



Earlham  
Institute

# Earlham Institute summer school on bioinformatics

25-29 July 2016

## Basic Bioinformatics Sessions

### Practical 3: Databases Searching

Wednesday 6 July 2016

## Searching for sequence similarities in databases.

The most popular way to investigate a sequence has always been to compare it with one of the sequence databases now accessible from sites all over the world. When sequences databases were more sparsely populated than now, the objective was to search hopefully, not always with success, for any convincingly similar sequence(s). When such a match was discovered, it could be supposed that known properties of the “similar” database sequence might provide insight to the properties of the query sequence. Now, the databases are full of sequences representative of most interesting conditions. Similarity searches are conducted in the expectation of finding many close “hits” for almost any sequence. Fewer database searches are conducted in complete ignorance of what the query sequence might be.

Here, take the **PAX6** genomic DNA sequence retrieved from **Ensembl** and conduct two searches analogous to those run in the **Ensembl** pipeline. Results should confirm that which has already been discovered using other sources.

**blast** is not the only sequence database searching program available, but it is the most popular by a very long way. **blast** searches are offered in many forms by many servers all over the world, but the most comprehensive and reliable service has to be that offered by the **NCBI**.

Go to the **NCBI** homepage at:

<http://ncbi.nlm.nih.gov>

Select the **BLAST** option (from the **Popular Resources** list). In the **Basic BLAST** section, select **nucleotide blast**. Use the **Enter Query Sequence** **Browse** (or **Choose File**) button to upload the file:

**pax6\_genomic.fasta**.

For results like those used by **Ensembl** to predict **PAX6** transcripts, you must compare your genomic sequence to a reliable set of human mRNA/cDNA (or similar) sequences.

In the **Choose Search Set** section, set the **Database** to **Reference RNA sequences (refseq\_rna)**.

You are now able to specify an **Organism**, choose **Human**.

**blast** is now set to compare the **PAX6** genomic region with all **Human** mRNA sequences in **RefSeq**.

The screenshot shows the NCBI BLASTN web interface. The 'Enter Query Sequence' section has a text box for 'Enter accession number(s), gi(s), or FASTA sequence(s)' and a 'Browse...' button next to 'pax6\_genomic.fasta'. The 'Choose Search Set' section has a 'Database' dropdown set to 'Reference RNA sequences (refseq\_rna)' and an 'Organism' dropdown set to 'human (taxid:9606)'. The 'Program Selection' section has 'Optimize for' set to 'Highly similar sequences (megablast)'. The 'BLAST' button is visible at the bottom.

Note that the default **Program Selection** is **Highly similar sequences (megablast<sup>1</sup>)**, which seems appropriate here as all the mRNA that correctly match should surely do so almost perfectly.

<sup>1</sup> **megablast** is a less sensitive but even faster version of **blast** only suitable when, as now, almost identical matches are sought.

Click on the **Algorithm Parameters** button. The defaults are fine here, but before starting your search, try changing the **Program Selection** and observing the different **Algorithm Parameters**.

The default settings of all shared parameters are identical for the two slower more sensitive **Program Selections**.

There are differences for **megablast**, where speed is of the essence and sensitivity can be sacrificed.

Smaller **Word sizes** slow searches but increase sensitivity. For **megablast** the default **Word size** is 28 otherwise it is 11.

Gapped alignment is time consuming and, by default, considered more crudely by **megablast** than the other two algorithms<sup>2</sup>.

**Filtering and Masking** matches with organism specific repeats and/or low complexity regions takes time, and so only avoiding **Low complexity regions**<sup>3</sup> is on by default for all **Program Selections**.

When **discontinuous megablast** is selected, an extra options section appears. Discussing how this flavour of **blast** works is a little beyond the scope of these notes, but briefly. Unlike the other **Program Selections**, **discontinuous megablast** does not just look for exactly matching “words” of given size as a first step towards identifying matching regions between sequences. It looks for a pattern of matching bases within a word. For example, the default choice assumes your query is **coding** and looks for 11 matching bases within a word of 18. Approximately, every third base is allowed not to match. Biologically, this can be justified as allowing for third codon position wobble. For more detail, use the appropriate button. Notice there are buttons by every parameter selection. Try one or two. In the process, discover:

When would **Mask lower case letters** be a useful thing to do? \_\_\_\_\_

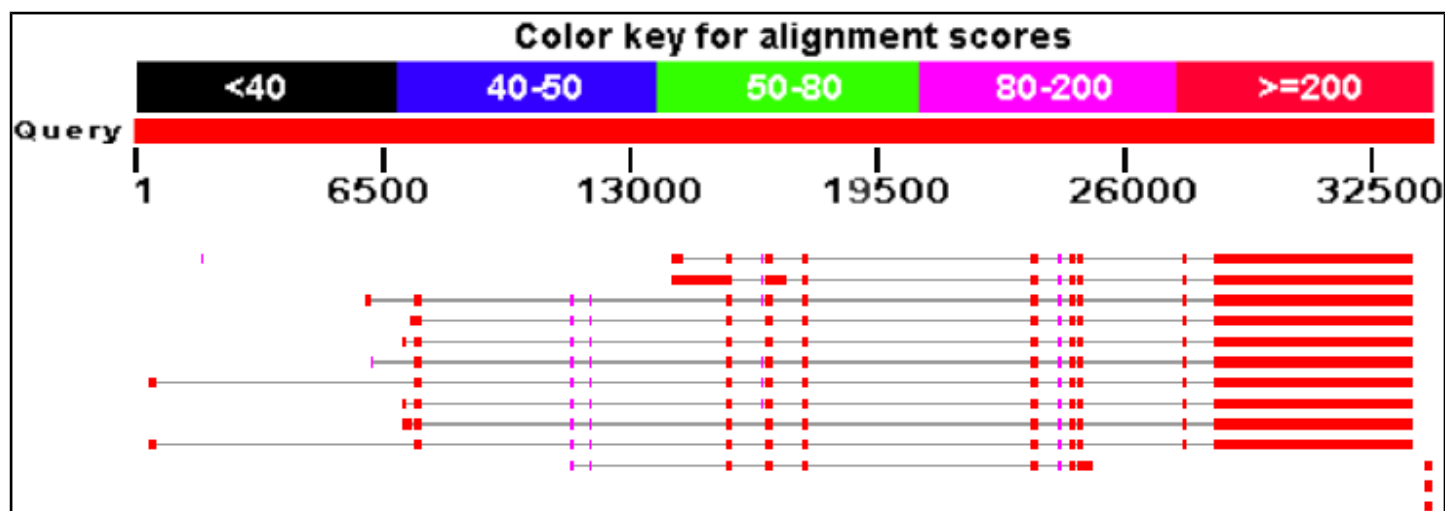
**Automatically adjust parameters for short input sequences** is independent of **Program selection**, and so remains unaltered.

Which parameters would **blast** need to **automatically adjust** to cater for short input sequences (such as primers being tested for uniqueness), and why? \_\_\_\_\_

<sup>2</sup> By default, **megablast** uses **Linear Gap Costs**. That is, it just multiplies the size of the gap with the **Mismatch** penalty. The other two algorithms employ the more common **Affine** strategy, using **Existence** and **Extension** penalties. For more about **Gap Penalties**, go [here](#).

<sup>3</sup> This filter avoids finding “hits” supported only by matches in regions not specific to the query. For example, a polyA tail cannot help to identify a specific mRNA as it is present in all mRNAs. The use of this filter will be evident when we look at the **blast** output.

Finally, ensure all the defaults are back in place<sup>4</sup> and the **megablast** is the **Program Selection**, ask **blast** to **Show results in a new window** and then click on the **BLAST** button. Impressively swiftly, you will have results. At the top of which will be a graphical overview.



This graphic implies that there are **11** full length matches between the genomic sequence and mRNAs in **RefSeq**. The **RefSeq** entries had to be “gapped” in order to compensate for the introns that are represented in the genomic sequence but not in the mRNA sequences. The **red blocks** therefore represent very closely matching (>=200 brownie points) exons, the lines joining the **red blocks** represent introns that have been spliced out. All **11** full length hits match reasonably uniformly except for the first few exons, implying significant variation in the **5' UTR**.

Why do you suppose that a few of the exons of the first 11 matches do not achieve the maximum score? \_\_\_\_\_

Explain why one exon in the reasonably consistent region, does not appear in all of the transcript matches? \_\_\_\_\_

In a previous Practical, you discovered directly that there were **11** high quality “NM\_” **PAX6** transcripts in **RefSeq**.

Until recently, there was a further **9** “XM\_” **PREDICTED** transcripts. However, in the last release of **RefSeq**, the **9** less reliable **XM\_** transcripts were removed. Nevertheless, you saw that the **9 PREDICTED** transcripts are still evidenced in **MapViewer**. That is because the current version of **MapViewer** is built on a previous version of **RefSeq**. As soon as **MapViewer** is updated, the dubious **9** will disappear. Then, I will have to discover another way to introduce the way that **RefSeq** divides its transcript entries by quality.

**Ensembl** claimed to have used **10** or the **11** high quality **NM\_ RefSeq** sequences to aid its transcript predictions. **Ensembl** would have ignored the **XM\_ PREDICTED RefSeq** sequences even if they still existed.

**blast** just sees sequences and cannot be influenced by the quality of the support for their existence, so **blast** will always report all **RefSeq PAX6** mRNAs matching the **PAX6** genomic region convincingly, however questionably they are evidenced.

There is a point to pursuing all this detail. You reference a collection of interdependent databases, all of which are updated regularly. More often than not you will notice inconsistencies due to asynchronous updates and differences in database management policy. A small price to pay for such a rich source of information, but one of which I suggest it is wise to be aware.

The message of the particular **blast** search here is that it is so easy to predict the same **PAX6** transcripts as you discovered in **MapViewer**, just with a simple **blast** search. That is, you can look things up, or work most of it out for yourself.

<sup>4</sup> If you have any non-default settings, they should be highlighted in yellow.

If you hover over the graphical hits, their origin will be displayed above the graphic<sup>5</sup>.

Below the **Graphic Summary** are the **Descriptions**, a simple list of the **15** matches represented in the graphic.

	Description	Max score	Total score	Query cover	E value	Ident	Accession
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 11, mRNA</a>	9659	12484	19%	0.0	99%	<a href="#">NM_001310161.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 10, mRNA</a>	9659	15161	24%	0.0	99%	<a href="#">NM_001310160.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 8, mRNA</a>	9659	12929	20%	0.0	99%	<a href="#">NM_001310158.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 7, mRNA</a>	9659	12729	20%	0.0	99%	<a href="#">NM_001258465.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 6, mRNA</a>	9659	12761	20%	0.0	99%	<a href="#">NM_001258464.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 5, mRNA</a>	9659	12737	20%	0.0	99%	<a href="#">NM_001258463.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 4, mRNA</a>	9659	12862	20%	0.0	99%	<a href="#">NM_001258462.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 2, mRNA</a>	9659	12833	20%	0.0	99%	<a href="#">NM_001604.5</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 1, mRNA</a>	9659	12942	20%	0.0	99%	<a href="#">NM_000280.4</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 3, mRNA</a>	9659	12791	20%	0.0	99%	<a href="#">NM_001127612.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens paired box 6 (PAX6), transcript variant 9, mRNA</a>	647	2630	4%	0.0	100%	<a href="#">NM_001310159.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens elongator acetyltransferase complex subunit 4 (ELP4), transcript variant 3</a>	433	433	0%	8e-118	100%	<a href="#">NM_001288726.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens elongator acetyltransferase complex subunit 4 (ELP4), transcript variant 2</a>	433	433	0%	8e-118	100%	<a href="#">NM_001288725.1</a>
<input type="checkbox"/>	<a href="#">Homo sapiens elongator acetyltransferase complex subunit 4 (ELP4), transcript variant 1</a>	433	433	0%	8e-118	100%	<a href="#">NM_019040.4</a>
<input type="checkbox"/>	<a href="#">Homo sapiens uncharacterized LOC440034 (DKFZp686K1684), long non-coding RNA</a>	141	141	0%	5e-30	100%	<a href="#">NR_033971.1</a>

These are such that:

- The top **11** hits, corresponding to the **11** full length hits of the **Graphic Summary**, are the quality (i.e. **NM\_** entries with good supporting evidence) **RefSeq** transcripts.
- There follows, corresponding to the **3** small **red blobs** in the extreme bottom right of the **Graphic Summary**, **3** hits that are the ends of **mRNAs** for the **ELP4** gene. They are exactly where you should expect them to be, assuming you paid full attention to the **ELP4** transcript predictions shown in both the **Ensembl** and **MapView** displays of the **Genomic** region around **PAX6**. Reject these contemptuously, they do not pertain to our investigation of **PAX6**.
- The **15<sup>th</sup>** match, corresponding to the barely visible tiny smudge match to the left of the top **Graphic Summary** hit, is recorded as “**uncharacterized**” and fails to fit in with my story, so I ignore it!

So, this **blast** search suggests the existence of **11 PAX6** transcripts supported by **RefSeq** data, as will be reported by **MapView** as soon as the **NCBI** get around to bringing it up to date! Also, the results are consistent with the information discovered in **Ensembl**.

Which of the **Refseq PAX6** transcripts corresponds to **isoform 5a**? \_\_\_\_\_

<sup>5</sup> Or you could just read the textual list that follows the graphic if you wish to insist on the simplistic.

Moving further down the results you will come to the alignments between the **PAX6** genomic sequence and the matching database entries. All similarity searches use local alignment strategies<sup>6</sup>, so you should not be surprised to see a number of alignments for each “hit” in the list. Here we have a genomic query sequence aligned exclusively with mRNA sequences from **RefSeq**. The expectation is therefore to find an alignments corresponding to exons. The alignments are ordered by quality, though you are provided with a **Sort by:** menu to alter the order to taste<sup>7</sup>.

Look at the first alignment for the best matching **PAX6** transcript. It is the alignment of the very last exon of a **RefSeq** transcript with the end of the gene you exported from **Ensembl**.

Notice the lower case string of 'a's. The case indicates that they were ignored (**filtered**) as a **Low complexity region** whilst **megablast** was looking for identically matching words that might suggest matching regions. By themselves, the 'a's are

not sufficient evidence that a biological match exists. Only because the surrounding sequence is compellingly similar, can it be assumed that such a match does exist. The 'a's are replaced (lower case to indicate they were filtered) when the final alignment is computed. If you look a little further down the same alignment, you will see several other runs of 'a's and 't's for which the same explanation applies.

Score		Expect	Identities	Gaps	Strand
9659 bits(5230)		0.0	5237/5240(99%)	2/5240(0%)	Plus/Plus
Query	28433	CCACTTC - - TAGGACTCATTTCCCCTGGTGTGTCAGTTCAGTTCAAGTTCCCGGAAGTG			28498
Sbjct	1490	CCACTTCAACAGGACTCATTTCCCCTGGTGTGTCAGTTCAGTTCAAGTTCCCGGAAGTG			1549
Query	28491	AACCTGATATGTCTCAATACTGGCCAAGATTACAGTaaaaaaaaaaaaaaaaaaaaaaG			28550
Sbjct	1550	AACCTGATATGTCTCAATACTGGCCAAGATTACAGTAAAAAAAAAAAAAAAAAAAAAAG			1609
Query	28551	GAAAGGAAATATTGTGTTAATTCAGTCAGTGACTATGGGGACACAACAGTTGAGCTTTCA			28610
Sbjct	1610	GAAAGGAAATATTGTGTTAATTCAGTCAGTGACTATGGGGACACAACAGTTGAGCTTTCA			1669


<sup>6</sup> To use a global approach would be to imply that you were only interested in database entries that matched your query sequence from end to end. Generally, this is not true. You would usually be interested in a database sequence that was similar over any significant region.

<sup>7</sup> Why not try them? End up with the alignments for the top hit in **E value** order.



Now use a version of **blast** (called **blastx**) to compare your genomic sequence with a protein database. **blastx** will translate a DNA query sequence in all six reading frames and compare each translation with a protein sequence database. Thus, in a similar fashion to that employed by the **Ensembl** pipeline, protein coding regions of the genomic DNA can be identified. For clarity, we will use only the well annotated human proteins of the **SwissProt** section of **Uniprot**. First go to the home of **blast** at:

<http://blast.ncbi.nlm.nih.gov/Blast.cgi>

Select . Use the **Enter Query Sequence Browse (or Choose File)** button to upload file **pax6\_genomic.fasta**.

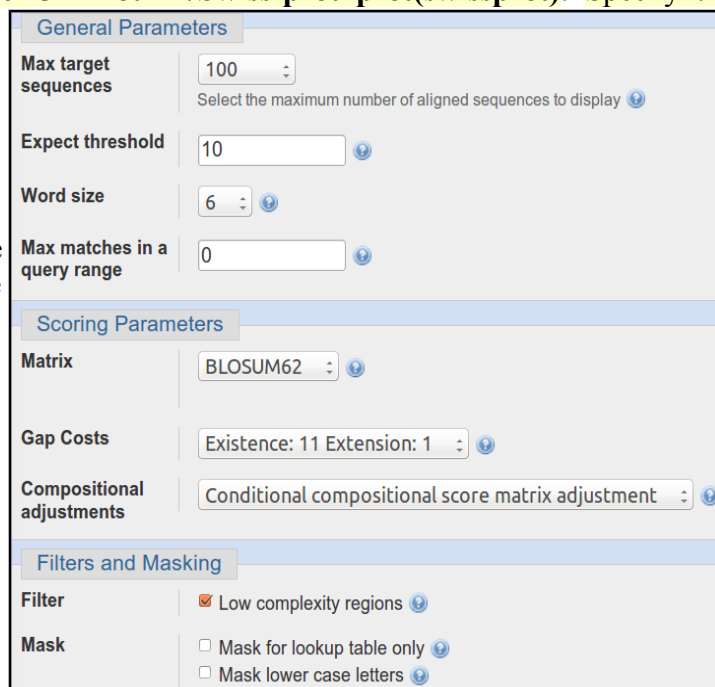
In the **Choose Search Set** section, set the **Database** to **UniProtKB/Swiss-prot prot(swissprot)**. Specify the **Organism** as **Human**.

Take a look at the **Algorithm parameters**<sup>8</sup>.

The **Word size** choice is **2, 3** or **6**. The default is **6**. We seek very close matches here, so the largest **Word size** would seem appropriate.

The default scoring matrix is **BLOSUM62**, but choices from both the **BLOSUM** and **PAM** families are offered.

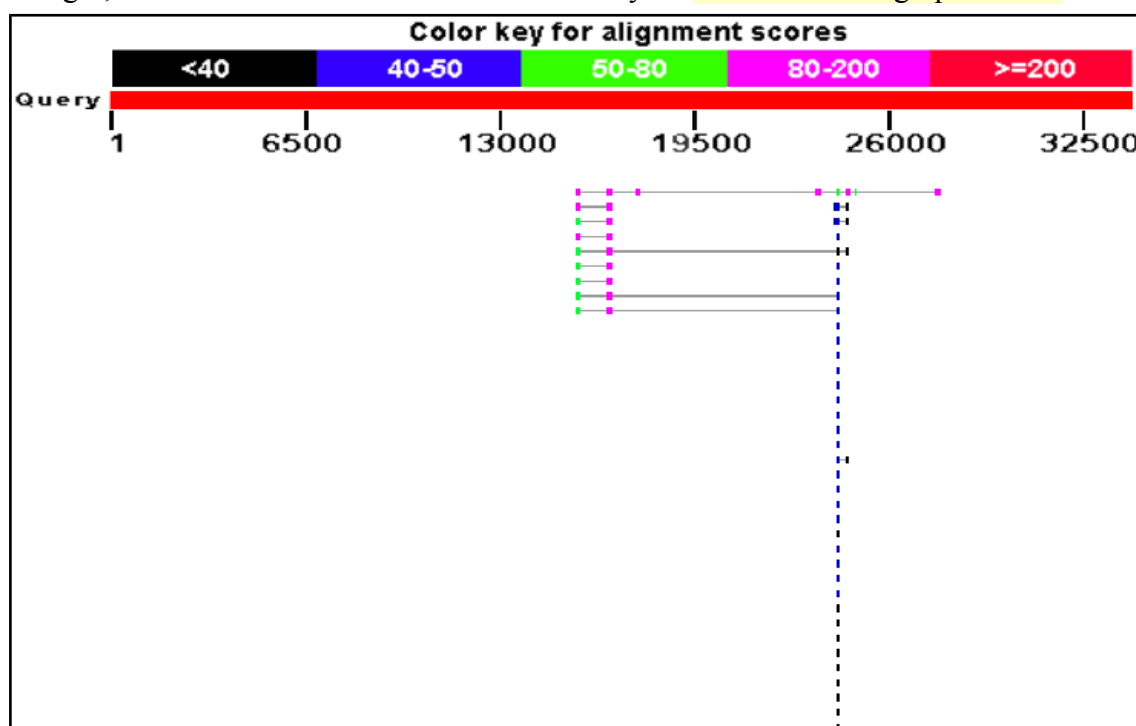
**Low complexity regions** will be filtered by default.



The screenshot shows the NCBI BLAST search interface. The **General Parameters** section includes: Max target sequences (100), Expect threshold (10), Word size (6), and Max matches in a query range (0). The **Scoring Parameters** section includes: Matrix (BLOSUM62), Gap Costs (Existence: 11, Extension: 1), and Compositional adjustments (Conditional compositional score matrix adjustment). The **Filters and Masking** section includes: Filter (Low complexity regions checked) and Mask (Mask for lookup table only and Mask lower case letters unchecked).

Change nothing other than to ask **blast** to **Show results in a new window**, and click the **BLAST** button.

After minimal thought, **blastx** will thrust its conclusions before you. Hover over the graphical hits for identification.



What are the 9 stronger matches around base position 16,000?

Why would you expect exactly 9 matches around this point?

<sup>8</sup> Here I will assume we have talked about these parameter and you are reasonably well informed of the issues.

What do you make of the plethora of matches around **24,000**?

Move down to the textual list of the matches. Hopefully as you fully expected you will find the expected number of **Paired box** matches at the top of the list followed by many many **Homeobox** matches.

Alignments Download GenPept Graphics							
	Description	Max score	Total score	Query cover	E value	Ident	Accession
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-6; AltName: Full=Aniridia type II protein; AltName: Full=Ocul	160	767	3%	3e-41	97%	<a href="#">P26367.2</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-2	131	214	1%	2e-31	74%	<a href="#">Q02962.4</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-8	131	208	1%	4e-31	76%	<a href="#">Q06710.2</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-5; AltName: Full=B-cell-specific transcription factor; Short=B	128	211	1%	1e-30	74%	<a href="#">Q02548.1</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-4	117	258	1%	5e-27	67%	<a href="#">Q43316.1</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-9	112	179	1%	1e-25	69%	<a href="#">P55771.3</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-1; AltName: Full=HuP48	111	177	1%	4e-24	69%	<a href="#">P15863.4</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-3; AltName: Full=HuP2	107	219	1%	7e-23	65%	<a href="#">P23760.2</a>
<input type="checkbox"/>	RecName: Full=Paired box protein Pax-7; AltName: Full=HuP1	105	217	1%	3e-22	68%	<a href="#">P23759.4</a>
<input type="checkbox"/>	RecName: Full=Retinal homeobox protein Rx; AltName: Full=Retina and anterior neural fold homeo	48.9	84.7	0%	1e-04	46%	<a href="#">Q9Y2V3.2</a>
<input type="checkbox"/>	RecName: Full=Retina and anterior neural fold homeobox protein 2; AltName: Full=Q50-type retinal	46.2	80.5	0%	2e-04	48%	<a href="#">Q96IS3.1</a>
<input type="checkbox"/>	RecName: Full=Homeobox protein aristaless-like 4	47.4	47.4	0%	4e-04	68%	<a href="#">Q9H161.2</a>
<input type="checkbox"/>	RecName: Full=Paired mesoderm homeobox protein 1; AltName: Full=Homeobox protein PHOX1; /	45.8	45.8	0%	7e-04	68%	<a href="#">P54821.2</a>
<input type="checkbox"/>	RecName: Full=Paired mesoderm homeobox protein 2; AltName: Full=Paired-related homeobox pr	45.8	45.8	0%	7e-04	68%	<a href="#">Q99811.2</a>
<input type="checkbox"/>	RecName: Full=Dorsal root ganglia homeobox protein; AltName: Full=Paired-related homeobox pro	45.8	45.8	0%	7e-04	71%	<a href="#">A6NNA5.1</a>
<input type="checkbox"/>	RecName: Full=Homeobox protein ARX; AltName: Full=Aristaless-related homeobox	46.6	46.6	0%	0.001	68%	<a href="#">Q96QS3.1</a>

Why do you suppose the **Paired box** matches precede the **Homeobox** matches?

How do you suppose the **Max matches in a query range** parameter might be of value if this order was reversed?

Take a look at the alignments. You will see many places where regions have been filtered as non-informative. I suggest the one illustrated was filtered because it would match anywhere that was sufficiently **Serine** rich.

Score	Expect	Method	Identities	Positives	Gaps	Frame
81.3 bits(199)	5e-29	Compositional matrix adjust.	51/52(98%)	51/52(98%)	0/52(0%)	+3
Query 24654	FQWFSNRRAKWRREEKLRNQRRQASNT	tpshipisssfstst	VYQPIQPQTTTP	24809		
	QWFSNRRAKWRREEKLRNQRRQASNT	PSHIPISSSFS	SVYQPIQPQTTTP			
Sbjct 254	IQWFSNRRAKWRREEKLRNQRRQASNT	PSHIPISSSFS	SVYQPIQPQTTTP	305		

How does this “non-informative” region match expectations suggested by **SMART** and the **Feature table** of **UniprotKB** for **PAX6\_HUMAN**?

THE END

DPJ – 2016.07.06



## Model Answers to Questions in the Instructions Text.

### Notes:

For the most part, these “**Model Answers**” just provide the reactions/solutions I hoped you would work out for yourselves. However, sometime I have tried to offer a bit more back ground and material for thought? Occasionally, I have rambled off into some rather self indulgent investigations that even I would not want to try and justify as pertinent to the objective of these exercises. I like to keep these meanders, as they help and entertain me, but I wish to warn you to only take regard of them if you are feeling particularly strong and have time to burn. Certainly not a good idea to indulge here during a time constrained course event!

Where things have got extreme, I am going to make two versions of the answer. One starting:

### Summary:

Which has the answer with only a reasonably digestible volume of deep thought. Read this one.

The other will start:

### Full Answer:

Beware of entering here! I do not hold back. Nothing complicated, but it will be long and full of pedantry.

This makes the Model answers section very big. **BUT**, it is not intended for printing or for reading serially, so I submit, being long and wordy does not matter. Feel free to disagree.

From your investigations of Searching for sequence similarities in databases

When would **Mask lower case letters** be a useful thing to do?

Generally, whenever one might suspect the automatic masking algorithms of **blast** might miss a non informative region in a specific query sequence, obviously.

A specific example might be when a query sequence contained a significant informative region that was known to be common amongst the sequences being searched. If this region was left unmasked, **blast** would pick up so many similar matches to this one region that other interesting similarities might be obscured. By manually masking such a region by changing it to lower case, its matches would not be seen by **blast** and matches with other regions of the query sequence should be more apparent.

Which parameters would **blast** need to **automatically adjust** to cater for short input sequences (such as primers being tested for uniqueness), and why?

The **word size**: Clearly, if you are trying to find matches for a primer (for example) of around **20** base pairs, it would be pretty silly to use a **word size** of **28** (default for **megablast**). A **word** the same size as the primer would find only exact matches. A **word** of about **7** would allow a couple of mismatches and would probably be most generally appropriate.

The **expect score**: As good chance matches between between a short query sequence and a large database will be abundant, it would not be sensible to choose a demanding (i.e. small) **expect score** to represent the limit of significance. In particular, a primer sized query sequence of around **20** base pairs might easily exactly match more than **10** times (generally the default maximum expect score for a significant match) just by chance. After all, there are only **4** bases, a string of **20** is not that long and the databases can be huge! Typically **blast** chooses very high **expect score** cut off for short query sequences, effectively removing the **expect score** filter altogether.

Earlier versions of **blast** did not automatically adjust these parameters. When a short query sequences were selected, suitable adjustment was left to the user. Without sensible parameter adjustment, results could be greatly confusing. For example, a **21** base pair primer could easily match perfectly more than **10** times against a large DNA sequence database. **blast** is set to ignore matches that are expected to occur more than **10** times by chance. Thus even exact matches with such a small sequences would be ignored! Now automatic parameter adjustment is undertaken by **blast**, the user does not really have to think too hard. However, it does seem to be a good idea to know what **blast** is doing and why.

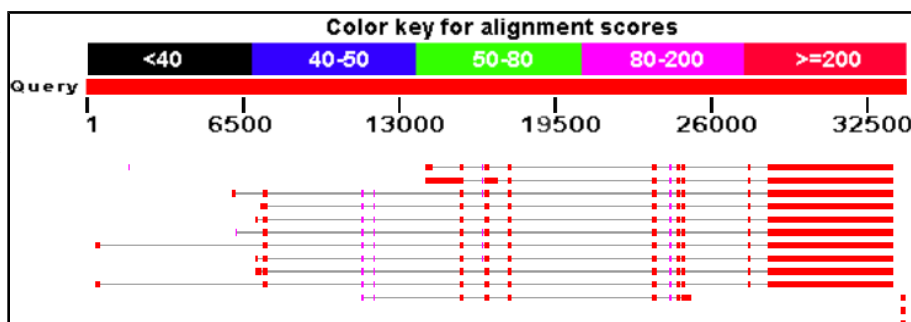
Why do you suppose that a few of the exons of the first 11 matches do not achieve the maximum score?

### Summary:

Each local region of significant alignment between a database entry and a query sequence is scored independently. The scoring method that governs the alignment score colour in this graphic, reflects both the quality of the match **and** its length. Unless a particular region is of sufficient length, it cannot achieve the **200 bit** threshold even if the alignment is perfect. Note that it is the shorter regions that fail to reach the **>=200** status. All of the illustrated local alignments associated with **PAX6** transcripts are essentially perfect.

### Full Answer:

In common with most database searching programs, **blast** compares query sequences with database entries using a local strategy. The overall evaluation of a particular query sequence is taken to be the highest local score.



Individual local matches are coloured according to individual quality. In this query, all true matches should be perfect, or very nearly so. Scores might therefore be expected to be maximal (**>=200**). However, they are not? Some only manage a score in the range **80-200**.

The score referenced for this purpose is the **bit score**. For a full, no holds barred definition of this score, try [here](#). I prefer this somewhat gentler version:

“The **bit score** gives an indication of how good the alignment is; the higher the score, the better the alignment. In general terms, this score is calculated from a formula that takes into account the alignment of similar or identical residues, as well as any gaps introduced to align the sequences. A key element in this calculation is the “substitution matrix”, which assigns a score for aligning any possible pair of residues. The **BLOSUM62** matrix is the default for most **BLAST** programs, the exceptions being **blastn** and **MegaBLAST** (programs that perform **nucleotide-nucleotide** comparisons and hence do not use protein-specific matrices). Bit scores are normalized, which means that the bit scores from different alignments can be compared, even if different scoring matrices have been used.”

Still too scary? The important things to note are that:

- These scores are based on a simple DNA scoring matrix (1 for a match, -2 for a mismatch by default for **megablast**), plus penalties for gaps. So scores will be limited by the length of the alignment, ignoring gaps.
- The scores reflect penalties for **indels** (insertions or deletions).
- The scores are normalised so that they do not depend on the chosen scoring matrix. This allows bits scores from searches using different scoring matrices to be compared.

This being so, **bit scores** will reflect the length of an alignment as well as its quality. If an alignment is very short, it might be perfect but still not achieve a very high value. **bit scores** are designed to reflect significance, not just local quality. A short perfect match clearly can be less significant than a longer less perfect match. That is what you see illustrated here.

You can see evidence of what is occurring in the alignments further down your results. Here is illustrated one of the **80-200** exons that occur in all transcripts at position 24,346<sup>9</sup>. The match is perfect, but the length of the exon is consistently just too short to get to the heady **>=200** level.

Range 7: 999 to 1086 <a href="#">GenBank</a> <a href="#">Graphics</a>					
Score	Expect	Identities	Gaps	Strand	
163 bits(88)	1e-36	88/88(100%)	0/88(0%)	Plus/Plus	
Query 24346	AGAGTTTGAGAGAACCCATTATCCAGATGTGTTTGGCCGAGAGAACTAGCAGCCAAAT	24405			
Sbjct 999	AGAGTTTGAGAGAACCCATTATCCAGATGTGTTTGGCCGAGAGAACTAGCAGCCAAAT	1058			
Query 24406	AGATCTACCTGAAGCAAGAATACAGGTA	24433			
Sbjct 1059	AGATCTACCTGAAGCAAGAATACAGGTA	1086			
Range 8: 1081 to 1234 <a href="#">GenBank</a> <a href="#">Graphics</a>					
Score	Expect	Identities	Gaps	Strand	
285 bits(154)	2e-73	154/154(100%)	0/154(0%)	Plus/Plus	
Query 24657	CAGGTATGGTTTTCTAATCGAAGGGCCAAATGGAGAAGAGAGAAAACTGAGGAATCAG	24716			
Sbjct 1081	CAGGTATGGTTTTCTAATCGAAGGGCCAAATGGAGAAGAGAGAAAACTGAGGAATCAG	1140			
Query 24717	AGAAGACAGGCCAGCAACACACCTAGTCATATTCCTATCAGCAGTAGTTTCAGCACCAGT	24776			
Sbjct 1141	AGAAGACAGGCCAGCAACACACCTAGTCATATTCCTATCAGCAGTAGTTTCAGCACCAGT	1200			
Query 24777	GTCTACCAACCAATTCCACAACCCACCAACCCGG	24810			
Sbjct 1201	GTCTACCAACCAATTCCACAACCCACCAACCCGG	1234			
Range 9: 1234 to 1350 <a href="#">GenBank</a> <a href="#">Graphics</a>					
Score	Expect	Identities	Gaps	Strand	
217 bits(117)	8e-53	117/117(100%)	0/117(0%)	Plus/Plus	
Query 24908	GTTTCCTCCTTCACATCTGGCTCCATGTTGGGCGAAGACAGACAGCCCTCACAACACC	24967			
Sbjct 1234	GTTTCCTCCTTCACATCTGGCTCCATGTTGGGCGAAGACAGACAGCCCTCACAACACC	1293			
Query 24968	TACAGCGCTCTGCCGCTATGCCAGCTTCACCATGGCAATAACCTGCCTATGCAA	25024			
Sbjct 1294	TACAGCGCTCTGCCGCTATGCCAGCTTCACCATGGCAATAACCTGCCTATGCAA	1350			

<sup>9</sup> In order to make this illustration, I needed set **Sort by:** (top of the alignments) to **Query start position**.

Note how imperfectly **blast** finds exon/intron boundaries. If the start of an intron happens to match the start of the next exon, **blast** will included the bases in two alignments<sup>10</sup>. It is not looking for exons and introns as was **spline**, it just mindlessly seeks matches.

Query	15745	CCCGAATTCTGCAG	15758
Sbjct	404	CCCGAATTCTGCAG	417

Range 3: 416 to 461 [GenBank](#) [Graphics](#) ▼ Next Match ▲ Previous Match ▲ First Match

Score	Expect	Identities	Gaps	Strand
86.1 bits(46)	2e-13	46/46(100%)	0/46(0%)	Plus/Plus

Query	16548	AGACCCATGCAGATGCAAAAGTCCAAGTCTGGACAATCAAAACGT	16593
Sbjct	416	AGACCCATGCAGATGCAAAAGTCCAAGTCTGGACAATCAAAACGT	461

Range 4: 460 to 677 [GenBank](#) [Graphics](#) ▼ Next Match ▲ Previous Match ▲ First Match

Score	Expect	Identities	Gaps	Strand
403 bits(218)	6e-109	218/218(100%)	0/218(0%)	Plus/Plus

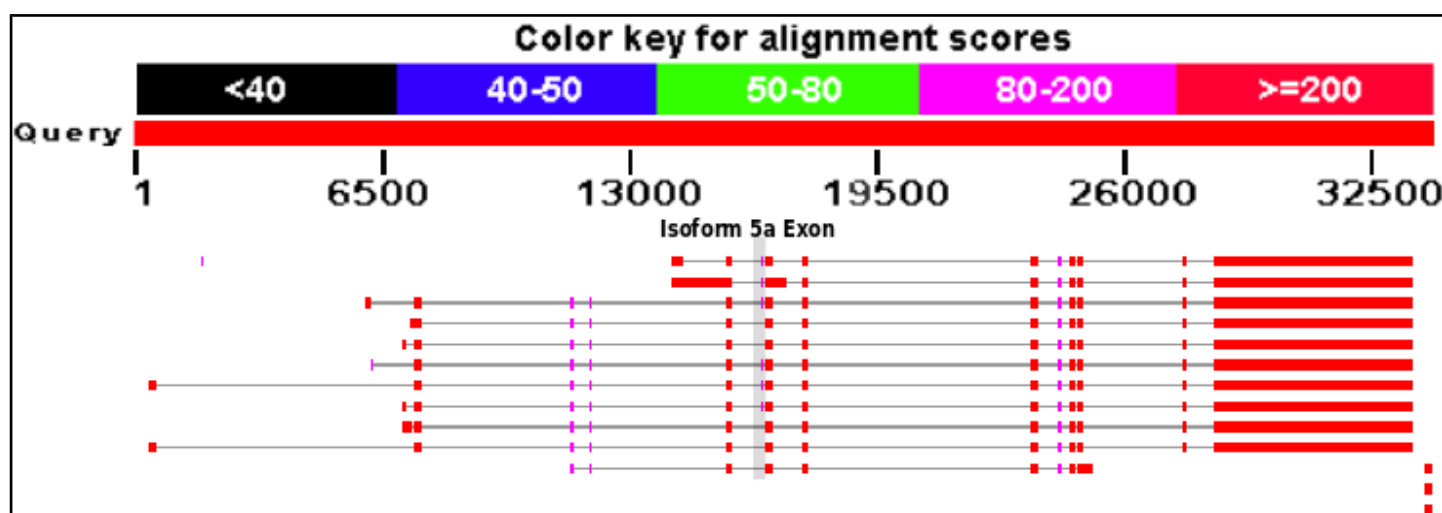
Query	16686	GTGTCCAACGGATGTGTGAGTAAATCTGGGCAGGTATTACGAGACTGGCTCCATCAGA	16745
Sbjct	460	GTGTCCAACGGATGTGTGAGTAAATCTGGGCAGGTATTACGAGACTGGCTCCATCAGA	519

For a further example, look at the exon that is found only in the **isoform 5a** transcripts. It is tiny (42 base pairs) and scores well below **>=200** even though it is a perfect match.

Note that the alignment is 46 base pairs long due to **blast** adding on two bases either side that are actually the highly conserved intron start and end base pairs. As you can see, these extra base pairs occur in the preceding and succeeding alignment also.

Explain why one exon in the reasonably consistent region, does not appear in all of the transcript matches?

Well I refer to the **isoform 5a** exon, of course. The tiny inconsistent one about 9 exons in from the right (when it exists). This will, clearly, only occur in **isoform 5a** transcripts.



<sup>10</sup> 6 base pairs (Sbjct: 1081-1086, CAGGTA) occur in both the first two matches illustrated. Just 1 base pair is shared between the 2<sup>nd</sup> and 3<sup>rd</sup> match (Sbjct: 1234, G).

Which of the Refseq PAX6 transcripts corresponds to isoform 5a?

### Summary:

As I am sure you are tired of noting by now, all the transcripts with the extra tiny exon around position **1,600** in the genomic sequence are **isoform 5a** transcripts. See the illustration for the previous answer.

### Full Answer:

The **isoform 5a** transcripts can be spotted most easily from the graphic. They are the ones with the extra small exon slightly to the left of middle (around base position **1,600**). For example, the **first**, **second** and **third blast** matches displayed. If you hover over all the full length matches with your mouse, you will see that they are **transcript variants 11, 10, 8, 7, 6, 5, 4, 2, 1, 3** and **9** (in the vertical order of the graphic).

Stated with the unequalled poetry of **RefSeq Accession Code** and lyrical **Title** Line, the list of those with the extra exon becomes:

<u>TITLE</u>	<u>ACCESSION CODE</u>
Homo sapiens paired box 6 (PAX6), transcript variant 11, mRNA	NM_001310161.1
Homo sapiens paired box 6 (PAX6), transcript variant 10, mRNA	NM_001310160.1
Homo sapiens paired box 6 (PAX6), transcript variant 8, mRNA	NM_001310158.1
Homo sapiens paired box 6 (PAX6), transcript variant 5, mRNA	NM_001258463.1
Homo sapiens paired box 6 (PAX6), transcript variant 4, mRNA	NM_001258462.1
Homo sapiens paired box 6 (PAX6), transcript variant 2, mRNA	NM_001604.5

Yes well, that was fun? The message of the question was to ensure you could see how to spot the **isoform 5a** transcripts (again!), not to list them! But, never mind, doing so was in fine tune with the ennui of the moment.



**Additional Meanderings:**

That really should not detain anyone but me? They belong after the consideration of masking the run of As for the **megablast** you ran. I just enjoyed this detour, so I keep it somewhere low profile. The whole journey leads nowhere of any note, so, should you decide to read, expect little!

The **500** base pairs of 3' flanking sequence added on to the **Ensembl** sequence for “good measure”, is not part of the alignment (as would be expected). This can be seen easily if you look at the end of the alignment illustrated above, which is the alignment of the last exon of a transcript.

The entire length of this transcript is **6,732** base pairs.

Homo sapiens paired box 6 (PAX6), transcript variant 11, mRNA  
Sequence ID: ref|NM\_001310161.1| Length: 6732 Number of Matches: 11

The entire length of the genomic query sequence is **34,170** base pairs.

<b>Query ID</b>	lcl Query_229907
<b>Description</b>	11 dna:chromosome chromosome:GRCh38:11:31784292:31818461:-1
<b>Molecule type</b>	nucleic acid
<b>Query Length</b>	34170

The alignment ends at position position **6,729** of the mRNA (**3** from the end) and position **33,670** of the genomic sequence (exactly **500** base pairs from the end).

Query	33601	ATTTGACATCCTGGCAAATCACTGTCATTGATTCAATTATTCTAATTCTGAATAAAAGCT	33660
Sbjct	6660	ATTTGACATCCTGGCAAATCACTGTCATTGATTCAATTATTCTAATTCTGAATAAAAGCT	6719
Query	33661	GTATACAGTA	33670
Sbjct	6720	GTATACAGTA	6729

The **3** missing base pairs of the mRNA are all As, due to **polyadenylation**. Position **6,729** being recorded as a **polyA** site by **RefSeq**. A very short **polyA** tail surely? But there is no telling what stage of the mRNA is recorded on **RefSeq**. **Wikipedia** says:

<b>polyA site</b>	2495
/gene=	"PAX6"
/gene_synonym=	"AN; AN2; D11S812E; FVH1; MGDA; WAGR"
6541	aaaaaaatag aataagaac ctgattttta gtactaatga aatagcgggt gacaaaatag
6601	ttgtcttttt gatattgac acaaaaata aactggtagt gacaggatat gatggagaga
6661	tttgacatcc tggcaaatca ctgtcattga ttcaattatt ctaattctga ataaaagctg
6721	tatacagta aa

“The tail is shortened over time, and, when it is short enough, the mRNA is enzymatically degraded.”

Of course, the neatness of this observation reflects less some profound biological truth than that this mRNA just happens to be one that extends furthest to the right in the genome, and there is no chance match between the **polyA** tail and the extra **500** bases of genomic sequence you added on when extracting it from **Ensembl**.

The journey was fun even though the destination was dubious. Much the way of a considerable portion of life in general one might reflect?

What are the **9** stronger matches around base position **16,000**?

Matches between the regions of the **PAX6** genomic region encoding the **PAX6 Paired Box** domain and **SwissProt** protein sequences representing human proteins including a **Paired Box** domain.

Why would you expect exactly **9** matches around this point?

Because that is how many human proteins including a **Paired Box** domain are suggested to exist according to **Interpro** (as shown in a previous Practical). There is **PAX6** plus its **8** paralogues, imaginatively all named:

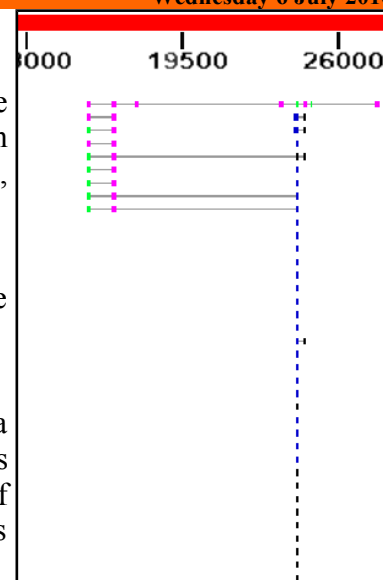
**PAX1, PAX2, PAX3, PAX4, PAX5, PAX6, PAX7, PAX8 & PAX9**

## What do you make of the plethora of matches around 24,000?

These are matches between the regions of the **PAX6** genomic region encoding the **PAX6 Homeobox** domain and **SwissProt** protein sequences representing human proteins including a **Homeobox** domain. As you discovered earlier from **Interpro**, there are lots of such proteins.

The thin line joining features implies that those features relate to the same database entry.

Notice that 4 of the 9 proteins including a **Paired box** domain also include a **Homeobox** domain. The remaining 5 do not. This implies that 4 of the 9 proteins corresponding to the hits detected here have a **Paired box** domain near the start of the protein and a **Homeobox** domain further along. This is exactly as was suggested by the **SMART** annotation you examined earlier.

Why do you suppose the **Paired box** matches precede the **Homeobox** matches?

Because they score more highly and so, in the opinion of **blast**, are more worthy. Primarily, they score more highly because they are longer. The list is ranked by **E Value**. Good matches with long sequence are less likely to occur by chance than equally good matches with shorter sequences.

Possibly a more interesting question" might have been: "Why are not all the hits which include both domains at the top of the list?". Surely they should be, as they match over a longer proportion of the query sequence and so must, in general at least, be of the greatest significance.

They do not always come at the top of the list because **blast** scores each matching region individually and uses the ranking scores associated with the single region with the highest **E Value** to evaluate the similarity of the entire database entry with the query. This has to be a dubious practice surely? But, it appears to work, so why complain.

Description	Max score	Total score	Query cover	E value	Ident	Accession
RecName: Full=Paired box protein Pax-6; AltName: Full=Aniridia type II protein; AltName: Full=Oculorhombin	160	767	3%	3e-41	97%	P26367.2

To justify this last assertion, Look at your top hit.

**E Val = 3e-41, Max score = 160, Total score 767** associated with the whole of **P26367.2**

Now look at the first few individual regional alignments for this hit.

RecName: Full=Paired box protein Pax-6; AltName: Full=Aniridia type II protein; AltName: Full=Oculorhombin						
Sequence ID: sp P26367.2 PAX6_HUMAN Length: 422 Number of Matches: 8						
Range 1: 46 to 123 <a href="#">GenPept</a> <a href="#">Graphics</a> <a href="#">Next Match</a> <a href="#">Previous Match</a>						
Score	Expect	Method	Identities	Positives	Gaps	Frame
160 bits(406)	3e-41	Compositional matrix adjust.	76/78(97%)	78/78(100%)	0/78(0%)	+3
Query	16860	MOVSNQCVSKILGRYYETGSI	PRAIGGSKPRVATPEVVS	KIAQYKRECP	SIFAW	EIRDR
Sbjct	46	+QVSNQCVSKILGRYYETGSI	PRAIGGSKPRVATPEVVS	KIAQYKRECP	SIFAW	EIRDR
Query	16860	LLSEGVCTNDNIPSVSSL	16913			
Sbjct	106	LLSEGVCTNDNIPSVSSI	123			
Range 2: 254 to 305 <a href="#">GenPept</a> <a href="#">Graphics</a> <a href="#">Next Match</a> <a href="#">Previous Match</a> <a href="#">First Match</a>						
Score	Expect	Method	Identities	Positives	Gaps	Frame
81.3 bits(199)	5e-29	Compositional matrix adjust.	51/52(98%)	51/52(98%)	0/52(0%)	+3
Query	24654	FQVWFSNRRRAKWRREKLR	NORRQASNT	PSHIPIS	SFSTSVYQ	PIQPTTP
Sbjct	254	IQVWFSNRRRAKWRREKLR	NORRQASNT	PSHIPIS	SFSTSVYQ	PIQPTTP
Range 3: 312 to 344 <a href="#">GenPept</a> <a href="#">Graphics</a> <a href="#">Next Match</a> <a href="#">Previous Match</a> <a href="#">First Match</a>						
Score	Expect	Method	Identities	Positives	Gaps	Frame
70.5 bits(171)	5e-29	Compositional matrix adjust.	33/33(100%)	33/33(100%)	0/33(0%)	+2
Query	24926	GSMLGRD	TALNTY	SALPP	MPSFT	MANNLPMQ
Sbjct	312	GSMLGRD	TALNTY	SALPP	MPSFT	MANNLPMQ

As you can see, the **E Value** and **Max score** values used to evaluate the whole protein were computed from just the best (ranked by **E Value**) local alignment! Crude, but never mind.

The **Total score** for the entire protein is the sum (rounded up to the nearest integer) of all the bit scores for all 8 local alignments computed for this protein (I suggest you just trust me on this assertion).

11 That I did not ask, because I only just thought of it.

How do you suppose the **Max matches in a query range** parameter might be of value if this order was reversed?

If **Paired boxes** had been more prolific, then the number of **Paired box** matches might have filled the **blast** hit list before the highest scoring **Homeo box** hit was registered.

If **Homeo boxes** were longer, and so justified a better **E value**, then the number of **Homeo box** matches might have filled the **blast** hit list before the highest scoring **Paired box** hit was registered.

Either of these situations would be very unfortunate, but easily avoided by setting the **Max matches in a query range** parameter to something sensible (**50** say). This would ensure that only the top **50** items in the **blast** hit list would be dominated by the strongest hit.

**UNFORTUNATELY** ... although that is the intention of this parameter, it currently simply will not work, except in very particular circumstances, because of the way it is implemented. This is a great pity, because it is a very good idea, in principle.

I will spare you the details until the energetic debate I am having with the **NCBI** people has come to a satisfactory (or more probably, otherwise) conclusion.

How does this “non-informative” region match expectations suggested by **SMART** and the **Feature table** of **UniprotKB** for **PAX6\_HUMAN**?

**blast** identifies two non-informative regions. I only discussed the prettiest one above. The region discussed is comprised largely of **Serines**, **Prolines**, **Threonines** & **Isoleucines** the **15** residues between **294-308**.

Score	Expect	Method	Identities	Positives	Gaps	Frame
81.3 bits(199)	5e-29	Compositional matrix adjust.	51/52(98%)	51/52(98%)	0/52(0%)	+3
Query 24654	FQWFSNRRRAKWRREEKLRNQRROASNT	tpshipissstst	VYQPIQPPTTP	24809		
Sbjct 254	IQWFSNRRRAKWRREEKLRNQRROASNT	PSHIPISSSTST	VYQPIQPPTTP	305		

The second (to be found much further down your **blast Alignments** output) is comprised entirely of **Arginines**, **Leucines** and **Lysines** and **Glutamines**, the **10** residues between **203 - 212**.

Score	Expect	Method	Identities	Positives	Gaps	Frame
85.9 bits(211)	3e-16	Compositional matrix adjust.	56/66(85%)	58/66(87%)	5/66(7%)	+3
Query 23649	YHPILFVP-----	DGCQQQEGGGENTNSISSNGEDSDEAQMRL	gkrkkr	NRTSFTQEQ	23813	
Sbjct 162	++P VP	DGCQQQEGGGENTNSISSNGEDSDEAQMRL	QLKRLQRNRTSFTQEQ		221	
Query 23814	IEALEK	23831				
Sbjct 222	IEALEK	227				

**Uniprotkb** also suggests there are two **compositionally biased** regions.

Compositional bias	131 – 209	79	Gln/Gly-rich
Compositional bias	279 – 422	144	Pro/Ser/Thr-rich

Well, hardly an exact match, but there is approximate agreement? One would certainly suppose that **blast** is only willing to mask fairly severe cases of **compositional bias**. It is also probable that **blast** has a rather more mechanistic (i.e. non-biological) interpretation of what **computational bias** is?

**SMART** also predicts the more obvious region of **computational bias**, rather more generally:

“An octapeptide and/or a homeodomain can occur C-terminal to the paired domain, as well as a Pro-Ser-Thr-rich C-terminus”

Not important points in themselves of course, the real message of the exercise is that you can discover so much by either:

Looking things up in databases

or:

Using the simple analytical software tools yourself.

DPJ – 2016.07.06