	10	34.74	-	2.48	198	4.49
Rat	11	54.36	7	20.90	-	0.71	138	5.18

Figure 3.9: Runtime comparisons of FastDiagonals, SPALN, and BLAT+SPALN for translated mapping in Quilter.



	Human	Mouse	Rat	Chicken	Cow	Dog	Horse	Pig	Rabbit	Sheep
FastDiagonals	74.5	72.5	65.4	-	-	-	-	-	-	-
SPALN	23.1	20.5	24.5	-	-	-	-	-	-	-
SPALN+BLAT	0.1	0.3	1.0	15.0	5.1	2.3	0	8.4	4.7	0
Miss	2.3	6.7	9.1	85.0	94.9	97.7	100	91.6	95.2	100.0

Figure 3.10: Percentage of sequences mapped to their genomes, by mapping method.



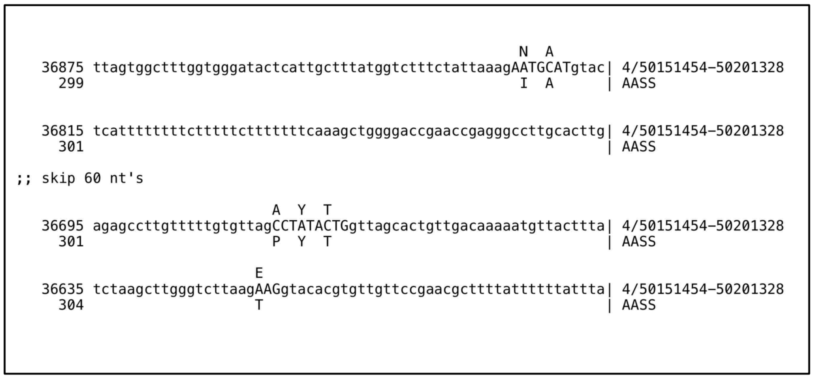
	Hours	Minutes	Seconds
Mirage	19	46	18
FasterMirage	6	37	25
Clustal-Omega	14	23	18
MAFFT	1	44	44
MUSCLE	2	7	53

Figure 3.11: Comparison of Mirage runtimes to other MSA tools, dividing work across 32 cores.

as Quilter's default mapping approach for the time being, is that there is a collection of characteristic errors associated with SPALN that erode our confidence in exclusively relying on its mappings. One may note in Figure 3.9 that the time required to parse SPALN output is 10 times greater than the amount of time that SPALN needs to run, due primarily to the amount of error-checking required to ensure that SPALN output is correct.

Perhaps the most pervasive issue presented by SPALN is the identification of "micro-exons," which are falsely asserted mappings of 1 to 3 amino acids to non-synonymous putative exons on the genome, frequently occurring in clusters, as in the SPALN output depicted in Figure 3.11. Micro-exons appear to be a product of SPALN over-penalizing the usage of non-canonical splice sites by genuine proteins, causing it to map amino acids from one end of an exon to any sufficiently-sized segment of genomic sequence that happens to be flanked by the canonical 'AG-GT' splice signal. To recover from micro-exons, Quilter has to check the length of each exon called by SPALN, and, wherever there is a cluster of micro-exons, identify the nearest sensible exons upstream and downstream from the micro-exon cluster and check whether or not they can be extended to encode the amino acids stranded on the micro-exons.

Another troubling bug in SPALN's output is that it occasionally misrepresents the per-



```
                                                           N  A
   36875 ttagtggctttggtgggatactcattgctttatggtctttctattaaagAATGCATgtac| 4/50151454-50201328
     299                                                   I  A      | AASS


   36815 tcattttttttctttttctttttttcaaagctggggaccgaaccgagggccttgcacttg| 4/50151454-50201328
     301                                                             | AASS

;; skip 60 nt's

                              A  Y  T
   36695 agagccttgtttttgtgttagCCTATACTGgttagcactgttgacaaaaatgttacttta| 4/50151454-50201328
     301                      P  Y  T                                | AASS

                            E
   36635 tctaagcttgggtcttaagAAGgtacacgtgttgttccgaacgcttttattttttattta| 4/50151454-50201328
     304                    T                                        | AASS
```

Figure 3.12: Example of a common SPALN error where amino acids are aligned without identity to a series of putative "micro-exons" that can be as short as short as a single codon.

cent identity of its mappings, with no observable pattern in either the frequency or degree of misrepresentation. Rather than immediately filtering out low-quality (¡97% identity) mappings based on SPALN's reported percent identity, Quilter has to fully read and process all SPALN output before computing the percent identity to determine whether or not the mapping met its quality threshold. SPALN typically computes high-quality mappings, so requiring Quilter to parse all SPALN output does not drastically impact Quilter's overall runtime, but it is nevertheless worrisome to see SPALN struggling to report its own alignment quality.

A final observation about SPALN output, and one which may turn out to be more of a feature than a bug, is that SPALN will occasionally the terminal exon(s) of a protein to map to the strand opposite the one that the other exons mapped to. While these alignments look bizarre, they are often high-quality and sufficiently long to challenge the assumption that they are spurious computational artifacts, thus warranting additional bioinformatic analysis as a future research topic.

As I note in my Future Directions, a near-term project is the development of a replacement for SPALN that will improve each of the errors that have been identified throughout the course of SPALN's employment within Quilter. The release of this SPALN replacement will likely coincide with an update of Mirage that replaces both FastDiagonals and SPALN with my improved translated alignment software, thus reducing the runtime of default Mirage while avoiding the computational overhead and epistemic quandries associated with SPALN. In the meantime, Mirage will rely on FastDiagonals as its default method for identifying high-quality protein-to-genome mappings.

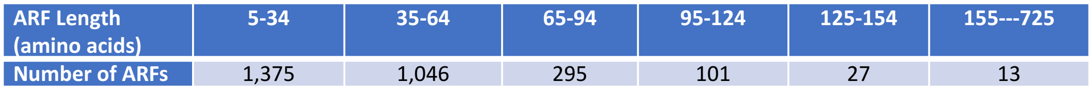
3.5 Alternative Reading Frames

Alternative reading frames (ARFs) are open reading frames that overlap on the genome but encode different protein peptides by using codons that are offset from one another by 1 or 2 nucleotides. We have observed that putatitive ARFs can span multiple exons, with the notable feature that proteins never appear to alternate between the use of standard and alternative exons reading frames in a multi-exonic ARF region. This may indicate that genuine ARFs are used to alter protein functionality on the scale of entire functional domains, since we would otherwise expect to observe occasional interpolations of standard and alternative reading frames.

It is unknown whether there is a general biological function that can be broadly attributed to ARFs, but our preliminary surveys of the peptides encoded by putative ARFs suggest that they may frequently be used to encode intrinsically disordered peptides. Encoding disordered sequence would allow ARFs to remove functionality associated with the standard reading frame while preserving necessary structural features for the other domains in the protein to function normally. This is analogous to more typical forms of alternative exon utilization, insofar as ARFs would make minor adjustments to protein functionality, but with the advantage of recycling genomic sequence for “spacer” peptides instead of encoding them separately from the standard, functional sequence. Whether ARFs can be widely associated with loss-of-function or if some ARFs provide alternative functional domains is an exciting open question whose answer may be established, in part, by Mirage.

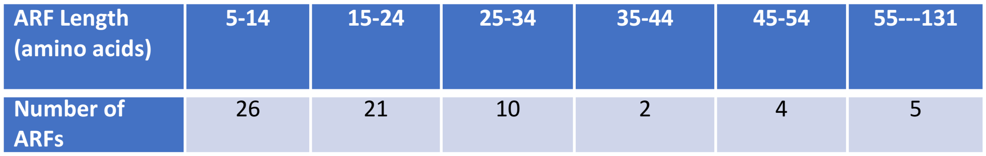
Mirage’s method of transitive alignment uniquely enables it to indicate putative ARFs during intra-species MSA generation. 2,926 putative ARFs have been identified in the

Uniprot dataset, with 2,267 gene families (10.3% of all families) exhibiting at least one ARF of 5 or more amino acids. The majority of putative ARFs use 2 overlapping reading frames, but a small number use all three reading frames, including a 3-frame ARF in the human MLL5 gene that encodes 131 overlapping amino acids and has is encoded by DNA that is conserved in the mouse genome (Note: much of the analysis of ARFs was performed in collaboration with Kaitlin Carey).



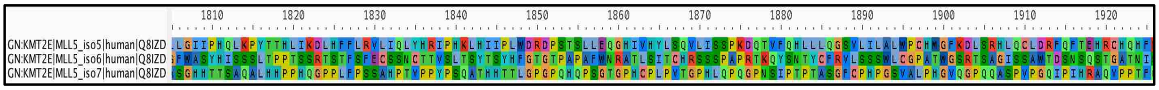
ARF Length (amino acids)	5-34	35-64	65-94	95-124	125-154	155---725
Number of ARFs	1,375	1,046	295	101	27	13

Figure 3.13: Length distribution of the 2,858 2-reading frame ARFs identified in our test database.



ARF Length (amino acids)	5-14	15-24	25-34	35-44	45-54	55---131
Number of ARFs	26	21	10	2	4	5

Figure 3.14: Length distribution of the 68 3-reading frame ARFs identified in our test database.



```
GN:KMT2E|MLL5_iso5|human|Q8IZD LGIIPHQLKPYTTHLIKDLHFFLRVLIQLYHRIPHKLHIIPLWDRDPSTSLLEQGHIVHYLSQVLISSPKDQTVFQHLLLQGSVLILALWPCHMGFKDLSRHLQCLDRFQFTEHRCHQHF
GN:KMT2E|MLL5_iso6|human|Q8IZD FWASYHISSSLTPPTSSRTSTFSFECSSNCTTVSLTSYTSYHFGTGTPAPAFWNRATLSITCHRSSSPAPRTKQYSNTYCFRVLSSSWLCGPATWGSRTSAGISSAWTDSNSQSTGATNI
GN:KMT2E|MLL5_iso7|human|Q8IZD SGHHTTSAQALHHPPHQGPPLFPSSAHPTYPPYPSQATHHTTLGPGPQHQPSGTGPHCPLPVTGPHLQPQGPNSIPTPTASGFCPHPGSVALPHGVQGPQQASPVPGQIPIHRAQVPPTF
```

Figure 3.15: Mirage transitive alignment of three human MLL5 sequences, displaying 120 of 131 amino acids in a putative 3-reading frame ARF.

Our lift-over analyses of human ARFs have also produced promising results on the veracity of putative ARFs identified by Mirage. We plotted (Figure 3.16, red dots) the correlation between the lengths of exons reported in the GTF file and the frequency with which exons of a given length had at least two overlapping open reading frames in the same strand direction. Unsurprisingly, there is a strong negative correlation, since the likelihood of a "stop" codon occurring on a string of random nucleotides increases with the string's length, and non-standard reading frames in non-ARF exons can be approximately characterized as random nucleotide strings. We then isolated the exons that Mirage had identified as encoding putative ARFs and used the lift-over analysis pipeline to extract the corresponding exons in the mouse genome and compute how frequently the ARF-associated mouse exons also encoded at least two open reading frames (Figure 3.16, blue dots). While a small number of putative human ARFs did not correspond to mouse exons with multiple viable open reading frames, almost every putative human ARF identified by Mirage was conserved in the mouse genome. This suggests that many of the ARFs we have identified may represent genuine cases of remarkable efficiency in biological coding.

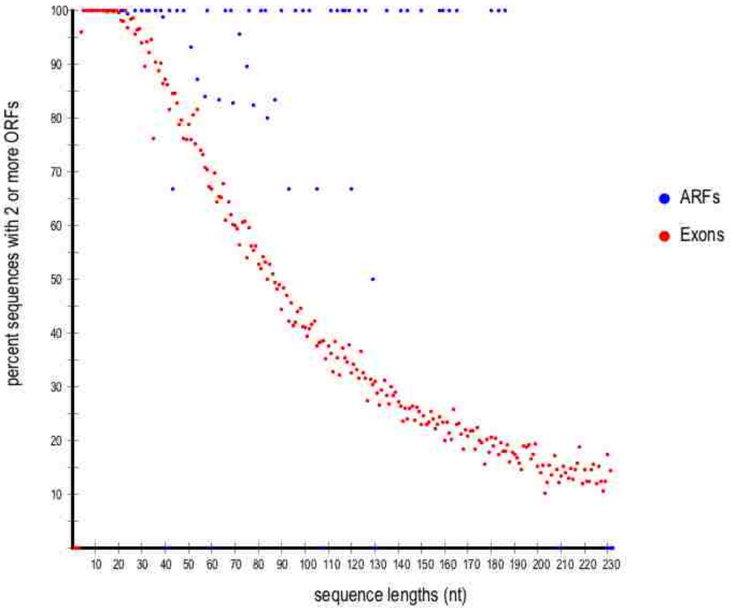


Figure 3.16: Graph showing differences in the frequency of multiple ORFs in all human exons compared to mouse exons identified as homologous to human exons with putative ARFs.

We further examined the frequency with which putative human ARFs correspond to open reading frames in lifted-over mouse DNA, allowing for variable-length windows of genomic sequence that only code in one reading frame to surround the region where multiple open reading frames overlap, as illustrated in Figure 3.17. We found that extending the permissible distance between the splice sites associated with the overlapping open reading frames can bring the percentage of putative human ARFs that correspond to ARF-viable mouse DNA up to 97.8% (when allowing offsets up to 45 bases from either end, excluding the one or two nucleotides necessary for the shift in reading frame) from the already exciting value of 89.7% (only allowing an offset of 3 nucleotides on either end).

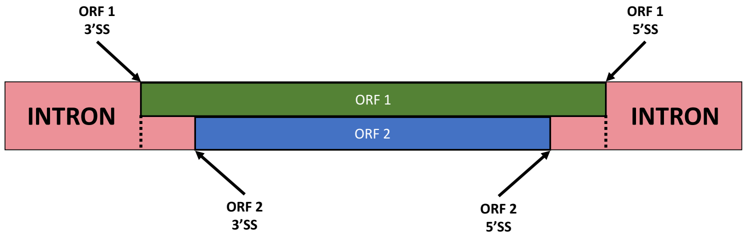
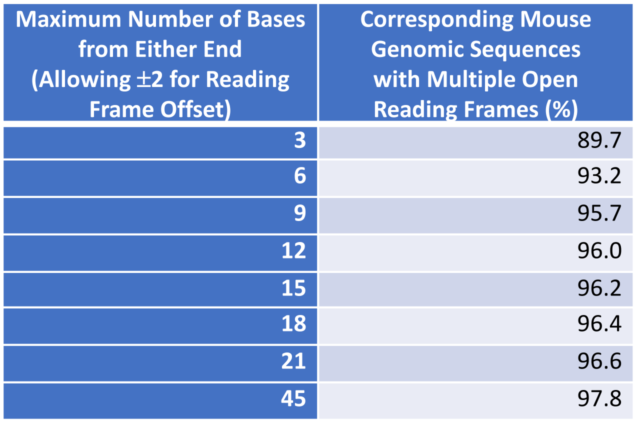


Figure 3.17: Splice sites for alternative reading frames may be offset from one another by several nucleotides.



Maximum Number of Bases from Either End (Allowing ±2 for Reading Frame Offset)	Corresponding Mouse Genomic Sequences with Multiple Open Reading Frames (%)
3	89.7
6	93.2
9	95.7
12	96.0
15	96.2
18	96.4
21	96.6
45	97.8

Figure 3.18: Percentages of putative human ARFs with lifted-over mouse genomic sequence that has multiple open reading frames, varying the permissible amount of flanking non-coding sequence.

CHAPTER 4 FUTURE DIRECTIONS

4.1 An Improved Translated Sequence Alignment Tool

Mirage's protein-to-genome mapping phase relies on the translated sequence alignment tool SPALN to identify high-quality spliced protein-genome alignments for proteins whose GTF entries failed to produce a full-protein mapping. SPALN is the strongest member of a small class of splice-aware translated alignment tools, none of which appear to be under active development [3, 20]. While presently the best in its category, SPALN also exhibits a number of characteristic flaws that occasionally filter into its output alignments and require substantial recovery work on the part of Mirage. Clusters of "micro-exons" comprised of fewer than 4 amino acids, inconsistent nucleotide indexing, misreported percents identity, and spontaneous changes in DNA strand direction are all features of SPALN output that Mirage has been programmed to detect and (where possible) correct. Mirage's extensive wrapper script guarantees that the protein-to-genome mappings it derives from SPALN are high-quality, but replacing SPALN with an improved tool for splice-aware translated alignment will be a necessary advancement for Mirage and for future bioinformatics applications. My most immediate research aim is to develop a splice-aware translated alignment software tool that will integrate probabilistic models of species-specific splice-site patterns with biochemical analyses of splice-site recognition proteins in order to achieve fast and accurate exon prediction, combined with the use of cutting-edge datastructures to optimize my alignment tool's speed and memory usage.

An additional component of this improved alignment tool will be an efficient method for aligning RNA and cDNA to the genome. This is a closely-related problem to protein-to-genome mapping, as cDNA and RNA (specifically, spliced mRNA) alignment also require attentiveness to splice signalling and the exonic structure of genomic DNA. Many labs that research proteins focus on RNA, primarily producing and analyzing RNAseq data instead of directly acquiring amino acid sequences from mass spectrometry, so a splice-aware sequence-to-genome alignment tool benefits from the ability to work with all three biological alphabets. My software will still work with these sequences on the level of translated proteins, as this produces a computationally simpler problem, but using additional post-processing to confirm that the nucleotide-to-genome alignments are as sensible as the translated protein-to-genome alignment. This generality will give my software wide appeal to protein and RNA research groups in addition to improving the speed and quality of Mirage by replacing FastDiagonals and SPALN.

4.2 Splice-Aware Translated Sequence Homology Search in HMMER

The HMMER software suite is a toolkit designed for database homology search through the use of probabilistic models called hidden Markov models (HMMs), and is one of the most widely used bioinformatics software packages [3]. For a given set of query sequences, HMMER is able to rapidly and accurately identify evolutionarily-related sequences in a large database and precisely quantify the statistical significance of each sequence pair's similarity. A recent addition to the HMMER suite is the translated search tool "thmmer" developed by Walt Shands under the guidance Dr. Travis Wheeler at the University of Montana, the lead developer of HMMER's DNA search tool "nhmmer" [24]. The addition of splice-awareness to thmmer translated homology search will greatly improve the program's functionality and have substantial scientific impact.

Following the development of my standalone translated alignment tool, I will be well-positioned to develop novel probabilistic graphical models for searching collections of protein HMMs against DNA sequence databases that will effectively account for intron splicing while evaluating sequence homology. The successful software implementation of these models will involve researching algorithms for the efficient application of splice-aware translated database search to large-scale bioinformatics datasets. The culmination of this project will be integrating splice-aware translated homology search into a full release on the HMMER webserver through collaboration with Rob Finn at the European Bioinformatics Institute.

4.3 ARF Research

The unexpected abundance of alternative reading frames (ARFs) revealed by Mirage has provided the Wheeler lab group with an exciting opportunity for original bioinformatics research. Over the past few months I have begun working with Kaitlin Carey, an advanced undergraduate with a background in molecular biology, towards verifying purported ARFs indicated by Mirage and examining the extent to which DNA encoding multiple reading frames is conserved across highly diverged species. We are extremely excited by the preliminary results, including the identification of a stretch of DNA that has 3 forward reading frames capable of encoding 131 amino acids and that is perfectly conserved between humans and mice. Ongoing research into the actual frequency with which alternative reading frames are translated and the effects that they have on protein behaviors will be an informative application of the Mirage pipeline towards a cutting-edge biological curiosity.

BIBLIOGRAPHY

[1] R. C. Edgar, “MUSCLE: multiple sequence alignment with high accuracy and high throughput,” *Nucleic Acids Research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[2] R. C. Edgar and S. Batzoglou, “Multiple sequence alignment,” *Current Opinion in Structural Biology*, vol. 16, no. 3, pp. 368–373, 2006.

[3] R. D. Finn, J. Clements, and S. R. Eddy, “HMMER web server: interactive sequence similarity searching,” *Nucleic Acids Research*, vol. 39, no. suppl_2, pp. W29–W37, 2011.

[4] O. Gotoh, M. Morita, and D. R. Nelson, “Assessment and refinement of eukaryotic gene structure prediction with gene-structure-aware multiple protein sequence alignment,” *BMC Bioinformatics*, vol. 15, no. 1, p. 189, 2014.

[5] H. Iwata and O. Gotoh, “Benchmarking spliced alignment programs including Spaln2, an extended version of Spaln that incorporates additional species-specific features,” *Nucleic Acids Research*, vol. 40, no. 20, pp. e161–e161, 2012.

[6] W. J. Kent, “BLATthe BLAST-like alignment tool,” *Genome Research*, vol. 12, no. 4, pp. 656–664, 2002.

[7] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler, “The human genome browser at UCSC,” *Genome Research*, vol. 12, no. 6, pp. 996–1006, 2002.

[8] E. Kovacs, P. Tompa, K. Liliom, and L. Kalmar, "Dual coding in alternative reading frames correlates with intrinsic protein disorder," *Proceedings of the National Academy of Sciences*, vol. 107, no. 12, pp. 5429–5434, 2010.

[9] D. G. Higgins and P. M. Sharp, "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer," *Gene*, vol. 73, no. 1, pp. 237–244, 1988.

[10] P. V. Hornbeck, B. Zhang, B. Murray, J. M. Kornhauser, V. Latham, and E. Skrzypek, "PhosphoSitePlus, 2014: mutations, PTMs and recalibrations," *Nucleic Acids Research*, vol. 43, no. D1, pp. D512–D520, 2014.

[11] S. Maas, A. Rich, and K. Nishikura, "A-to-I RNA editing: recent news and residual mysteries," *Journal of Biological Chemistry*, vol. 278, no. 3, pp. 1391–1394, 2003.

[12] V. Marx, "Biology: The big challenges of big data," *Nature*, vol. 498, no. 7453, pp. 255–260, 2013.

[13] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[14] Q. Pan, O. Shai, L. J. Lee, B. J. Frey, and B. J. Blencowe, "Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing," *Nature Genetics*, vol. 40, no. 12, pp. 1413–1415, 2008.

[15] S. Pundir, M. J. Martin, and C. O'Donovan, "UniProt tools," *Current Protocols in Bioinformatics*, pp. 1–29, 2016.

[16] S. Pundir, M. J. Martin, and C. ODonovan, "UniProt protein knowledgebase," *Protein Bioinformatics: From Protein Modifications and Networks to Proteomics*, pp. 41–55, 2017.

[17] C. Sander and R. Schneider, "Database of homology-derived protein structures and the structural meaning of sequence alignment," *Proteins: Structure, Function, and Bioinformatics*, vol. 9, no. 1, pp. 56–68, 1991.

[18] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding *et al.*, "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega," *Molecular Systems Biology*, vol. 7, no. 1, p. 539, 2011.

[19] G. S. C. Slater and E. Birney, "Automated generation of heuristics for biological sequence comparison," *BMC bioinformatics*, vol. 6, no. 1, p. 31, 2005.

[20] D. Staiger and J. W. Brown, "Alternative splicing at the intersection of biological timing, development, and stress responses," *The Plant Cell*, vol. 25, no. 10, pp. 3640–3656, 2013.

[21] K. Tamura, J. Dudley, M. Nei, and S. Kumar, "MEGA4: molecular evolutionary genetics analysis (MEGA) software version 4.0," *Molecular Biology and Evolution*, vol. 24, no. 8, pp. 1596–1599, 2007.

[22] T. J. Wheeler and S. R. Eddy, "nhmmer: DNA homology search with profile HMMs," *Bioinformatics*, vol. 29, no. 19, pp. 2487–2489, 2013.