9.1 BLAST

9.1.1 Sequence Alignment

Sequence Alignment

Often in genetics it is important to determine the similarity of two sequences, and conversely, to identify differences between a pair of sequences. During the COVID-19 pandemic, SARS-CoV-2 samples were sequenced and compared to earlier virus variants to identify mutations, which aided in the prediction of variant severity.

Similarities and differences between a pair of sequences can be determined by **sequence alignment**. Sequence alignment is the process of identifying regions of similarity between nucleotide or amino acid sequences. There are many algorithms to perform these alignments, including the Smith-Waterman algorithm which you can read about here: https://cs.stanford.edu/people/eroberts/courses/soco/projects/computers-and-the-hgp/smith_waterman.html. (Understanding the algorithm is not required for this course.)

Here are two short 14 base DNA sequences:

```
ACCGTATTGCTAGG          ACTATCGCTACGGT
```

The alignment of these two sequences would look like this:

```
ACCGTATTGCTA-GG-
||   |||  ||||  ||
AC--TATCGCTACGGT
```

Each vertical bar (|) represents a match between the two sequences, and missing vertical bars indicate a mismatch or gap. A mismatch occurs when the bases in the two sequences are not the same at a particular position in the alignment (e.g., one sequence has a T and the other a G). Gaps, or bases that exist in only of the two sequences, are represented by hyphens (-) in the sequence in which they are missing (e.g., one sequence contains AGC and the other only AG). The similarity between two sequences can be quantified using an alignment score. In an alignment score, each match increases the score, and each mismatch or gap decreases the score.

Sequence alignments can also be performed using amino acid sequences, although these alignments are generally scored differently. Here are two short protein sequences, 26 and 30 amino acids long, respectively:

```
PPEDILPSPHCMDDLLLPQDVEEFFE    PENNVLSPLPSQAMDDLMLSPDDIEQWFTE
```

The alignment of these two sequences would look like this:

```
PPEDIL---PSPHCMDDLLL-PQDVEEFF-E
| +++|    ||   +||||+|  |+|+|++|  |
PENNVLSPLPSQ-AMDDLMLSPDDIEQWFTE
```

Where the residues in each sequence match exactly, there are vertical bars, and hyphens represent gaps, as in the nucleotide sequence alignment. However, there are also plus signs between some mismatched residues. These mismatches are called **positives**.

If an amino acid is replaced by an amino acid with similar properties (e.g., charge, hydrophobicity), it may cause little to no difference in the structure and function of the protein. Therefore, in amino acid sequence alignments, **substitution matrices** are often used to score mismatches based on the ability of one amino acid to substitute for another. The most commonly used substitution matrices contain probabilities for each possible amino acid substitution based on analysis of evolutionarily conserved sequences.

Look back at the alignment of the two amino acid sequences above. Most of the mismatches are positives, meaning that the substitution matrix has identified them as amino acids that are often replaced with one another. These amino acids will have conserved physico-chemical properties, for example, where valine and isoleucine are aligned there is a plus sign. Both residues are aliphatic and can thus be substituted for one another without drastically altering protein structure and function in many cases. When scoring amino acid alignments, positive substitutions can be included so that mismatches that have little affect the protein do not decrease the alignment score.

The similarity between two sequences can be quantified by **percent identity** (also % identity). The percent identity is the percentage of exact matches in an alignment.

$$\%\ identity = 100\ \times \frac{\#\ of\ matches}{length\ of\ alignment}$$

To determine the percent identity of an alignment, we must determine the number of matches and the length of the alignment. DNA sequence alignment from above:

```
ACCGTATTGCTA-GG-
||   ||| |||| ||
AC--TATCGCTACGGT
```

There are 11 matches in this alignment, and the total length is 16. Despite each sequence being 14 bases long, the length of the entire alignment is 16 due to the gaps in the alignment. Thus, the percent identity of the two sequences is 68.8%

$$\%\ identity = 100\ \times \frac{11}{16} = 68.8$$

For amino acid sequence alignments, percent identity can be calculated, as well as the **percent positive substitutions**.

$$\%\ positive\ substitutions = 100\ \times \frac{\#\ of\ matches + \#\ of\ positives}{length\ of\ alignment}$$

Amino acid sequence alignment from above (matches=14, positives=9, length=31):

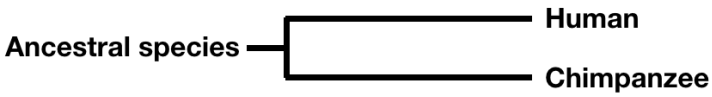
```
PPEDIL---PSPHCMDDLLL-PQDVEEFF-E
| +++|    ||   +||||+| |+|+|++| |
PENNVLSPLPSQ-AMDDLMLSPDDIEQWFTE
```

While the percent identity is only 45.2% (calculation not shown), the % positive substitutions is 74.2%:

$$\%\ identity = 100 \times \frac{14 + 9}{31} = 74.2$$

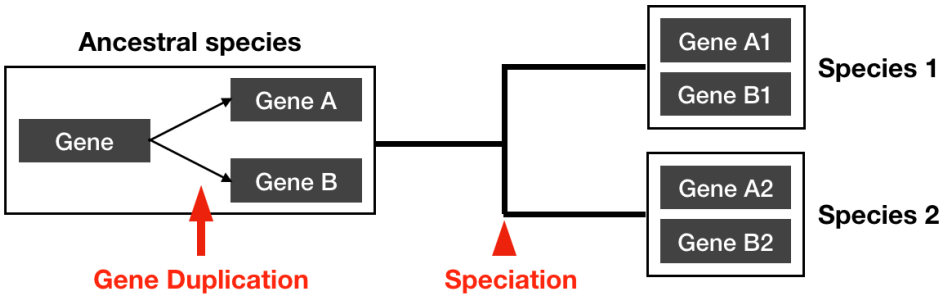
Homologs

A **speciation event** occurs when an ancestral species evolves into two or more distinct species. For example, before the human and chimpanzee lineage diverged, there was an ancestral species that existed prior to these species. Genetic changes caused the lineage to split into two distinct species.



One of the main mechanisms of evolution of new **genes** is **duplication and divergence**. When a gene is duplicated in a species, it gives rise to two genes. Because the new copy will create some redundancy, the new gene has the potential to pick up mutations over time. These new mutations can lead to **neofunctionalization** or **subfunctionalization**. Neofunctionalization occurs when one of the gene copies gains mutations give the gene a new function. Subfunctionalization occurs when both genes gain mutations that lead them to share the role of the ancestral gene.

The figure below visualizes the duplication of a gene in an ancestral species followed by a speciation event. Both species that arise from the ancestral species each have a copy of gene A and gene B.



Gene A1, gene B1, gene A2, and gene B2 are all **homologs**. Homologs are genes derived from the same common ancestral gene. Because these four genes were all derived from the original ancestral gene, they are all homologs, or homologous sequences. The identification of homologs is important aspect of biological research. Homologs aid in our understanding of how evolution occurred, can be used to predict the gene functions, and allow the study of human diseases in animal models.

Two important subtypes of homologs are **paralogs** and **orthologs**. Both paralogs and orthologs descend from an ancestral gene (i.e., they are homologs), but in slightly different ways. Orthologs are ONLY separated by speciation events. In other words, they were the same gene in the most recent common ancestral species. In the figure above, gene A1 and A2 are orthologs, because they descend directly from gene A. Alternatively, paralogs are separated by gene duplication. In the figure above, gene A1 is paralogous to gene B1, because, although they are both descended from "gene", they were separated by gene duplication in the ancestral species.

More simply, a paralog is a homolog that is in the same species and an ortholog is a homolog that is in a different species. Gene A1 and B1 are paralogs, as are gene A2 and gene B2. Gene A1 and gene A2 are orthologs, as are gene B1 and gene B2.

A key application of sequence alignment is the identification of homologs. Sequence alignment also allows quantification of sequence conservation and the building phylogenetic trees. Processing data from sequencing experiments is another application of sequence alignment; sequencing reads must be aligned to the genome or transcriptome to identify the sequence.

9.1.2 Installing Packages in Terminal

BLAST

BLAST (Basic Local Alignment Search Tool) is one of the most commonly used programs for sequence alignment. BLAST searches for sequence alignments between a set of **query sequences** and a set of **subject sequences** (or a **sequence database**) and scores the alignments. The query sequence is the sequence being searched for, and the subject sequences, also called the database is the set of sequences to search within. This is analogous to using the "Find" function to search a document for a specific word. The word would be the query and the document would be the subject.

BLAST is composed of multiple programs for different situations: `blastn` performs nucleotide sequence alignments, `blastp` performs protein sequence alignments, and `blastx` and `tblastn` search for alignments between the two different types of sequences. `blastx` translates a nucleotide query sequence into an amino acid sequence and then searches for matches to a set of amino acid subject sequences. `tblastn` takes an

amino acid sequence, determines the possible nucleotide sequences that would translate to that amino acid sequence, and uses them to search within a set of nucleotide sequences.

Installing Packages

The BLAST package of programs can be installed and used in the command line. A **package** is a collection of computer programs that can be installed and run on a computer. The programmers who build packages often use programs from existing packages as part of their code. This means that the package they build relies on the presence of other packages to function, these other packages are called **package dependencies**.

Conda is a package and environment management system, it installs, runs, and updates packages. Conda can install packages from specific **channels** (repositories of packages). One of these channels is **Bioconda**, which contains thousands of bioinformatics packages including BLAST.

The command to install a package using Conda is `conda install`:

```
conda install -c CHANNEL PACKAGENAME
```

The -c (channel) option tells Conda that to install a package from a specific channel. To install BLAST, which is hosted on Bioconda, issue the following command:

```
j:~$ conda install -c bioconda blast
```

Conda may find that the package being installed has dependencies. If so, it will ask if it should proceed:

```
Proceed ([y]/n)?
```

Respond by typing the letter `y` and pressing enter.

When working on JupyterHub, BLAST will need to be installed every time a new terminal instance is launched. When using Terminal on another computer, or when using another command line interface, a package will only need to be installed once, however, Conda may need to be installed first. Instructions for installing Conda (which you will not need in this course) can be found here: https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html

Once a package is installed via Conda, the commands contained in the package can be run from any directory in the directory structure, like bash commands (`cd`, `pwd`). The BLAST package includes multiple command line tools, including the four that were discussed previously (`blastn`, `blastp`, `blastx`, and `tblastn`), and others, such as `makeblastdb`. To get help with a package command, `-help` can be typed after the name of the command:

```
j:~$ blastn -help
```

These pages can be quite long and confusing to parse. Most packages have an online manual which tends to be easier to read and understand. The BLAST manual is located here: https://www.ncbi.nlm.nih.gov/books/NBK279690/

9.1.3 Basics of BLAST

FASTA Format

The **FASTA format** is a text-based format that is used for storing sequences. It can store either nucleotide or amino acid sequences. Below is the content of a file containing two amino acid sequences in FASTA format. These are the amino acid sequences of the human TP53 and mouse Tp53 proteins (sourced from UniProt https://www.uniprot.org/):

```
>sp|P04637|P53_HUMAN Cellular tumor antigen p53 OS=Homo sapiens GN=TP53
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGP
DEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAK
SVTCTYSPALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHE
RCSDSDGLAPPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNS
SCMGGMNRRPILTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELP
PGSTKRALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPG
GSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD
>sp|P02340|P53_MOUSE Cellular tumor antigen p53 OS=Mus musculus GN=Tp53
MTAMEESQSDISLELPLSQETFSGLWKLLPPEDILPSPHCMDDLLLPQDVEEFFEGPSEA
LRVSGAPAAQDPVTETPGPVAPAPATPWPLSSFVPSQKTYQGNYGFHLGFLQSGTAKSVM
CTYSPPLNKLFCQLAKTCPVQLWVSATPPAGSRVRAMAIYKKSQHMTEVVRRCPHHERCS
DGDGLAPPQHLIRVEGNLYPEYLEDRQTFRHSVVVPYEPPEAGSEYTTIHYKYMCNSSCM
GGMNRRPILTIITLEDSSGNLLGRDSFEVRVCACPGRDRRTEEENFRKKEVLCPELPPGS
AKRALPTCTSASPPQKKKPLDGEYFTLKIRGRKRFEMFRELNEALELKDAHATEESGDSR
AHSSYLKTKKGQSTSRHKKTMVKKVGPDSD
```

Each sequence in a FASTA file is preceded by a line containing a greater than symbol and a description of the sequence. Lines in a FASTA file containing sequence descriptions should be unique. In the example file, this first sequence contains a protein identifier, a description of the protein, the species, and the name of the gene that encodes the protein. All lines following the single-line sequence description contain the sequence of the protein described above it.

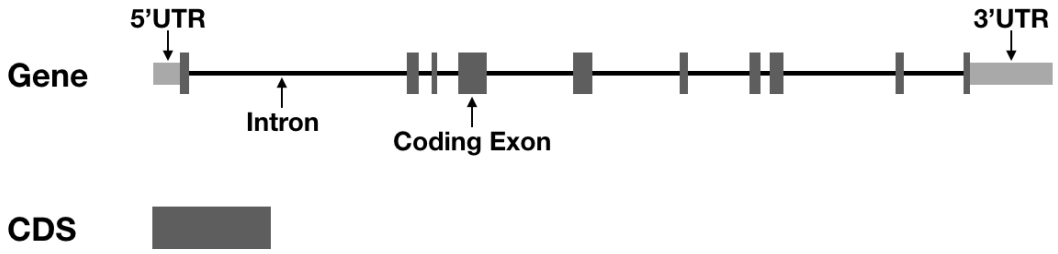
In the example, line 1 contains a description for the human TP53 protein, lines 2-8 contain the full amino acid sequence for the human TP53 protein, line 9 contains a description for the mouse Tp53 protein, and lines 10-16 contain the full amino acid sequence for the mouse Tp53 protein.

A FASTA file can contain one sequence or multiple sequences, and it contain nucleotide or amino acid sequences, however it should never contain both types of sequence in the same file. FASTA files are used to store:

- Specific sequences of interest (a set of protein or gene sequences being studied)
- Whole genomes (each chromosome is stored as one sequence)
- Whole proteomes (each protein sequence is stored as one sequence)
- Whole transcriptomes (each transcript sequence is stored as one sequence)
- Sets of coding sequences (CDSs) for a species (each CDS is stored as one sequence)

FASTA files usually have the extensions ".fasta" or ".fa", however, there are other extensions that are less commonly used like ".fna" and ".faa".

A **coding sequence (CDS)** is the nucleotide sequence that determines the amino acid sequence of a gene's protein product. It only contains the nucleotides that are translated into protein, from the start codon to the stop codon. This differs from a gene sequence which contains the 5'UTR sequence, intronic sequences, and the 3'UTR. Therefore, a CDS sequence is much shorter and only contains the part of the gene sequence meaning it does not contain introns or untranslated regions.



Nucleotide BLAST

When using the BLAST package of programs for sequence alignment, the query and subject sequences are provided to the blast program as FASTA files.

BLAST can be used to determine the similarity of the coding sequences of the *Homo sapiens* (human) TP53 gene and the *Mus musculus* (mouse) Tp53 gene. The *Homo sapiens* TP53 CDS sequence will be used as the query, which is contained in the file `Hs_TP53_CDS.fa`:

```
j:~/Week9/9.1.BLAST$ cat Hs_TP53_CDS.fa
> TP53 Homo sapiens CDS
ATGGAGGAGCCGCAGTCAGATCCTAGCGTCGAGCCCCCTCTGAGTCAGGAAACATTTTCAGACCTATGGA
AACTACTTCCTGAAAACAACGTTCTGTCCCCCTTGCCGTCCCAAGCAATGGATGATTTGATGCTGTCCCC
GGACGATATTGAACAATGGTTCACTGAAGACCCAGGTCCAGATGAAGCTCCCAGAATGCCAGAGGCTGCT
CCCCCCGTGGCCCCTGCACCAGCAGCTCCTACACCGGCGGCCCCTGCACCAGCCCCCTCCTGGCCCCTGT
CATCTTCTGTCCCTTCCCAGAAAACCTACCAGGGCAGCTACGGTTTCCGTCTGGGCTTCTTGCATTCTGG
GACAGCCAAGTCTGTGACTTGCACGTACTCCCCTGCCCTCAACAAGATGTTTTGCCAACTGGCCAAGACC
TGCCCTGTGCAGCTGTGGGTTGATTCCACACCCCCGCCCGGCACCCGCGTCCGCGCCATGGCCATCTACA
AGCAGTCACAGCACATGACGGAGGTTGTGAGGCGCTGCCCCCACCATGAGCGCTGCTCAGATAGCGATGG
```

```
TCTGGCCCCTCCTCAGCATCTTATCCGAGTGGAAGGAAATTTGCGTGTGGAGTATTTGGATGACAGAAAC
ACTTTTCGACATAGTGTGGTGGTGCCCTATGAGCCGCCTGAGGTTGGCTCTGACTGTACCACCATCCACT
ACAACTACATGTGTAACAGTTCCTGCATGGGCGGCATGAACCGGAGGCCCATCCTCACCATCATCACACT
GGAAGACTCCAGTGGTAATCTACTGGGACGGAACAGCTTTGAGGTGCGTGTTTGTGCCTGTCCTGGGAGA
GACCGGCGCACAGAGGAAGAGAATCTCCGCAAGAAAGGGGAGCCTCACCACGAGCTGCCCCCAGGGAGCA
CTAAGCGAGCACTGCCCAACAACACCAGCTCCTCTCCCCAGCCAAAGAAGAAACCACTGGATGGAGAATA
TTTCACCCTTCAGATCCGTGGGCGTGAGCGCTTCGAGATGTTCCGAGAGCTGAATGAGGCCTTGGAACTC
AAGGATGCCCAGGCTGGGAAGGAGCCAGGGGGGAGCAGGGCTCACTCCAGCCACCTGAAGTCCAAAAAGG
GTCAGTCTACCTCCCGCCATAAAAAACTCATGTTCAAGACAGAAGGGCCTGACTCAGACTGA
```

The *Mus musculus* CDS will be used as the subject sequence, which is contained in the file `Mm_TP53_CDS.fa`:

```
j:~/Week.9/9.1.BLAST$ cat Mm_Tp53_CDS.fa
> Tp53 Mus musculus CDS
ATGACTGCCATGGAGGAGTCACAGTCGGATATCAGCCTCGAGCTCCCTCTGAGCCAGGAGACATTTTCAG
GCTTATGGAAACTACTTCCTCCAGAAGATATCCTGCCATCACCTCACTGCATGGACGATCTGTTGCTGCC
CCAGGATGTTGAGGAGTTTTTTGAAGGCCCAAGTGAAGCCCTCCGAGTGTCAGGAGCTCCTGCAGCACAG
GACCCTGTCACCGAGACCCCTGGGCCAGTGGCCCCTGCCCCAGCCACTCCATGGCCCCTGTCATCTTTTG
TCCCTTCTCAAAAAACTTACCAGGGCAACTATGGCTTCCACCTGGGCTTCCTGCAGTCTGGGACAGCCAA
GTCTGTTATGTGCACGTACTCTCCTCCCCTCAATAAGCTATTCTGCCAGCTGGCGAAGACGTGCCCTGTG
CAGTTGTGGGTCAGCGCCACACCTCCAGCTGGGAGCCGTGTCCGCGCCATGGCCATCTACAAGAAGTCAC
AGCACATGACGGAGGTCGTGAGACGCTGCCCCCACCATGAGCGCTGCTCCGATGGTGATGGCCTGGCTCC
TCCCCAGCATCTTATCCGGGTGGAAGGAAATTTGTATCCCGAGTATCTGGAAGACAGGCAGACTTTTCGC
CACAGCGTGGTGGTACCTTATGAGCCACCCGAGGCCGGCTCTGAGTATACCACCATCCACTACAAGTACA
TGTGTAATAGCTCCTGCATGGGGGGCATGAACCGCCGACCTATCCTTACCATCATCACACTGGAAGACTC
CAGTGGGAACCTTCTGGGACGGGACAGCTTTGAGGTTCGTGTTTGTGCCTGCCCTGGGAGAGACCGCCGT
ACAGAAGAAGAAAATTTCCGCAAAAAGGAAGTCCTTTGCCCTGAACTGCCCCCAGGGAGCGCAAAGAGAG
CGCTGCCCACCTGCACAAGCGCCTCTCCCCCGCAAAAGAAAAAACCACTTGATGGAGAGTATTTCACCCT
CAAGATCCGCGGGCGTAAACGCTTCGAGATGTTCCGGGAGCTGAATGAGGCCTTAGAGTTAAAGGATGCC
CATGCTACAGAGGAGTCTGGAGACAGCAGGGCTCACTCCAGCTACCTGAAGACCAAGAAGGGCCAGTCTA
CTTCCCGCCATAAAAAAACAATGGTCAAGAAAGTGGGGCCTGACTCAGACTGA
```

Because a CDS is a nucleotide sequence, the `blastn` command should be used, which takes nucleotide sequences for both the query and subject. The syntax for running the `blastn` command is:

```
blastn -query QUERY.fa -subject SUBJECT.fa
```

The query sequence is provided in the file `QUERY.fa` and the subject sequence in the file `SUBJECT.fa`, after the options `-query` and `-subject`, respectively. To align the Homo sapiens TP53 CDS and the Mus musculus Tp53 CDS, the following command is run.

```
j:~/Week.9/9.1.BLAST$ blastn -query Hs_TP53_CDS.fa -subject
Mm_Tp53_CDS.fa
```

The output of this command is shown in smaller portions to go over each part individually.

The first lines of the output show the version of BLAST along with references to the BLAST paper. This is directly followed by information describing the database, which in this case is the subject sequence. The file name, the number of sequences it contains, and the length of the sequence are shown. Next, the name and length of the query sequence are displayed, followed by a description of the alignments of the query sequence to the subject sequence(s). In this case there is only one alignment. It provides a score and an E-value for the alignment.

```
BLASTN 2.12.0+


Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.



Database: User specified sequence set (Input: Mm_Tp53_CDS.fa).
           1 sequences; 1,173 total letters



Query= TP53 Homo sapiens CDS

Length=1182
                                                                  Score     E
Sequences producing significant alignments:                      (Bits)  Value

Tp53 Mus musculus CDS                                             915     0.0
```

The next portion of the output shows alignments, in this case there is only one. Each alignment starts with an overview of the alignment:

```
> Tp53 Mus musculus CDS
Length=1173

 Score = 915 bits (495),  Expect = 0.0
 Identities = 961/1187 (81%), Gaps = 28/1187 (2%)
 Strand=Plus/Plus
```

It begins with the name and length of the subject sequence. Again, we see the alignment score, which in this case is 915 bits. This course will not go over the details of how this score is generated, but information can be found here: https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html

The **Expect-value (E-value)** quantifies "the number of hits one can expect to see by chance when searching a database of a particular size." (https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=FAQ) Therefore, an E-value of 10 would indicate that one could expect to find 10 hits to

the query sequence of the same quality purely by chance. The closer the value is to 0, the more significant the match. An expect value of 0 is very significant.

This section also provides the number of matches, the length of the alignment and the percent identity: `Identities = matches/alignment length (% identity)`
As well as the number of gaps, and the proportion of the alignment that contains gaps:
`Gaps = # bases in gaps/alignment length (% gaps)`
The strand information indicates if one of the sequences was reverse complemented to align to the other, `Plus/Plus` indicates that this was not done.

Following this is the alignment of the query sequence and the subject sequence described above:

```
Query  1     ATGGAGGAGCCGCAGTCAGATCCTAGCGTCGAGCCCCCTCTGAGTCAGGAAACATTTTCA  60
             |||||||||  |  |||||  |||   |||  |||||||  |||||||||  |||||  |||||||||
Sbjct  10    ATGGAGGAGTCACAGTCGGATATCAGCCTCGAGCTCCCTCTGAGCCAGGAGACATTTTCA  69

Query  61    GACCTATGGAAACTACTTCCTGAAAACAACGTTCTGTCCCCCTTGCCGTCCCAAGCAATG  120
             |  |  ||||||||||||||||||  |  |   |  | |||   ||  |  ||  ||  |  ||  |||
Sbjct  70    GGCTTATGGAAACTACTTCCTCCAGAAGATATCCTG---CCATCACC-TCAC-TGC-ATG  123

Query  121   GATGATTTGATGCTGTCCCCGGACGATATTGAACAATGGTTCACTGAAGACCCAGGTCCA  180
             ||  |||  ||  |||||  ||||   |  |||  |||  |   |   |||    |||||  ||||      |
Sbjct  124   GACGATCTGTTGCTG-CCCC--AGGATGTTG-AGGA--GTTTTTTGAAGGCCCA-----A  172

Query  181   GATGAAGCTCCCAGAATGCCAGAGGCTGCTCCCCCCGTGGCCCCTG-CACCAGCAG-CTC  238
             |  ||||||  |  |  ||  ||  |||   |||  ||  |    |    ||  |||||  ||||  |  ||  |  |
Sbjct  173   G-TGAAGCCCTCCGAGTGTCAGGAGCTCCTGCAGCACAGGACCCTGTCACC-G-AGACCC  229

Query  239   CTACACCGGCGGCCCCTGCACCAGCCCCCTCC-TGGCCCCTGTCATCTTCTGTCCCTTCC  297
             ||    ||  |  |||||||||  ||||||   ||||  ||||||||||||||||  |||||||||
Sbjct  230   CTGGGCCAGTGGCCCCTGCCCCAGCC-ACTCCATGGCCCCTGTCATCTTTTGTCCCTTCT  288

Query  298   CAGAAAACCTACCAGGGCAGCTACGGTTTCCGTCTGGGCTTCTTGCATTCTGGGACAGCC  357
             ||  |||||  ||||||||||  |||  ||  ||||   |||||||||  ||||  ||||||||||||
Sbjct  289   CAAAAAACTTACCAGGGCAACTATGGCTTCCACCTGGGCTTCCTGCAGTCTGGGACAGCC  348

Query  358   AAGTCTGTGACTTGCACGTACTCCCCTGCCCTCAACAAGATGTTTTGCCAACTGGCCAAG  417
             ||||||||  |   ||||||||||||  |||  |||||||  |||  |  ||  |||||  |||||  |||
Sbjct  349   AAGTCTGTTATGTGCACGTACTCTCCTCCCCTCAATAAGCTATTCTGCCAGCTGGCGAAG  408

Query  418   ACCTGCCCTGTGCAGCTGTGGGTTGATTCCACACCCCCGCCCGGCACCCGCGTCCGCGCC  477
             ||  ||||||||||||  |||||||      |||||||  ||   |  ||  |  |||  |||||||||
Sbjct  409   ACGTGCCCTGTGCAGTTGTGGGTCAGCGCCACACCTCCAGCTGGGAGCCGTGTCCGCGCC  468

Query  478   ATGGCCATCTACAAGCAGTCACAGCACATGACGGAGGTTGTGAGGCGCTGCCCCCACCAT  537
             ||||||||||||||  ||||||||||||||||||||||||  |||||  ||||||||||||||||
Sbjct  469   ATGGCCATCTACAAGAAGTCACAGCACATGACGGAGGTCGTGAGACGCTGCCCCCACCAT  528

Query  538   GAGCGCTGCTCAGATAGCGATGGTCTGGCCCCTCCTCAGCATCTTATCCGAGTGGAAGGA  597
             ||||||||||  |||  |  |||||  |||||  |||||  ||||||||||||||||  |||||||||
Sbjct  529   GAGCGCTGCTCCGATGGTGATGGCCTGGCTCCTCCCCAGCATCTTATCCGGGTGGAAGGA  588

Query  598   AATTTGCGTGTGGAGTATTTGGATGACAGAAACACTTTTCGACATAGTGTGGTGGTGCCC  657
             ||||||  |   ||||||  ||||  |||||  |  ||||||||  ||  ||  ||||||||  ||
Sbjct  589   AATTTGTATCCCGAGTATCTGGAAGACAGGCAGACTTTTCGCCACAGCGTGGTGGTACCT  648
```

```
Query  658   TATGAGCCGCCTGAGGTTGGCTCTGACTGTACCACCATCCACTACAACTACATGTGTAAC  717
             |||||||| || ||||  |||||||| | ||||||||||||||||||| ||||||||||
Sbjct  649   TATGAGCCACCCGAGGCCGGCTCTGAGTATACCACCATCCACTACAAGTACATGTGTAAT  708

Query  718   AGTTCCTGCATGGGCGGCATGAACCGGAGGCCCATCCTCACCATCATCACACTGGAAGAC  777
             || |||||||||||  ||||||||||   | || ||||| ||||||||||||||||||||
Sbjct  709   AGCTCCTGCATGGGGGGCATGAACCGCCGACCTATCCTTACCATCATCACACTGGAAGAC  768

Query  778   TCCAGTGGTAATCTACTGGGACGGAACAGCTTTGAGGTGCGTGTTTGTGCCTGTCCTGGG  837
             |||||||| || || ||||||||| |||||||||||||  ||||||||||||||  |||||||
Sbjct  769   TCCAGTGGGAACCTTCTGGGACGGGACAGCTTTGAGGTTCGTGTTTGTGCCTGCCCTGGG  828

Query  838   AGAGACCGGCGCACAGAGGAAGAGAATCTCCGCAAGAAAGGGGAG-CCTCACCAC-GAGC  895
             |||||||| || ||||| ||||| ||| ||||||| |||||  || |||   | | || |
Sbjct  829   AGAGACCGCCGTACAGAAGAAGAAAATTTCCGCAA-AAAGGA-AGTCCTTTGCCCTGAAC  886

Query  896   TGCCCCCAGGGAGCACTAAGCGAGCACTGCCCAACAACACCAGCTCCTCTCCCCAGCCAA  955
             |||||||||||||| | ||| |||| ||||||| |  ||| ||| ||||||||| || ||
Sbjct  887   TGCCCCCAGGGAGCGCAAAGAGAGCGCTGCCCACCTGCACAAGCGCCTCTCCCCCGCAAA  946

Query  956   AGAAGAAACCACTGGATGGAGAATATTTCACCCTTCAGATCCGTGGGCGTGAGCGCTTCG  1015
             |||| |||||||| |||||||| |||||||||||  ||||||| |||||| | |||||||
Sbjct  947   AGAAAAAACCACTTGATGGAGAGTATTTCACCCTCAAGATCCGCGGGCGTAAACGCTTCG  1006

Query  1016  AGATGTTCCGAGAGCTGAATGAGGCCTTGGAACTCAAGGATGCCCAGGCTGGGAAGGAGC  1075
             |||||||||| |||||||||||||||| ||  | |||||||||| |||    |||||
Sbjct  1007  AGATGTTCCGGGAGCTGAATGAGGCCTTAGAGTTAAAGGATGCCCATGCTACAGAGGAGT  1066

Query  1076  CAGGGGGGAGCAGGGCTCACTCCAGCCACCTGAAGTCCAAAAAGGGTCAGTCTACCTCCC  1135
             | || |  |||||||||||||||||| |||||||| |||| ||||| |||||||| ||||
Sbjct  1067  CTGGAGACAGCAGGGCTCACTCCAGCTACCTGAAGACCAAGAAGGGCCAGTCTACTTCCC  1126

Query  1136  GCCATAAAAAACTCATGTTCAAGACAGAAGGGCCTGACTCAGACTGA  1182
             ||||||||||   ||| |||||| ||  ||||||||||||||||||
Sbjct  1127  GCCATAAAAAAACAATGGTCAAGAAAGTGGGGCCTGACTCAGACTGA  1173
```

Each section of the alignment shows the portion of the query and subject that are aligned. In this first sequence, the query sequence from base 1 to base 60 align with the subject sequence from base 10 to base 69. This indicates that the first 9 bases in the subject sequence are not present in the query sequence.

Between each aligned section of the subject and query sequences we see lines where the base match one another, and blanks where there are mismatches and gaps. Gaps are shown with dashes. For example, if you view the alignment of query sequence bases 61-120 and subject sequence 70-123, there are a 6 bases that are present in the *Homo sapiens* (query) sequence that missing from the *Mus musculus* (subject) sequence, creating a total of 4 gaps.

Finally, there is a section at the end of the output that describes the parameters used in the alignment and statistical calculations. These will not be discussed in this course, however more information can be found here:
https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html

```
Lambda      K        H
    1.33    0.621     1.12

Gapped
Lambda      K        H
    1.28    0.460    0.850

Effective search space used: 1356040

  Database: User specified sequence set (Input: Mm_Tp53_CDS.fa).
    Posted date:  Unknown
  Number of letters in database: 1,173
  Number of sequences in database:  1

Matrix: blastn matrix 1 -2
Gap Penalties: Existence: 0, Extension: 2.5
```

It is important to remember when running command line tools that files provided as arguments, such as the `QUERY.fa` and `SUBJECT.fa` files, must be provided with their relative or absolute paths. In the above `blastn` command the file names themselves were provided as the files are in the PWD. If the files were in a different directory, the BLAST command would be unable to find and use the files.

For example, if the same command was run from the parent directory `Week.9` (note the directory in the command prompt), the command will not run as the files are in the directory `9.1.BLAST`.

```
j:~/Week.9$ blastn -query Hs_TP53_CDS.fa -subject Mm_Tp53_CDS.fa
Command line argument error: Argument "subject". File is not accessible:
`Mm_Tp53_gene.fa`
```

However, if the relative paths are provided, the command will run.

```
j:~/Week.9$ blastn -query 9.1.BLAST/Hs_TP53_CDS.fa -subject
9.1.BLAST/Mm_Tp53_CDS.fa
BLASTN 2.12.0+

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb
Miller (2000), "A greedy algorithm for aligning DNA sequences", J
Comput Biol 2000; 7(1-2):203-14.

                                    ...
```

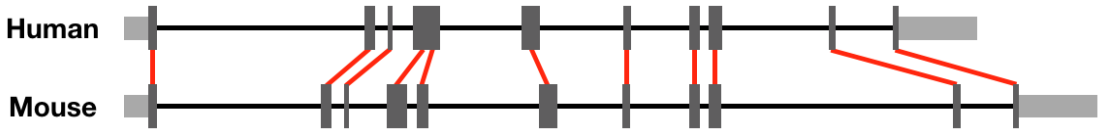
9.1.4 BLAST Arguments

Aligning Gene Sequences

In section 9.1.3 the CDS sequences for two genes were aligned instead of the full gene sequences. The average human gene is 10,000-15,000 base pairs long, but only ~900 of the base pairs are coding sequence (code for the protein product). Therefore, the full gene sequence is generally 10 times longer than the CDS. A gene's coding sequence is contained within the gene's exons. The average human gene has about 9 exons, and most human exons are shorter than 200 base pairs.

CDSs are under strong positive selection because they determine the amino acid sequence of the gene's protein. Therefore, CDSs are generally well conserved between species. Introns and untranslated regions, however, diverge much more rapidly and often do not align well between species. These are important considerations when aligning two gene sequences.

If orthologous human and mouse genes are aligned, ideally a hit would be returned for each pair of matching exons, as displayed in figure above. Due to the long evolutionary time since the divergence of human and mouse alignments between the introns and UTRs are not necessarily expected.



A convenient BLAST option for viewing the output in a different format is `-outfmt`. When run with the number 7 as the argument, only the key information is provided, in tab delimited format with a header. The human TP53 gene sequence is 19,070 base pairs long and the mouse Tp53 gene sequence is 11,514 base pairs long. Align the human and mouse TP53 gene ortholog sequences:

```
j:~/Week.9/9.1.BLAST$ blastn -query Hs_TP53_gene.fa -subject
Mm_Tp53_gene.fa -outfmt 7
# BLASTN 2.12.0+
# Query: hg38_knownGene_ENST00000269305.9 range=chr17:7668421-7687490
5'pad=0 3'pad=0 strand=- repeatMasking=none
# Database: User specified sequence set (Input: Mm_Tp53_gene.fa)
# Fields: query acc.ver, subject acc.ver, % identity, alignment length,
mismatches, gap opens, q. start, q. end, s. start, s. end, evalue, bit
score
# 1 hits found
hg38_knownGene_ENST00000269305.9   mm39_knownGene_ENSMUST00000108658.10
86.567  134     17      1       13184   13317      8766    8898
1.22e-36        147
# BLAST processed 1 queries
```

The 4th line of the output explains the fields of the tab delimited results. In detail these fields are (and in brackets the value from the above alignment):

- query description (hg38_knownGene_ENST00000269305.9)
- subject description (mm39_knownGene_ENSMUST00000108658.10)
- percent identity (86.567)
- alignment length (134)
- # of mismatches (17)
- # number of gaps (not the number of bases in gaps) (1)
- the base in the query at which the alignment starts (13184)
- the base in the query at which the alignment ends (13317)
- the base in the subject at which the alignment starts (8766)
- the base in the subject at which the alignment ends (8898)
- the Expect-value (1.22e-36)
- the alignment score in bits (147)

The CDSs of human and mouse TP53 orthologs have 81% identity with nearly all bases in both sequences contained in the 1187 base alignment (see 9.1.3). However, only a single 134 base alignment is returned when the entire gene sequences are aligned, despite the entire CDSs being contained within the gene sequence.

The BLAST algorithm will only begin to align two sequences if at least 28 bases match exactly, this is called the BLAST **word size**. Thus, if two genes with a pair of perfectly matching exons are aligned, if the exons are less than 28 bases long, no hit will be returned by blast. Similarly, if two 100 base exons have a mismatch every 10 bases, they would have 90% identity, but no hit will be returned by BLAST, as there is no continuous match of 28 bases.

The results of the alignment of the human TP53 gene and the mouse Tp53 gene indicates that only one pair of exons between the two species have 28 bases that align perfectly, an exon from base 13184-13317 in the human gene and 8766-8898 in the mouse gene.

The BLAST word size can be altered with the `-word_size` option, which has a default value of 28. Performing the gene alignment again with the word size set to 20 increases the number of hits to 6:

```
j:~/Week.9/9.1.BLAST$ blastn -query Hs_TP53_gene.fa -subject
Mm_Tp53_gene.fa -outfmt 7 -word_size 20
# BLASTN 2.12.0+
# Query: chromosome:GRCh38:17:7661179:7688138:-1
# Database: User specified sequence set (Input: Mm_Tp53_gene.fa)
# Fields: query acc.ver, subject acc.ver, % identity, alignment length,
mismatches, gap opens, q. start, q. end, s. start,s. end, evalue, bit
score
# 6 hits found
hg38_knownGene_ENST00000269305.9   mm39_knownGene_ENSMUST00000108658.10
78.240  409     73      10      12883      13285   8587    8985
5.09e-67        248
hg38_knownGene_ENST00000269305.9   mm39_knownGene_ENSMUST00000108658.10
73.453  501     91      26      1171       1656    1167    1640
1.48e-37        150
hg38_knownGene_ENST00000269305.9   mm39_knownGene_ENSMUST00000108658.10
82.286  175     26      5       12019      12189   7748    7921
1.91e-36        147
```

```
hg38_knownGene_ENST00000269305.9   mm39_knownGene_ENSMUST00000108658.10
86.567  134     17      1       13832       13965   9366    9498
1.91e-36        147
hg38_knownGene_ENST00000269305.9   mm39_knownGene_ENSMUST00000108658.10
86.813  91      12      0       11539       11629   7136    7226
4.19e-23        102
hg38_knownGene_ENST00000269305.9   mm39_knownGene_ENSMUST00000108658.10
86.486  37      4       1       4885        4921    10420   10455
3.33e-04        39.9
# BLAST processed 1 queries
```

When the word size is decreased, the number of matches, including less significant is increased. The option -evalue sets the Expect-value cutoff, the highest Expect-value a match can have to be counted as a hit. The default value is 10. Remember that an Expect-value of 10 means one would expect 10 matches of that quality to found by chance.

Reducing the word size further to 15 returns 26 hits.

```
j:~/Week.9/9.1.BLAST$ blastn -query Hs_TP53_gene.fa -subject
Mm_Tp53_gene.fa -outfmt 7 -word_size 15 | head -n 5 | tail -n 1
# 26 hits found
```

To ensure only significant hits are returned, the E-value threshold can be reduced to 0.0001, returning only 14 hits.

```
j:~/Week.9/9.1.BLAST$ blastn -query Hs_TP53_gene.fa -subject
Mm_Tp53_gene.fa -outfmt 7 -word_size 15 -evalue 0.0001 | head -n 5 |
tail -n 1
# 14 hits found
```

Although the settings can be tinkered with to improve gene alignments, to determine the similarity of the coding capacity of two genes, it is better to align the CDSs. Aligning full gene sequences, however, is necessary to view intronic and UTR similarities.