**Genome Solver BLAST Introduction**

BLAST is a key tool for investigating similarity between and among sequences. This slide deck “Basic Local Alignment Search Tool (BLAST)” serves as an introduction to the tool, how to use it, and how it works.

Slide 1: Understanding BLAST (Basic Local Alignment Search Tool)

Slide 2: BLAST aims to identify regions of similarity between two sequences

Slide 3: The reason similarity is important is that it might tell you something about evolutionary relatedness. Sequences that are similar are more likely to have arisen by divergence from a common ancestor than by convergence on similar sequences.

Slide 4: A few definitions:

Paralogs are genes that have arisen from duplication within a species – an example is the globin genes that encode the subunits of hemoglobin

Orthologs are ones that are similar between species – an example is the HBB gene, encoding beta-globin in humans

Slide 5: Regions of genomes that are important for function, such as protein encoding regions, are likely to be conserved, i.e., having similar sequences in closely related organisms. In the figure shown here, an alignment of a set of transcription factor proteins in the Myb family is documented. The conservation among the sequences is highlighted by the color scheme – yellow meaning conserved in all members, while blue and green means reasonably well conserved. Based on this alignment, the consensus sequence, at the bottom is generated – the amino that occurs most frequently in that spot (column) in the relatives. A blank means that the amino acid in that spot is variable.

Slide 6: Another reason for examining sequence similarity is that by comparing sequences of unknown function to ones that are similar of known function, one can help determine the function of the unknown sequence.

Slide 7: Aligning sequences

Note – there is more about this topic in **Lesson III – Annotation**, and especially **Lesson IV – Comparative Genomics.**

Slide 8: Sequence alignment depends on “counting up” the places where two sequences match. For this protein sequence using this representation, the same amino acid is represented by a ‘\*’, a substitution of a closely related amino acid (i.e., E for Q) is represented by a ‘:’, a substitution of a less closely related amino acid (i.e., T for P) is represented by a ‘.’, and a blank shows these two amino acids are not in the same family. The idea is to bring the two sequences in register so that the number of changes between the two sequences in minimized.

Slide 9: So how does BLAST actually do this? With two sequences, you can try to align by eye, but there are now millions of sequences in the databases (more about databases in Lesson II). The BLAST algorithm allows for rapid searching of your ‘query’ sequences against the sequences in the database.

Slide 10: The BLAST tools is freely available at <https://blast.ncbi.nlm.nih.gov/Blast.cgi>. There are a number of related tools, but basically one can search for either proteins or nucleotide sequences with either protein or nucleotide sequences.

Slide 11: Once you choose the type of BLAST you’re interested in, you’ll get a page that looks like this. In the next several slides, we’ll illustrate the general procedure using a protein sequence.

Slide 12: In this example, we’re looking a a protein and breaking it up into words that are 3 letters long (this word length is a parameter that can be determined by the user, but for this example, we’ll say the words are 3 letters long. The entire protein is broken down into overlapping three letter words (bottom left). Then each word is used to generate a list of closely related or ‘neighborhood’ words (using GTW as the example).

Slide 13: The matches are then given scores based on how well they match. For proteins we’re taking into account the identity of the amino acid, but also the similarity – for example, replacement of S with T would get a higher score (they’re both hydroxyl containing side chains) than replacement of S with W (a small hydroxyl containing side chain with a bulky aromatic side chain).

Slide 14: The same process is repeated for all the overlapping 3-letter words in the protein.

Slide 15: Then the database is scanned to look for words (close matches) from the list.

Slide 16: Once a match is found, then the word is “extended” to examine the extent to which the two sequences are similar.

Slide 17: How to perform a BLAST search

Slide 18: Initially, one starts with a sequence of interest. The choice of this sequence depends on the question you want to answer.

Sequences can be downloaded from primary nucleotide or protein databases – more about how to acquire sequences in **Lesson II – Databases**.

Slide 19: The sequence input needs to be in a format that the program can understand. More about Accession Numbers in **Lesson II – Databases.**

Slide 20: Then one chooses the BLAST program of interest. For this example, we will be using a protein sequence and then using BLASTP to look for similar proteins.

Slide 21: Working with the BLAST interface

Slide 22: There are a number of different versions of BLAST, using either nucleotide or protein sequence data and comparing to either nucleotide or protein sequence databases. In addition, there are different versions that take into account the evolutionary distance you’re wanting to cover in your search (i.e., are you look for the rat ortholog of a mouse protein? or are you looking for the yeast ortholog of a mouse protein?).

**Blastp** compares a protein query to a database of proteins

**Blastn** compares a nucleotide query to a database of nucleotides

**Blastx** translates the DNA into six protein sequences using all reading frames and compares them to a protein database-if you have a DNA sequence and you want to know what protein it encodes

**tBlastn** compares a protein query sequence against a nucleotide sequence database dynamically translated into all reading frames- does a protein query yield any matches in a database of genomic DNA fragments from the genome sequencing project of a particular organism.

**tBlastx** compares the 6 frame translations of a nucleotide query against the 6 frame translations of a nucleotide sequence database

Slide 23: Performing a BLAST search is easy, simply paste the sequence (or Accession Number) into the box and hit submit.

Slide 24: Some additional details for nucleotide blast programs

Slide 25: Some additional details for protein BLAST versions

Slide 26: Once you’ve chosen the type of analysis you’d like to perform, the next step is to choose a database.

Slide 27: There are a number of different databases to choose from. These are available via pull-down menus on the query page.

Slide 28: Next, you have the ability to modify your search parameters from the default settings. The default works well if you’re looking for sequences that are relatively close (i.e., you’re looking for other mammalian sequences or perhaps other vertebrates, but not invertebrates).

Slide 29: However it pays to know something about how the algorithm works to get the results most appropriate to your question. There are a number places where the user can alter the default settings. These include the database, the organism(s), exclusions, and the Program Selection.

Slide 30: Some of the other options in more detail.

Slide 31: Now – how to analyze the results? Once you’ve submitted a sequence to the program, you’ll end up with a new page with details to interpret.

Slide 32: At the top of the page is a reiteration of what you’ve told the program to do (red ovals). Next comes a graphic that tells you matches. The best matches are red, then pink, and so on.

Slide 33: From this example, we can see that there are several hits that match very well over the entire length, but lower on the graphic are matches that begin to lose some of the material in the middle, and then even lower, change color.

Slide 34: Scrolling down below the graphic on the previous slides reveals a table. This table has a lot of information that this useful. At left is the NCBI Accession Number assigned to each sequence pulled up from the database as a match. More about Accession Numbers in Lesson II – Databases

Slide 35: Next comes a description of the sequence, species (strain if bacterial), etc.

Slide 36: At the right are a series of scores. The best hit is always the first one. It has the highest max score and the lowest e-value. Depending on the database chosen, this might be the original query sequence (in other words, if the query is in the database, then it will find itself as a perfect match).

Slide 37: Sometimes the match does not extend for the entirety of the query. Depending on your question, this might be an important feature to consider.

Slide 38: The last two parameters are E-value and Max Identity.

Slide 39: Scrolling down below the table reveals the individual matches between the hits in the table and the query sequence. Some of the information from the table is repeated, but in addition, there are details that are specific to this graphic (identities, gaps, etc.)

Slide 40: Another important feature to note is where the query matches the subject. This could be important if you need to map the match onto the chromosome.

Slide 41: Picking the best hit could simply involve scrutinizing the data to make sure the line at the top is not the query itself. However, again, depending on the question posed, the interpretation may be more involved than this simple example.

Slide 42: Some additional considerations for choosing how to interpret the data.