Reciprocal Best Match Problem

Matthew Preston

June 2017

Version 1.0 (Public Beta)

Acronyms

A adenine. 6, 7, *Glossary:* adenine

BLAST Basic Local Alignment Search Tool. 13

C cytosine. 6, 7, *Glossary:* cytosine

DNA Deoxyribonucleic acid. 6, *Glossary:* DNA

G guanine. 6, 7, *Glossary:* guanine

HSP High-scoring segment pair. 15, 18, *Glossary:* high-scoring segment pair

mRNA Messenger ribonucleic acid. 7, *Glossary:* mRNA

NCBI National Center for Biotechnology Information. 13

RNA Ribonucleic acid. 6, *Glossary:* RNA

T Thymidine. 6, *Glossary:* thymine

TLDR Too long, didn't read. 5

U Uracil. 7, *Glossary:* uracil

Glossary

adenine One of the four nucleotides used in DNA and RNA, hydrogen bonds with guanine. 6

amino acid The building blocks for proteins, typically 20 different types are used. 7

annotate To assign a name to a sequence with an unknown identity. 13

central dogma DNA makes RNA makes protein. 8

codon A triplet of nucleotides used to encode for a certain amino acid, found in mRNA when being translated to make proteins. 7

contig A contiguous or continuous stretch of RNA created by overlapping shorter pieces of RNA known as reads, the output of *de novo* assemblers. 11

cross entropy Used in information theory to describe how well one probability distribution can explain data drawn from another probability distribution. 25

cytosine One of the four nucleotides used in DNA and RNA, hydrogen bonds with thymidine in DNA, uracil in RNA. 6

De Bruijn graph A mathematical graph construct used to assemble short RNAs or reads into larger RNAs known as contigs. 12

de novo Begin again, from nothing. 9, 11, 15, 16

differentiation The specialization of tissue or cell types, i.e. a stem cell differentiating into a neural cell. 12

DNA The priciple molecule responsible for inheritance, harbors genes used for creating proteins. 6

E-value The expected number of HSPs with a score greater than a particular score, or the expected number of hits when searching the database of the same length, found in the statistical output of BLAST. 14

express The act of a gene being active and being used to make proteins. 12

gap A space in an alignment caused by either a deletion or an insertion. 20

gene The fundamental inheritance unit, comprised of DNA and is responsible for making one (or more) protein(s). 6

guanine One of the four nucleotides used in DNA and RNA, hydrogen bonds with cytosine. 6

high-scoring segment pair A pairwise alignment with a large number of exact or very similar matches, used for comparing two sequences together. 15

model organism An organism that can reproduce quickly, has a short lifespan, high fertility, and other characteristics that make it desirable to study. 15

mRNA A specific class of RNA, used solely to be translated to produce proteins. 7

mutation A change in a sequence that may give rise to a new product that is formed. 8

next generation sequencing A novel technique that allows for quick and cost effective sequencing of DNA and RNA. 10

nucleotide The building block of DNA and RNA. 6

paralog A duplicated gene. 17

phylogenetics The study of phylogenies, which represent the predicted evolutionary relationship between different species. 8

plasticity The ability for an organism to adapt to the local environment. 12

protein A type of molecule produced by cells that can have diverse functions, made of amino acids. 6

read A short stretch of DNA/RNA that was derived from full length DNA/RNA. 11

reference The set of known sequences used to assign names to unknown sequences. 11

RNA A molecule that is transcribed from DNA, used in translation of proteins (mRNA), immune functions (miRNA, siRNA), or binding to molecular machinery (lncRNA). 6

score A number that describes how good an alignment is, more same or similar matches leads to a better score and dissimilar or gaps in the alignment leads to a worse score. 14

thymine One of the four nucleotides used in DNA, hydrogen bonds with adenine. 6

transcribe To create RNA from DNA. 7

translate To create protein from RNA. 7

uracil One of the four nucleotides used in RNA, hydrogen bonds with adenine. 7

Chapter 1

Introduction

TLDR: Matt has a problem and needs your advice.

As you may know, I am currently working at a research lab at the University of Toronto, where I employ my skills as a computer scientist to assist others. It is not imperative to note that the only computer science course I have taken was with Mr. Smith at Mentor College but such details do not matter. It also does not help that I currently study the field of biology and sometimes require more sophisticated mathematics to solve my problems. Thus I have taken the initiative to learn some math topics on my own such as multivariable calculus, linear algebra, and statistics. However, this does not mean that I am formally trained in it; just enough to get by common problems. But currently, I do not have a trivial conundrum. This is where you, the reader, come in. I have sent this document to you as a cry of help as I am now venturing uncharted territory and would like your help.

Go grab a drink of something. It's gonna be a long ride.

Chapter 2

Biology 101

To present the problem, you must learn some of the basics of biology, in particular: the biological sequences one may encounter, how speciation plays a role in the evolution of genes, and some bioinformatics along the way.

2.1 The Central Dogma

Biological sequences are central towards living and reproduction. Thus, I will teach you the 3 main sequences: double stranded **DNA** (the library / source code), the single stranded **RNA** (the temporary messenger / compiled code), and the **protein** (the functional molecule / executable).

Your genome is comprised of **nucleotides**, the building blocks of DNA (and RNA). These can be differentiated from one another by being covalently bonded to a certain nitrogenous base. For DNA, these bases are **adenine (A)**, **thymidine (T)**, **guanine (G)**, and **cytosine (C)**. These nucleotide molecules form a polymer, or a long sequence such that information can be encoded in them. You can think of this as a contiguous block of memory where the bits (tetrits?) have 4 states. In terms of humans, there are 23 pairs of these nucleotide polymers, a.k.a. chromosomes, stored in the nucleus of each cell of your body[1], excluding your germ cells (eggs/sperm) which are unpaired.

Within these chromosomes, there are short stretches of DNA called **gene** that encode a specific protein. Relative to chromosomal length, which can be millions of nucleotides long. In humans, we have approximately 20,000 protein encoding genes with relatively distinct functions. However, some genes have very high sequence similarity, implying that a phenomenon known as gene duplication took place. Essentially, during cellular replication (when one cell becomes two cells), each chromosome must be copied. Errors may arise and can lead to duplications or deletions of genetic material albeit rarely. If any of these typographical accidents occurs in one's germ cells, that error will propagate onto their offspring.

[1] Erythrocytes (red blood cells) and thrombocytes (platelets) also lack nuclear DNA

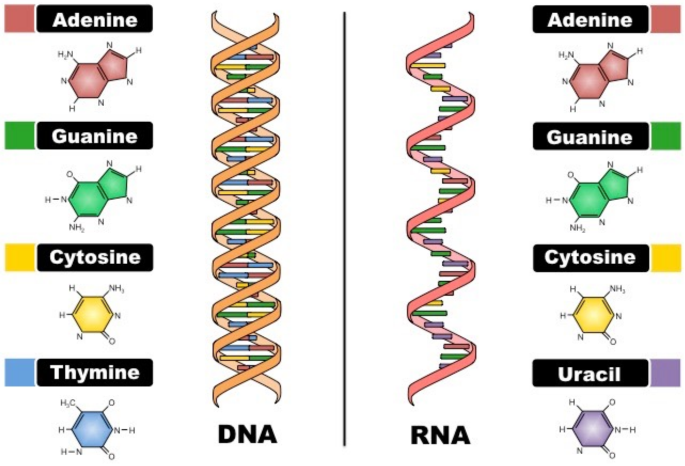


Figure 2.1: Double Stranded DNA and Single Stranded RNA

The transcriptome, or the set of all RNA molecules, is a transient group. Like DNA, they are polymers of 4 types nucleotides: A, U, C, G; where thymine is substituted for **uracil (U)**. There are copious classes of RNA but the major type is **mRNA**. The role of mRNA is to serve as a temporary messenger for DNA by harbouring physical instructions intended to produce a particular protein. More specifically, a gene from a chromosome is **transcribed** into mRNA where DNA's A, T, C, G's are copied into RNA's A, U, C, G's while maintaining the information encrypted in the particular sequence of nucleotides. This mRNA is shuttled out of the nucleus and into the main compartment of the cell: the cytosol. This is where the mRNA is translated into a functional protein, which can happen multiple times until it is degraded.

The proteome, as you may have guessed, must then be the set of all proteins. These molecules play a staggering number of roles in and outside of cells. Such duties include: cellular maintenance, energy production, motility, immune responses, garbage disposal, etc. Proteins are polymers of **amino acids** which are either created by the cell or absorbed via diet. Typically, cells employ 20 amino acids to encode their proteins, each with distinct chemical and physical characteristics. In order to manufacture proteins, the mRNA must be **translated**. To use more programming innuendo, 3 tetrits encode a byte (tetryte?), i.e. ACU, UGA, CCC are these tetrytes. Biologists call these tetrytes (I made that word up by the way) **codons**. Now if your spidey senses are tingling, if there are 4 different nucleotides to choose from and these codons are of length 3, then there should be $3^4 = 64$ permutations. However as we said previously, there are only 20 amino acids. The reason for this discrepancy is due to redundancy in the code. For example, the amino acid leucine (L) can be encoded by CUU, CUC, CUA, CUG, UUA, and UUG.

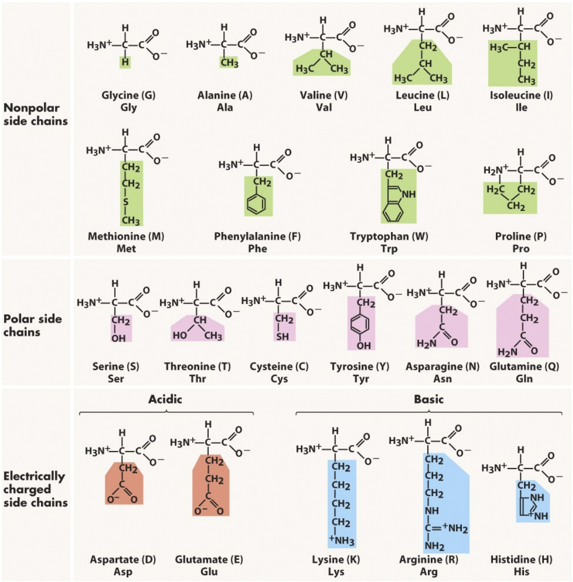
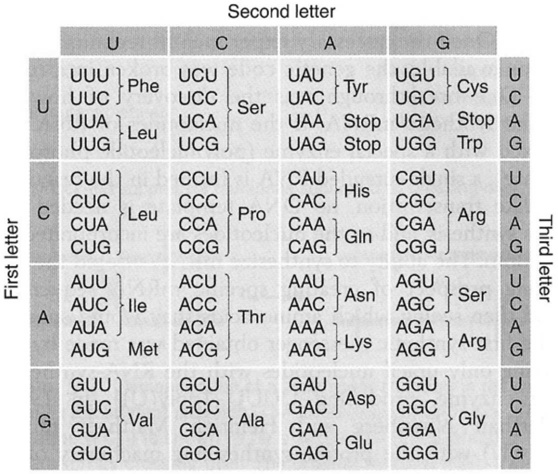


Figure 3-5 Biological Science, 2/e

© 2005 Pearson Prentice Hall, Inc.

Figure 2.2: The 20 Standard Amino Acids



	Second letter								
First letter	U		C		A		G		Third letter
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu	UCA		UAA	Stop	UGA	Stop	A
	UUG		UCG		UAG	Stop	UGG	Trp	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lys	AGA	Arg	A
	AUG	Met	ACG		AAG		AGG		G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu	GGA		A
	GUG		GCG		GAG		GGG		G

Figure 2.3: The Standard Codon Table (for Cryptologists in Biology)

In conclusion, DNA, RNA, and proteins are sequences comprised of their own alphabet as to perform their own respective function. There is a famous concept in biology called **the central dogma** coined by Francis Crick, one of the scientists who helped discover DNA's double stranded nature. The central dogma states: DNA makes RNA makes protein; that is the real takeaway of this section.

2.2 Phylogenies

Now we take about 15 steps back and look at the bigger picture. Different species have different genomes, which is the main reason for their differences in differentiation, differing cell types, and divergent body structures[2]. This is why plants possess the biochemical propensity to form sugars from carbon dioxide due to specific plant proteins unlike us humans who must expel it since man is wasteful. But what about species that are closely related, such as apes and humans? Our genomes are quite similar to theirs as well as sharing many of the same genes. This notion of likeness between species is described in the field of **phylogenetics** which terms this idea as evolutionary distance.

Over time, **mutations** or changes in the genome can accumulate. These mutations are caused by mundane activites such as errors in cellular replica-

[2] As well as the relative expression levels and their location of expression of these genes which truly determine these above factors, but I ran out of words starting with 'd'

tion, standing in a X-ray machine for prolonged periods, quaffing quantities of benzene (a known carcinogen so can't quite recommend for consumption), or walking/waltzing outside on a sunny day (UV light can cause 50,000 errors in replication per cell, but most of these get fixed[3]). We saw previously about gene duplication, which creates a new copy of the gene to play with, or gene deletion (but typically not). Finally, genes can be created ***de novo*** or "from nothing" as I like to think of it, which is through random gene mutations that happen to make a new gene. However, this last method is an extremely seldom occurance.

For this example, let us examine the following phylogenetic tree which details the genetic relation for a gene family called Hedgehog.

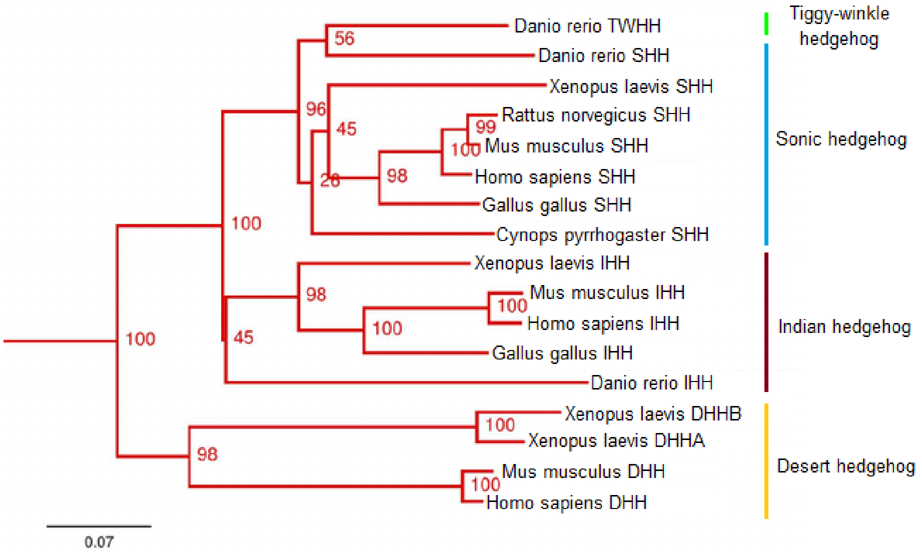


Figure 2.4: Hedgehog Genes but not the Cute Kind

There are many members in this gene family, all professionally named. The terminal nodes or leaves displays a species followed by an abbreviated gene name. For example, *Danio rerio* is the official name for the zebrafish, *Gallus gallus* is the chicken, and *Homo sapiens* is the most dangerous species on Earth.

The internal nodes or branch points represent one of two scenarios. If the eminating branches contain genes of the same species, the node signifies a gene duplication event which we talked about in the last section. For the other case where the species are different, this indicates a speciation event, i.e. some ancestral organism gave birth to two individuals where one of them became the parent of one species and one for the other. For an example of the former, we see the zebrafish harbouring a duplication, which then led to one copy becoming the Sonic hedgehog (SHH) gene while the other became the Tiggy-winkle hedgehog

[3] Source: Matt's 2016 BCH311H1S class that was at freaking 10AM on a Friday, too early man

(TWHH) gene. For the latter, inspect the node between *Rattus norvegicus*, the rat, and *Mus musculus*, the mouse, which then bifurcates to allow for each of the species to have their own copy of SHH.

On the horizontal axis is some denotation of evolutionary time, i.e. the shorter a horizontal branch, the more related in sequence similarity to some ancestral gene, which can be thought of as the root for our purposes. You'll notice that the zebrafish has the Indian hedgehog (IHH) which was duplicated long before the era of the SHH and TWHH genes, implying IHH tanked a bunch mutations over time, leading to a long branch length. There is no real vertical axis measurement. The order of how these genes are listed is only tailored by the biologist to compose a presentable figure.

One last detail to note for those curious. You'll find numbers at these internal nodes. These are statistical measures of confidence for the predicted partitions. You see, the only real data biologists have to work with are the raw DNA sequences for genes. They assume some model for how genes mutate over time and employ various maths which I'll leave out partially (Bayesian statistics + hidden Monte Carlo Markov chains = kill me dude) which results in a hypothetical tree that portrays the relationship between genes within and across species.

The takeaway is that genes are passed on through generation to generation. This is seen during the formation of novel species which harbour these genes. Within a species, a gene may also duplicate and diverge to become two different genes. Also that biologists are great at naming genes.

2.3 Modern Sequencing

Cells are small, nearly insignificant living beings. So how do we extract the sequences of DNA, RNA, and protein if these are molecules on the scale of femtometers? Well we don't have to collect all three as if we know the DNA sequence, we can deduce the mRNA and protein sequences through transcription and translation respectively. But since DNA and RNA are relatively synonmyous in terms of information (transcription of DNA to RNA is not lossy[4]), RNA can also be captured to determine both the original gene and protein. In addition, RNA is found in higher concentrations than DNA as multiple copies of mRNA are created from one DNA template. Thus it is desirable to target RNA for high throughput or highly parallelizable applications.

There are 3 main **next generation sequencing** techniques: Illumina, Roche 454, and Ion torrent. Their subtleties are unimportant, except for what they share in common. RNA molecules can be quite large, so they have to be sheared down to 100-1000 nucleotides long in order for these sequencing procedures to elucidate the exact arrangement of As, Ts, Cs, and Gs. An overview of the procedure: the biologist gets their tissue / cell culture of interest, extracts the RNA, turns the RNA into cDNA (DNA has higher stability

[4] Excluding the splicing of introns, which are internal segments of RNA which are excised out before translation

than RNA, important when you're shipping your samples to a company thousands of kilometers away), adds some barcode tags to them (specified by the manufacturer's protocol), and send them off to a company to sequence them, as well as some money of course. They will send you a huge electronic text file, on the order of gigabytes, consisting of short **reads**, which represent the sheared RNA sequences. You then run some assembly programs such as BWA (`http://bio-bwa.sourceforge.net/`), Trans-Abyss (`http://www.bcgsc.ca/platform/bioinfo/software/trans-abyss`), and Trinity (`https://trinityrnaseq.github.io/`) to name a few, which will try output your original RNA sequences you put in (formally, the output sequences are called **contigs**, which is short for contiguous sequences which could be the reconstructed RNA sequences or just misassembled junk).

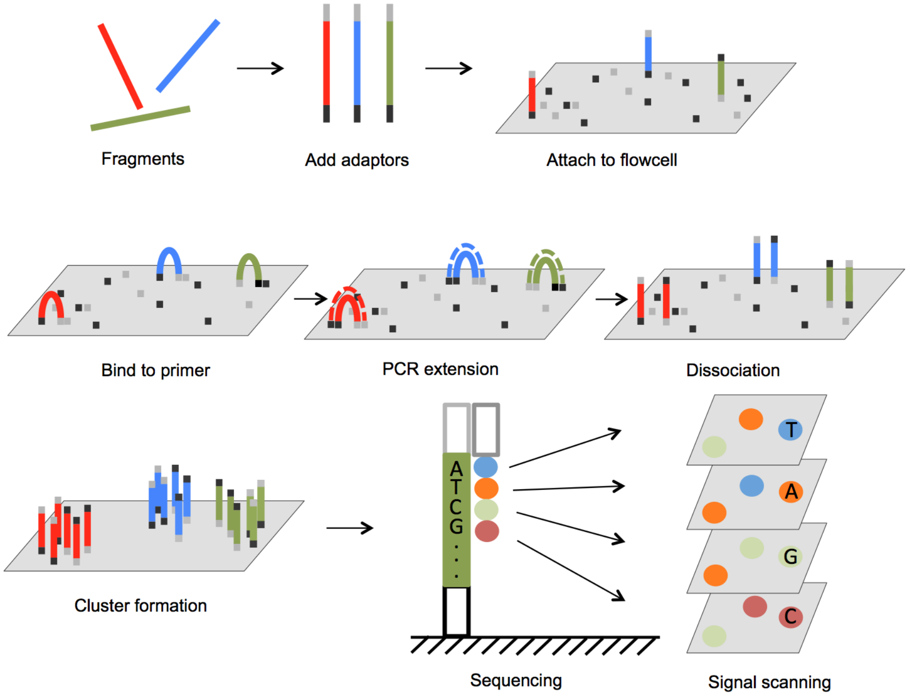


Figure 2.5: All Hail Illumina Sequencing

Actually let me clear up that important false statement I wrote in that last paragraph. There are two flavours of assembly programs: guided assembly and *de novo* assembly.

Guided assemblers make use of a **reference** genome / transcriptome, whereby scientists have already figured out many to most to all of the sequences. Thus it is a *relatively* simple matter of reassembling these reads back into their original sequence using the reference as a guide. "Why do this?", you may ask. If we already have the sequences, why are we reinventing the wheel by extracting more RNA to do it again? Well, different tissues may have different levels of the RNA of the genes created or **expressed**, and these levels are useful when considering scenarios such as **differentiation** (changing of one type of tissue for another; important during development), tumorigenesis (tumours produce odd levels of RNAs / proteins), and **plasticity** (the ability of an organism to locally adapt to its environment, i.e. tanning of your skin protects from UV light).

De novo assemblers do not have the luxury of a reference dataset. They instead employ **De Bruijn graphs**, to reassemble the reads back into their full, intact form. In a layman's example, imagine being an FBI agent searching through Neilbob's shredded NSFW documents. You would take a small bit of fragmented paper (like a read) and compare the ends of the text of that parchment against other remnants until you find those that would then create a coherent sentence when merged (the full RNA). Unfortunately, since Neilbob's material is so dank and voluminous in its girth, you may accidentally create some pseudo-sentences, which actually don't exist. They are more like of chimeras, merging two unrelated bits together. This will be a source of error in my grand problem as I will later present. Because we didn't have a reference, we will not be able to label which reassembled sequence is which (i.e. they will be left unnamed). This is where the next tool comes in: BLAST.

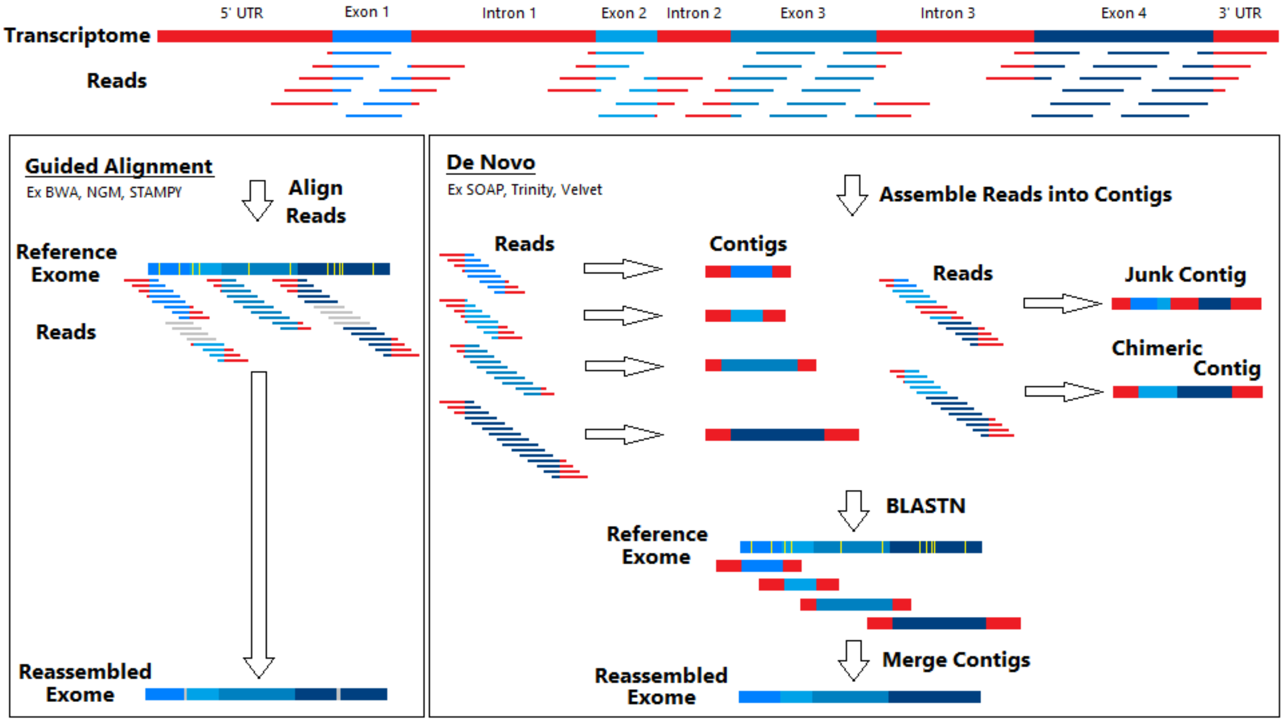


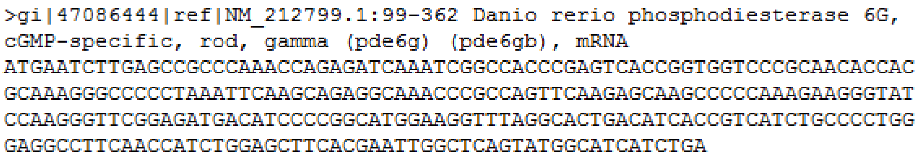
Figure 2.6: Guided Versus De Novo Assemblers

2.4 BLAST

The **NCBI** doesn't even make sense as a title, since biotechnology is a lame adjective for the noun information. Being U.S. government funded, it is an invaluable resource for bioinformaticians everywhere. They have developed an utility known as the **BLAST** which makes use of NCBI's grandiose sequence database. The user passes in a query sequence, typically of unknown name, and "BLAST"s it against the database, where this program will return all matching or similar sequences to it. Essentially, BLAST assigns a name or **annotates** the sequence. To determine which is the best match, BLAST makes use of statistical data, such as p-values which will be explored more in depth later.

My pipeline that I helped program (`https://www.ncbi.nlm.nih.gov/pubmed/28137744`) utilizes this step as to annotate sequences reassembled using de novo assemblers. It is an indispencible tool for me and bioinformaticians alike. So without further ado, let's see what we can do!

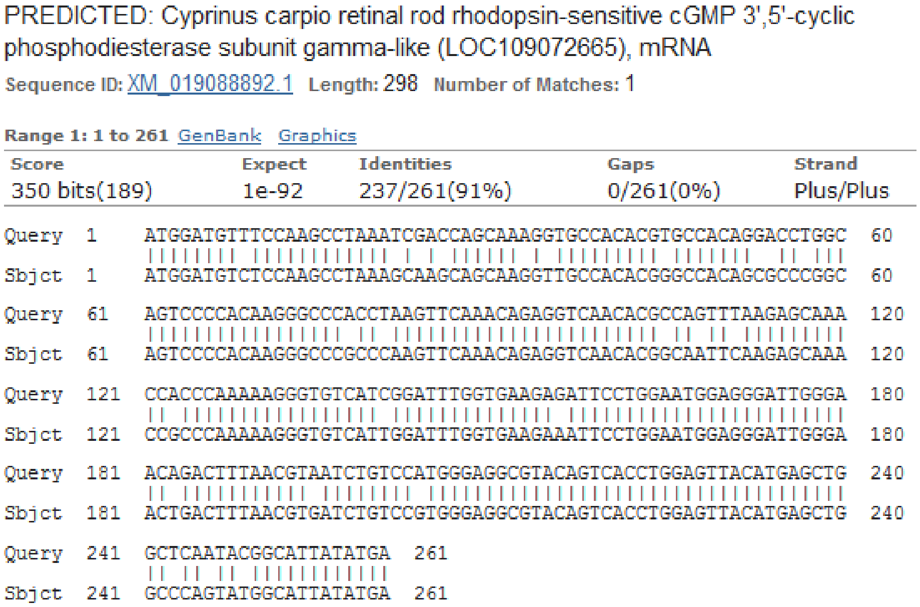
2.4.1 Go BLAST yourself



```
>gi|47086444|ref|NM_212799.1:99-362 Danio rerio phosphodiesterase 6G,
cGMP-specific, rod, gamma (pde6g) (pde6gb), mRNA
ATGAATCTTGAGCCGCCCAAACCAGAGATCAAATCGGCCACCCGAGTCACCGGTGGTCCCGCAACACCAC
GCAAAGGGCCCCCTAAATTCAAGCAGAGGCAAACCCGCCAGTTCAAGAGCAAGCCCCCAAAGAAGGGTAT
CCAAGGGTTCGGAGATGACATCCCCGGCATGGAAGGTTTAGGCACTGACATCACCGTCATCTGCCCCTGG
GAGGCCTTCAACCATCTGGAGCTTCACGAATTGGCTCAGTATGGCATCATCTGA
```

Figure 2.7: Zebrafish PDE6G gene

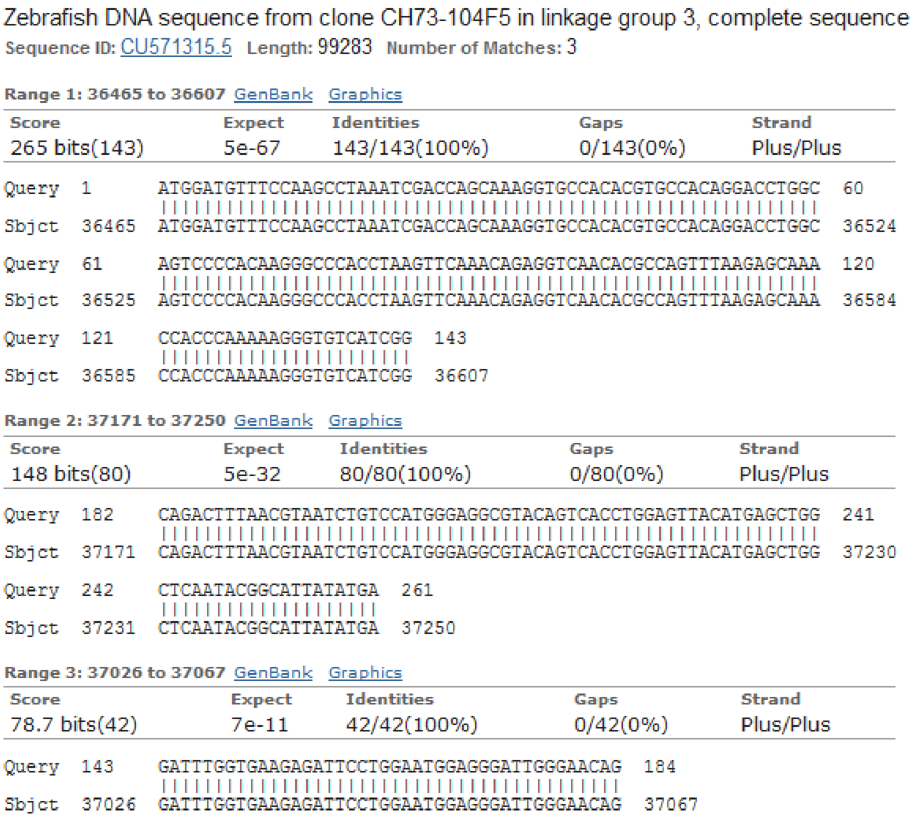
Let's see an example. I will use a Zebrafish phosphodiesterase 6 gamma (PDE6G) gene as my query sequence as shown above. Then using the online BLAST tool (`https://blast.ncbi.nlm.nih.gov/Blast.cgi`), I find all known reference sequences that align to this sequence which this program nicely spits out to me.



```
PREDICTED: Cyprinus carpio retinal rod rhodopsin-sensitive cGMP 3',5'-cyclic
phosphodiesterase subunit gamma-like (LOC109072665), mRNA
Sequence ID: XM_019088892.1  Length: 298  Number of Matches: 1

Range 1: 1 to 261  GenBank  Graphics
Score            Expect   Identities      Gaps         Strand
350 bits(189)    1e-92    237/261(91%)    0/261(0%)    Plus/Plus

Query  1    ATGGATGTTTCCAAGCCTAAATCGACCAGCAAAGGTGCCACACGTGCCACAGGACCTGGC  60
            |||||||| |||||||||||| | | |||||| | ||||||||| |||||||  || |||
Sbjct  1    ATGGATGTCTCCAAGCCTAAAGCAAGCAGCAAGGTTGCCACACGGGCCACAGCGCCCGGC  60

Query  61   AGTCCCCACAAGGGCCCACCTAAGTTCAAACAGAGGTCAACACGCCAGTTTAAGAGCAAA  120
            ||||||||||||||||| || ||||||||||||||||||||||| || || |||||||||
Sbjct  61   AGTCCCCACAAGGGCCCGCCCAAGTTCAAACAGAGGTCAACACGGCAATTCAAGAGCAAA  120

Query  121  CCACCCAAAAAGGGTGTCATCGGATTTGGTGAAGAGATTCCTGGAATGGAGGGATTGGGA  180
            || ||||||||||||||||| ||||||||||||||| |||||||||||||||||||||||
Sbjct  121  CCGCCCAAAAAGGGTGTCATTGGATTTGGTGAAGAAATTCCTGGAATGGAGGGATTGGGA  180

Query  181  ACAGACTTTAACGTAATCTGTCCATGGGAGGCGTACAGTCACCTGGAGTTACATGAGCTG  240
            || ||||||||||| |||||||| ||||||||||||||||||||||||||||||||||||
Sbjct  181  ACTGACTTTAACGTGATCTGTCCGTGGGAGGCGTACAGTCACCTGGAGTTACATGAGCTG  240

Query  241  GCTCAATACGGCATTATATGA  261
            || || || ||||||||||||
Sbjct  241  GCCCAGTATGGCATTATATGA  261
```

Figure 2.8: BLAST Output V1.0

Here is one excerpt taken from BLAST. We see that BLAST believes my query sequence to be similar to a phosphodiesterase subunit gamma-like gene of *Cyprinus carpio*. Below it details how the two sequences were aligned where the "Query" is our Zebrafish PDE6G and the "Sbjct" is the reference sequence. For the most part, there are good overlaps between these two. This is summarized in a few statistical parameters as seen in the top row. The **score** which is a measure of how well two sequences align, quantified in "bits", a measurement used in information theory. In essence, the more the better. The expect or really the **E-value** which is the expected number of alignments as good as this one found in this database (more on this later). Typically we'd like this number to be as close to zero as possible as to show that this match is more due to biological relatedness as opposed to random chance. The percent identities counts the number of exact matches and finally there is the number of gaps or breaks in the sequence.



```
Zebrafish DNA sequence from clone CH73-104F5 in linkage group 3, complete sequence
Sequence ID: CU571315.5   Length: 99283   Number of Matches: 3

Range 1: 36465 to 36607 GenBank  Graphics
Score            Expect   Identities          Gaps          Strand
265 bits(143)    5e-67    143/143(100%)       0/143(0%)     Plus/Plus

Query  1      ATGGATGTTTCCAAGCCTAAATCGACCAGCAAAGGTGCCACACGTGCCACAGGACCTGGC  60
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  36465  ATGGATGTTTCCAAGCCTAAATCGACCAGCAAAGGTGCCACACGTGCCACAGGACCTGGC  36524

Query  61     AGTCCCCACAAGGGCCCACCTAAGTTCAAACAGAGGTCAACACGCCAGTTTAAGAGCAAA  120
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  36525  AGTCCCCACAAGGGCCCACCTAAGTTCAAACAGAGGTCAACACGCCAGTTTAAGAGCAAA  36584

Query  121    CCACCCAAAAAGGGTGTCATCGG  143
              |||||||||||||||||||||||
Sbjct  36585  CCACCCAAAAAGGGTGTCATCGG  36607

Range 2: 37171 to 37250 GenBank  Graphics
Score            Expect   Identities          Gaps          Strand
148 bits(80)     5e-32    80/80(100%)         0/80(0%)      Plus/Plus

Query  182    CAGACTTTAACGTAATCTGTCCATGGGAGGCGTACAGTCACCTGGAGTTACATGAGCTGG  241
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  37171  CAGACTTTAACGTAATCTGTCCATGGGAGGCGTACAGTCACCTGGAGTTACATGAGCTGG  37230

Query  242    CTCAATACGGCATTATATGA  261
              ||||||||||||||||||||
Sbjct  37231  CTCAATACGGCATTATATGA  37250

Range 3: 37026 to 37067 GenBank  Graphics
Score            Expect   Identities          Gaps          Strand
78.7 bits(42)    7e-11    42/42(100%)         0/42(0%)      Plus/Plus

Query  143    GATTTGGTGAAGAGATTCCTGGAATGGAGGGATTGGGAACAG  184
              ||||||||||||||||||||||||||||||||||||||||||
Sbjct  37026  GATTTGGTGAAGAGATTCCTGGAATGGAGGGATTGGGAACAG  37067
```

Figure 2.9: BLAST Output V2.0

Here is another annotation prediction of the same Zebrafish query sequence, but instead showcases series of three alignments. Each of these are known as a **high-scoring segment pair (HSP)**. What this entails is that there are local overlaps that are statistically significant but outside of this alignment, the two sequences diverge, i.e. are dissimilar to each other (see figure below). Each one of these HSPs has their own statistical parameters as seen in Figure 2.8, so one could consider the alignment to *Cyprinus carpio* PDE6G-like to having one HSP while the alignment to Zebrafish clone CH73-104F5 to having three HSPs. We will get more into the mathematics of how HSPs are found when we examine the inner workings of BLAST.

The sequence database it employs can be refined by the user, as to select for a certain organism. For example, if I took tissue from Neilbob's left bollock, I would limit the database sequence to only use *Homo sapiens* as my **model organism**. If you've been paying attention and somehow really quick at thinking (as well as not given up when you got to the first part of the background text), you'd question, "If we're BLASTing against a database of known sequences, then why are we using *de novo* instead of guided assembly?" Guided assembly is highly constrained to the reference it employs while *de novo* theoretically creates all full length RNA molecules, i.e. the entire transcriptome, which allows for a greater degree of freedom as well as preventing the hassle of retrieving a bunch of reference sequences which one would get from NCBI anyways. Being able to choose an organism is useful, especially if the organism one is studying is not a common one. If it isn't a model organism, it is unlikely to find sequences online since no one has looked at them yet. To tackle this problem, one can use the closest (in terms of evolutionary time) model organism to BLAST their assembled sequences, as their organism would be more related to this model organism versus others. For example, if one is using *Oryzias latipes* or the Medaka fish as their subject of interest, one should use the model organism *Danio rerio* or the Zebrafish instead of say *Mus musculus* or the mouse.

Figure 2.10: Googling "Medaka"

Figure 2.11: Googling "Medaka Fish"

If you remember or have the ability to scroll a few paragraphs back, I talked about gene duplication. The sequences of duplicated genes can drift over evolutionary time and thus mutations can accumulate. In terms of BLASTing, if a query sequence is due to a duplication, then it could align to either of the gene doubles in the reference organism. This may not matter as the genes would have the same function... right? Not necessarily, as now one of the genes can tank mutations to its sequence and the organism won't die due its backup copy. Thus now these genes are allowed to specialize, meaning they can be expressed at different times throughout life (i.e. development vs young vs puberty vs old) and in different tissues, as well as having their own subfunctions. Their sequences however may remain quite similar to each other, which boggles biologists and baffles bioinformaticians. So now there's a problem of how to determine which out of the two database sequence is the one given by our query sequence.

As seen in Figure 2.4, let us say that we have the sequence for *Homo sapiens* SHH but we don't know that it's SHH. We could BLAST this to the set of known Zebrafish genes to suss out its identity or if its just junk. We would look at the statistical output between all alignments and choose the best one accordingly. But as we've seen with the Zebrafish, they have four replicates: DHH, IHH, SHH, and TWHH, and all of these will have high scoring alignments. One may triumph over the other, but not by much and the statistics of BLAST is rather heuristic as we'll see later. To solve this conundrum, we will employ the techinique of reciprocal best matching.

2.4.2 Reciprocal Best Matching

Hey this subsection is the name of the document, so it must be nearing the end right? Oh wait this is still the Basics Chapter, so maybe use the washroom if you have to.

To differentiate gene duplicates or **paralogs** from each other, one method is the use reciprocal best matching. What this entails is the following: BLAST the query against a database, find the best matching sequence from the database, BLAST the database sequence against the set of queries, and check that the best matching sequence finds the original query again.

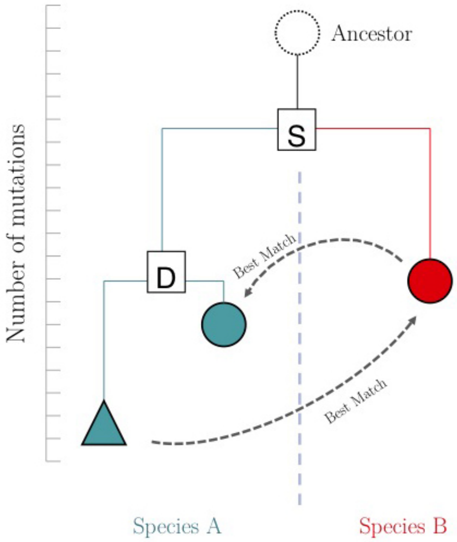


Figure 2.12: Reciprocal Best Match Idea[5]

The output from BLAST is sequentially ordered by highest scoring alignments first. As both paralogs may point to the same database entry, using the reverse or reciprocal procedure will be able to suss out the true culprit. This is convenient for if there's only duplication in one of the species. If there are 2 or more, it is not guaranteed perfect reciprocal best matching. Thus, this is the reciprocal best matching problem.

[5] Thanks to Boris Steipe, my bioinformatic professor, for "donating" this figure

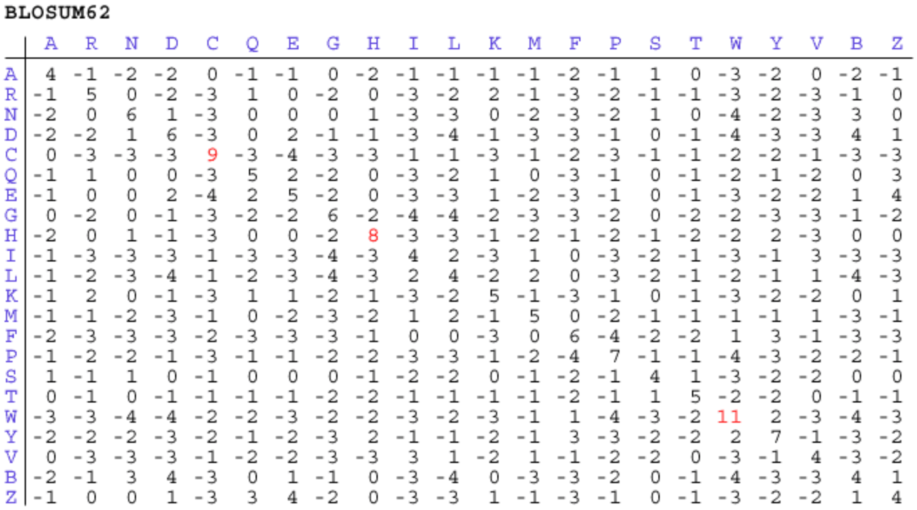
Chapter 3

Mathematical Musings

Enough of that boring, yet essential, biology lecture material; let's get into the juicy math that I promised you.

3.1 The Programming of BLAST

We begin with the idea of assigning a score to an alignment between two sequences. Let us designate the alphabet of the sequence by $\{a_1, a_2, ..., a_r\}$; for example, the alphabet of a DNA sequence would be $\{A, C, G, T\}$. Since BLAST aligns two sequences together, we assign these independent "random" sequences with letter probabilities $\{p_1, p_2, ..., p_r\}$ for our query sequence and $\{p'_1, p'_2, ..., p'_r\}$ for our reference sequence as certain genes may be biased towards having more of certain characters on others (depending on the target protein sequence selected by evolution). BLAST pairs a letter a_i from the first sequence to a_j of the second sequence; let the score of this pair be s_{ij}. Before carrying on further, we must make some assumptions. We require at least one of the scores to be positive, i.e. $\exists\ 1 \leq i, j \leq r : s_{ij} > 0$ and that the expected pair score to be negative, i.e. $\sum_{i,j} p_i p'_j s_{ij} < 0$. The reason for the former is that if two sequence segments are to match or be similar to each other, we expect their score to be positive. If there were no pair scoring values that were positive, it is impossible to have one of these HSPs. The latter, in contrast, requires that these HSPs not extend throughout the aligned sequences just by random chance; this would not be meaningful if two random sequences created by slamming one's face against a keyboard were reported as highly similar.



BLOSUM62

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4

Figure 3.1: BLOSUM 62

Scoring matrices for determining if two letters are similar to each other have been developed in the last 50 or so years. For proteins, the first iteration was the PAM suite which was created through a library of pseudorandom sequences. This is now superseded by the BLOSUM matrices, utilizing more empirical (actual) sequences. Their scores are still quite arbitrary as they are just common integers from -10 to 10 instead of convoluted quantum physics using applied orbital theory or chemical bonding mechanics. Alas, bioinformaticians feel that these matrices are suitable for harvesting desirable results (or they're just lazy). DNA substitution matrices have also been proposed, but with more confusing names (JC69, K80, T92, etc.). Typically they are modelled using a continuous-time Markov chain and thus are pretty neato.

$$P(t) = \begin{pmatrix} p_{AA}(t) & p_{GA}(t) & p_{CA}(t) & p_{TA}(t) \\ p_{AG}(t) & p_{GG}(t) & p_{CG}(t) & p_{TG}(t) \\ p_{AC}(t) & p_{GC}(t) & p_{CC}(t) & p_{TC}(t) \\ p_{AT}(t) & p_{GT}(t) & p_{CT}(t) & p_{TT}(t) \end{pmatrix}$$

Figure 3.2: DNA Nucleotide Substitution Matrix

To align two sequences, a huge m x n matrix A is created, where m and n are the lengths of the two sequences: query and reference. The entry of $A_{ij}, 0 \leq i \leq m, 0 \leq j \leq n$ is set to the score s_{ij}. We then select some top maxima above a certain threshold and try to extend these paths throughout the matrix. What I mean by this is that if s_{ij} is one of these maxima, we try to extend this HSP by looking at the scores of letters diagonally before and ahead of it, namely s_{i-1j-1} and s_{i+1j+1}. Many algorithms are based around this technique such as the Smith-Waterman and Needleman-Wunsch. However, complications can occur, such as internal parts of the gene being deleted, extended, or added, thus leading into **gaps**. What this entails is that instead of extending the HSP from say s_{ij} to s_{i+1j+1}, it may be more favourable to extend to s_{i+1j} or s_{ij+1}. In this case, there would be a letter long gap. Heuristic algorithms (scientists performing voodoo magic instead of reason) claim that creating gaps should be penalized and extending them (creating gaps of length greater than 1) should be less so. The total score of the HSP is then docked by the gap tax, whereby the values for gap formation and elongation are also reasonably arbitrary. Note here that gap penalties will throw a wrench into the works of the later statistical theory as gaps make matters terribly muddy. Quick! Try to think of all the permutations that one could get given a sequence with an arbitrary number of gaps![1]

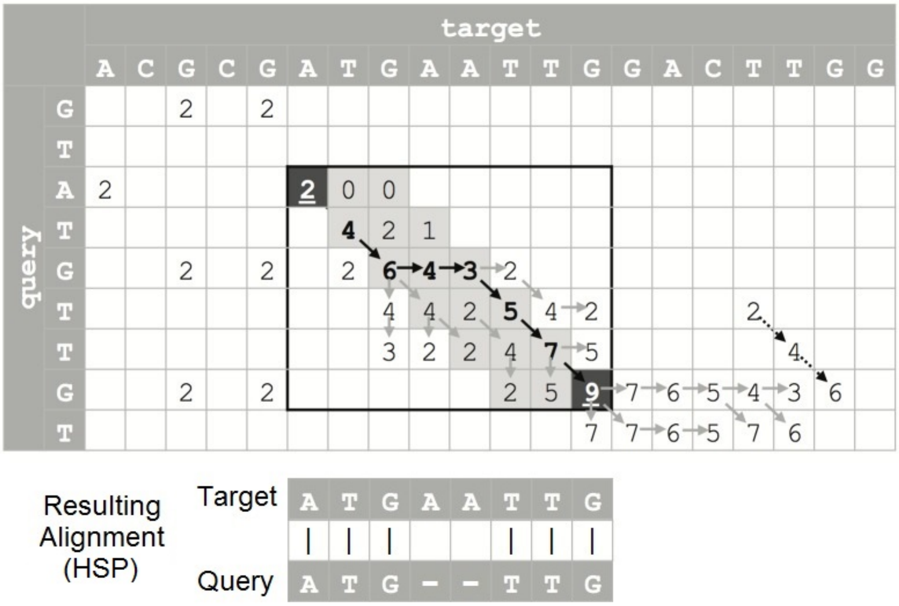


Figure 3.3: Alignment Matrix. Matches yield a score of 2, gap creation penalty is -2, and gap extension penalty is -1

3.2 The Statistics of BLAST

So what, we can align two sequences and find some high scoring alignments using BLAST. Matt, what the deuce is the point of all this chatter? Here is the first problem out of the trinity. We've seen that BLAST outputs a series of

[1] You should get $\aleph_0$. Now that's gonna take a while to compute.

HSPs each with their own statistical parameters. What do they mean exactly? Better yet, how should we apply them? To summarize those questions, I will ask a third: given a HSP, what is its statistical signifance in relating the query to a reference sequence?

I have tackled this problem for ~~3~~ 4 days so far. I have not much to report as it is way to dense the statistical papers it is derived from. But I'll try my best.

Let us recall the properties of a Poisson process. Let $\mathbf{X} = (X_1, X_2, ...)$ denote the sequence of inter-arrival times, $\mathbf{T} = (T_0, T_1, T_2, ...)$ denote the sequence of arrival times, and $\mathbf{N} = (N_t : t \geq 0)$ as the counting process. Based on the strong renewal assumption, we know that the process restarts at each fixed time and each arrival time, independently of the past. For some $\lambda \in (0, \infty)$, $\mathbf{X}$ is a sequence of independent $Exponential(\lambda)$ variables, $\mathbf{T}$ is a sequence of independent $Gamma(i, \lambda)$ variables for $i \in \mathbb{N}$, and $\mathbf{N}$ is a sequence of independent $Poisson(\lambda t)$ variables for $t \geq 0$. Some of these processes can be rearranged to get others, specifically

$$N_t = max\{n \in \mathbb{N} : T_n \leq t\} \qquad T_n = min\{t \geq 0 : N_t = n\}$$

Now onto the paper. Recall the finite alphabet in use is $\{a\}_1^r = \{a_1, a_2, ..., a_r\}$ (for DNA, think A, C, G, T). Let $X_1, X_2, ..., X_{mn}, ...$ be independent and identically distributed random variables based on observations from a pair of two letters a_i and a_j such that

$$\mathbb{P}(X = s_{i,j}) = p_i p_j', \quad i, j \in \{1, 2, ..., r\},\, p_i, p_j' > 0,\, \sum p_i = 1, \sum p_j' = 1,$$

is thought of as sampling the pair of letters a_i and a_j yields the score s_{ij}. In layman's terms, this shows that the probability of getting a pairwise alphabet score $s_{i,j}$ is the probability of the first letter a_i appearing multiplying the probability of the second letter a_j. For example, refer to Figure 3.3 where we aligned a query against a reference/targer sequence. It appears that if A is matched with A, we get a score of 2. If the probability of an A is 1/4 in the query and 1/3 in the reference, then the probability of aligning two A's is 1/12. Carrying on, we now define a partial sum process

$$S = \{S_{(i,j)\to(i+\min(m-i,n-j),j+\min(m-i,n-j))} : i = 0,\ 0 \leq j \leq n\ or\ 0 \leq i \leq m,\ j = 0\},$$

$$S_{(i,j)\to(i+c,j+c)} = \sum_{k=0}^{c} s_{i+k,j+k}$$

where m and n represent the lengths of query and reference sequences respectively. All that this mumbo jumbo represents is the total score of some path starting at (i, j) in the matrix and ending up at $(i + c, j + c)$ by adding up all the scores in that diagonal path. Some (proof-esque) assumptions are made on this process, namely that the moment generating function for X exists, i.e.

$$\mathbb{E}[e^{\theta X}] < \infty, \quad -\theta_1 < \theta < \theta_2, \quad \theta_1, \theta_2 > 0$$

and that the expected value for X is negative, so that $\{S_{(i,j)\to(i+q,j+q)}\}$ is negative for sufficiently large q. We define two seemingly "random" (both actual and pun) quantities

$$M(k,l) = \sup\{S_{(i,j)\to(i+\min(k-i,l-j),j+\min(k-i,l-j))} : 0 \le i \le k, 0 \le j \le l\},\\ 0 \le k \le m,\ 0 \le l \le n \tag{1}$$

which corresponds to a path with maximal score within a rectangular block of size $k \times l$ in the scoring matrix and

$$T(y) = \inf\{\sqrt{k^2+l^2} : M(k,l) > y\} \tag{2}$$

which is the smallest partial sum process, i.e. the shortest path with a score greater than y. Note that according to the definition of $T(y)$ that

$$M(k,l) = \sup\{y : T(y) < \sqrt{k^2+l^2}\}. \tag{3}$$

According to our Poisson process discussion above, these last two identities look familiar. $M(k,l)$ looks like a part of a counting process while $T(y)$ appears to denote arrival time of a HSP of a score greater than y. But in actuality, not quite at all. Their distributions are dependent on a certain parameter defined as θ^*, which is the *unique positive* root of the equation $\mathbb{E}[e^{\theta X}] = 1$. If you can tell me why we need this particular solution, let me know. Now for the part that you've accidentally glimpsed anyways and already began to dread trying to help Matt. Here's a modified statistics dump from the minds of Iglehart (1972) and Karlin, Dembo, and Kawabata (1990): *When X is nonlattice*

$$\lim_{\substack{m\to\infty,k\to m\\ n\to\infty,l\to n}} \mathbb{P}\Big(M(k,l) - \frac{\ln mn}{\theta^*} \le x\Big) = \exp(-K^* e^{-\theta^* x}), \tag{4}$$

where

$$K^* = \frac{\exp\bigg(-2\sum_{k=1}^{\infty}\frac{1}{k}\Big(\mathbb{E}[e^{\theta^* S_k}|S_k<0] + \mathbb{P}(S_k \ge 0)\Big)\bigg)}{\theta^*\mathbb{E}[Xe^{\theta^* X}]}. \tag{5}$$

where S_k is a random variable representing the sum of the scores of k independently chosen letter pairs; and when X is a lattice variable of span δ, 4 is replaced by

$$\begin{aligned}\exp(-K_+e^{-\theta^* x}) &\le \liminf_{\substack{m\to\infty,k\to m\\ n\to\infty,l\to n}} \mathbb{P}\big(M(k,l) - \frac{\ln mn}{\theta^*} < x\big)\\ &\le \limsup_{\substack{m\to\infty,k\to m\\ n\to\infty,l\to n}} \mathbb{P}\big(M(k,l) - \frac{\ln mn}{\theta^*} < x\big) \qquad (6)\\ &\le \exp(-K_- e^{-\theta^* x}),\end{aligned}$$

where

$$K_{-} = \frac{\theta^* \delta}{e^{\theta^* \delta} - 1} K^*, \qquad K_{+} = \frac{\theta^* \delta}{1 - e^{\theta^* \delta}} K^*.$$

Are you frightened? Don't be (yes you should be actually I don't understand what I just typed out either). Technically for our purposes we should be using Equation 6 since we are dealing with a lattice case instead of a nonlattice one (we're using a well ordered set i.e. $\mathbb{N}$ as opposed to ones with multiple infinitums or supremums). However, δ is relatively small so the inequality bounds are sufficiently close to the magical number K^*. The details of how these are defined are gratuitous, but may be helpful if you are powerful enough to wield a PhD level of statistics as this next part makes no sense to me. Karlin, Dembo, and Kawabata (1990) claim

> *"A concomitant of [the previous theorem] is that the asymptotic* $(n \to \infty)$ *distribution of the number of separate excursions attaining a score in excess of ln* $n/\theta^* + x$ *is Poisson with parameter* $K^* e^{-\theta^* x}$*".*

How they ever managed to assert this (and everything before it) without a nicely laid out proof is quite irritating. This statement on the contrary suits my needs decently. In English, the number of HSPs between a query and a database hit follows a Poisson distribution with a particular parameter. This parameter, however complicated it appears to be, is just a number. Treat it with as much respect as this manuscript: next to nothing. What is fancy about it is that we don't have to even think about calculating that parameter; we can just read it from the output of a BLAST file. "How?" as Atticus Finch once famously quoted. Think about the *expected value* of a Poisson distribution (or Wikipedia it). What you should get is the parameter you put in. Rather convenient. For our purposes, this number is clearly listed as the **]glsE-value**. In conclusion, the parameter $K^* e^{-\theta^* x}$ is the E-value for a score x.

Let us talk about briefly the two constants involved in the E-value: K^* and θ^*. These are indeed constants and not variables. BLAST performs some trickery by taking all the sequences in a database and calculating all letter frequencies in the sequence alphabet used. It then shuffles the innards of sequences about, creating pseudo sequences. Using these, it then applies math (oooh spooky) hard coded in the inner workings of the program and spits out these two constants. Yes I don't know how it exactly works either but the documentation kind of sucks for explaining it more than this. But in essence, these two parameters depend on the database used and will differ from one to another.

Given our newly founded Poisson distribution, we can calculate p-values in order to show how good these HSPs really are. As a reminder, the p-value represents the probability of finding a match assuming some base/null distribution, in our case, we'll be using a Poisson distribution with the appropriate E-value. Now we can fully calculate what are the chances of getting at least one HSP greater (or equal pretty much) to a given score s:

$$\mathbb{P}(1 \textit{ or more HSPs scoring} > s;\ E) = 1 - e^{-E}$$

where E is the E-value. We are finally able to answer the question I asked in the beginning of this section. Using this formula, we can associate a HSP with a p-value, which relates the query to a reference sequence in a statistical sense. But as seen in Figure 2.9, we can have multiple HSPs. It is a trivial matter to generalize our equation, which is a step that I must partake on for my quest. Employing the quote, the probability of finding at least n such HSPs scoring greater than s:

$$\mathbb{P}(n\ or\ more\ HSPs\ scoring > s;\ E) = 1 - e^{-E} \sum_{k=0}^{n-1} \frac{E^k}{k!}$$

So all I would have to do is given a query that contains several HSPs matching a database sequence, I take the lowest score and apply the equation above to discover a singular p-value.

Let me summarize this section for you. We wanted to know how to relate a HSP's alignment score to a probability. We've seen that this fundamentally relies on a Poisson distribution with an associated E-value. We then decided to extend our question, well what if our alignment has multiple HSPs? Applying the concomitant yields another trivial use of the Poisson distribution involving a summation over the number of HSPs as well as using the lowest HSP score. How this is useful is this: given any number of HSPs, we can derive a single p-value to show statistical significance that the query and reference sequences are related. One question down, two to go!

An ammendment. I listed all these bludgeoningly hard mathematical excerpts for two reasons. Firstly, to scare you. Secondly, to show that it is easy to just accept a conclusion from a paper but to reverse engineer it proves mre than aggravating. The quote I showed you will do fine, but it applies to if the HSPs have the *same score*. So what if the HSPs have different scores? Then this lemma doesn't apply. That is why I took the minimum score in order to apply that conclusion. Could I instead consider these differentially scoring HSPs to be independent and with their own respective Poisson distributions? I personally believe I could as they would be *sufficiently* far enough apart to not become one big HSP and thus could be considered to be independent. But I am uncertain as I cannot parse through the sheer statistical obfuscation seen in these papers. I mention this now because this may be a better mathematical approach. If you can lift the sword from the stone and crack these statistical codes, let me know. But for now, onwards!

3.3 The Theory of Information

Here's a part that could be super easy or the worst of the trinity of problems. For the record, this is the second conundrum; we solved the first in the last section. Now we explore the fringes of statistics, mainly information theory.

What is information? One could define it as a particular sequence of characters that gives meaning. For example, this very text your reading is considered information as opposed to apqojc kljivo joi, which appears meaningless. Let us see how this idea apply to our problem.

We have two sets of DNA sequences, a set of query sequences and a sequence database. We utilize BLAST to align each query, resulting in one or more HSPs with one or more reference sequences, each with their own score and thus E-value. We condense these into one meaningful p-value using the equation derived in the last section. Now we perform the reciprocal procedure: we BLAST the database against the set of queries, find HSPs, and create another p-value for each reference sequence. Now imagine one of these queries aligned to a single reference sequence, each with a newly created p-value relating to each other.

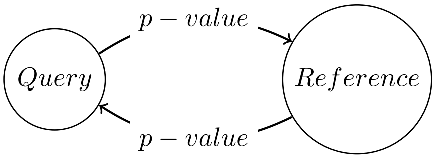


Figure 3.4: Query VS Reference

These p-values may not be necessarily the same due E-value depending on K^* and θ^*, or there may not be the same number of HSPs, or the same scores for each of these HSPs, or the guarantee of there being any HSPs to this query at all. So we're left with two p-values, one denoting the probability that the query forms a set of HSPs of greater score s with the reference and vice-versa. The question is now: can we further condense these p-values into one measure of significance such as the probability that these two sequences are the same?

Now this is going to be the sparsest of the sections because I have no clue how to even attempt this problem. But here's some small analysis that I can give to show that I believe that information theory is the most likely candidate to solve this problem. First of all, we're dealing with sequences that encode information (DNA encodes instructions to create proteins with) and these sequences must be of a certain configuration to be biologically meaningful. Secondly, information theory has a concept called **cross entropy**. This looks at a sequence that utilizes the members of a given character set and two probability distributions which one of them is used to define the frequencies at which the characters are used in the sequence. Cross entropy looks at the average number of characters needed to be drawn in order to distinguish between which probability distribution that was utilized for the underlying sequence. Applying this to our example, our character set is the set of aligned DNA nucleotides (i.e. A-A, A-C, A-G, A-T, C-A, C-C, etc.) and our probability distributions could be the frequencies of As, Cs, Gs, and Ts in the pooled query and reference sequence (i.e. combine both query and reference sequences and use the DNA nucleotide frequencies there) and the distribution could be uniform (i.e. A-A = A-C = ... = 1/16). (Note the following figure is filler until I feel less lazy to actually do the example I say here)

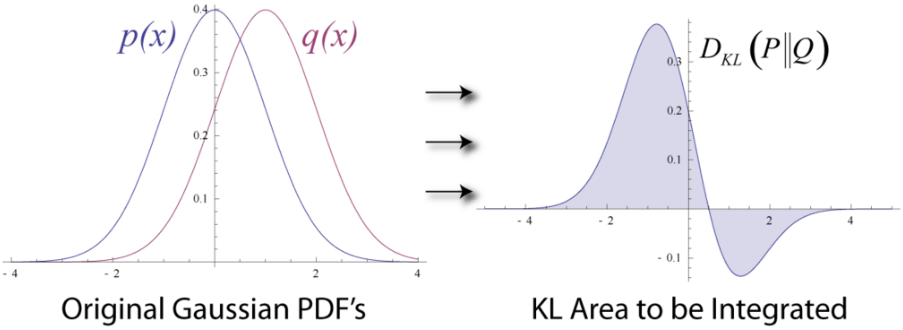


Figure 3.5: Cross Entropy, shows how well the probability distribution $q(x)$ can explain $p(x)$

Now I'm not certain upon whether the original two p-values could be somehow used to create certain distributions in order to apply some information theory and other mumbo-jumbo, but I am certain that there is some way to compute a nice, succinct number that relates the query to the reference sort of like a singular p-value. What this new p-value could be interpreted as the probability that these two sequences are the same given two random sequences created from aligned nucleotide frequencies from the pooling of the query and reference characters. Or not. I'm not sure. So any help on this matter would be greatly appreciated.

3.4 The Theory of Graphs

All my previous discussion has been leading up to graph theory which I have next to nil in experience and only cursory knowledge/terminology in. I'll try not to embarass myself too hard.

Let us envision a bipartite graph G = (Q, R, E) where Q represents the set of query sequences, R represents the set of reference sequences, and E be a directed edge set from Q to R or R to Q. "But Matt," you'll interject, "you can't have a bipartite graph with directed edges!" Bear with me, it'll do for our purposes. Each edge is associated with an HSP with its strength equal to it's p-value. For some vertex $q_i \in Q$, we can have multiple directed edges going to say vertex $r_j \in R$, each corresponding to a different HSP located within the aligned sequences. Here is a beautiful rendition of my blundering about:

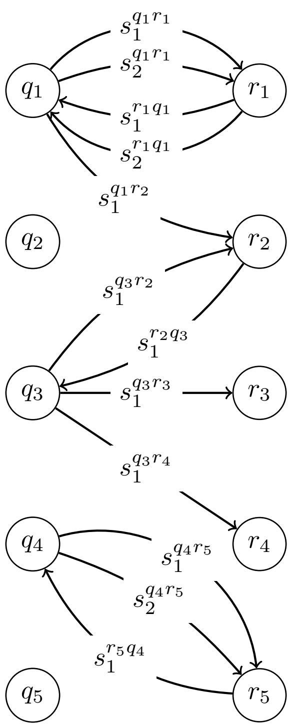
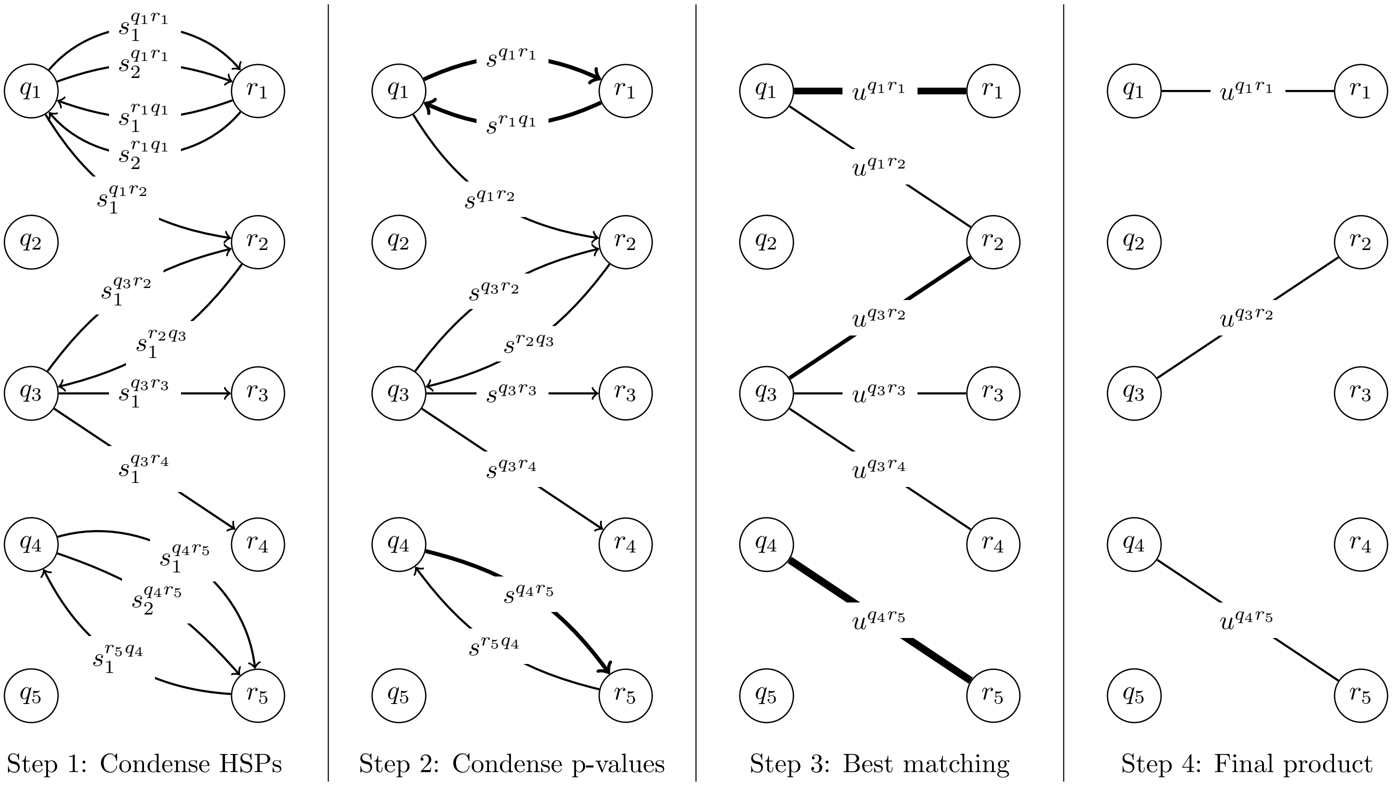


Figure 3.6: Directed Bipartite Graph

Now I want the *best matching* for this “directed bipartite graph” if that makes any sense, which it shouldn’t. This is the final problem of the trifecta and why we underwent some spurrious details on how to think of the other two. We needed to solve the other two before we could tackle this one because without those, we cannot employ a best matching algorithm. What I mean is this. We take our set of directed edges $\{s_1^{q_i r_j}, s_2^{q_i r_j}, ...\}$ from q_i to r_j and condense them into a single directed edge $s_{q_i r_j}$, repeating this procedure for all pairs of vertices. We do this by solving the first problem: sequester the separate HSP excursion scores into one meaningful p-value. Then we rid of the directed edges by merging $s_{q_i r_j}$ and $s_{r_j q_i}$ into one unordered pair $u^{q_i r_j}$ by solving the second problem: merging the p-values into some quantity to signify their relationship. Finally we solve the third problem: look at these resulting edges and their strengths to find the best matching between the two sets of vertices. In diagramatic form:



Step 1: Condense HSPs

Step 2: Condense p-values

Step 3: Best matching

Step 4: Final product

Figure 3.7: Overview of Methodology

This is probably the easiest of the trio of problems, such that you can just yell an algorithm name in my general direction and you'd get the Nobel prize for your effort. I would like some control over the algorithm though, just due to the nature of biological sequences and paralogs in general. Perhaps a species I'm using the query has a single copy but my reference had a duplication. Then I would like the singleton to be used for both references, but alas that may be too difficult and too error prone. Just an algorithm or some nice caring words would suffice.

Chapter 4

Finale

Congratulations! You've read all 30 of my beautiful LaTeXdocument! I've never worked with this program before and I can tell you this: it sucks. It really does. I mean trying to get the figures the way I wanted them was really just trial and error as well as profuse use of Tex - the LaTeX Stack Exchange (equivalent of StackOverflow). Writing this document was relatively easy, stealing and editing figures in MS Paint was moderate, but trying to typeset all the equations was terribad. Why did I choose LaTeX? I knew it could make equations look nice, which it does for the most part, but also it could make vertices and edges like those in graph theory. Man the documentation for that part is in French so yay partial bilingualism (`http://mirror.its.dal.ca/ctan/macros/latex/contrib/tkz/tkz-graph/doc/tkz-graph-screen.pdf`).

And now you should definitely be solving my problems. Come on, you don't even have a pencil ready and you read the whole thing. Gosh. I'm kidding of course, this document doesn't have a macro that interfaces with a webcam, sends the visual data to a UofT server which analyzes the objects it sees in front of it, and autogenerates this paragraph subsequently. It's me wasting your time, my time, and my professor's grant money on this document. But hopefully you can help a poor young soul out. I'm counting on you friendo.