# CSCI 4180

Introduction

*Blouin*

**Alex Safatli**

Thursday, September 6, 2012

# Contents

## Class Details

- **Optional Textbook**: *Understanding Bioinformatics*
- **Midterm Tests (x2)**: Open-laptop/book, short answer format.
- Hybrid of Graduate Students, Undergraduate Students – ultimately, treated the same.
- **Assignment #1**: Due later – keep in mind for first 2, 3 lectures.
- **Term Project**: Will be due on November 29, 2012. Poster.
- **In a Nutshell**: Introduction course into Bioinformatics – no assumed knowledge. Build up vocabulary. Dealing with large amounts of data.

## Let's Talk About Bioinformatics

### Brief Overview

- Remember: Many elements of biology and life are not supposed to make sense. Supposed to work.
- Bioinformatics is a *data-driven discipline.*
- This class features a broad coverage of what most people agree is bioinformatics.
- Algorithms in bioinformatics will be looked over on parsing information, etc.
- Evaluation scheme online in syllabus.
  1. **Written assignment** — 40%
  2. 2 × **Midterm Tests** — 30%
  3. **Project** – Identify a problem. Make a poster (way of delivering smaller results). — 30%

### Assignment #1

Take all concepts and map them, organize them in a way that makes sense. Uses an open-source mind-mapping software known as *vym.* There is no single solution for this and it will feel like an egg hunt. All of the concepts that must be mapped will have been talked about in lecture. Should be enough information to cover the biology needed for bioinformatics.

**Mindmapping** is a way to get a chance to sort of terms and arrange them into a structure that makes sense (to you). Useful way of organizing thoughts. The core of the map should be *molecular biology.*

## The Basic Types of Lifeforms

What is **life**? How large is the range of diversity and types of organisms? There is a *structure* to life's diversity: three classifications of life.

1. **Eukaryotes**: Cells with a nucleus; humans, plants, etc. Theory is that an archaea absorbed a bacterium that eventually lost genes due to not needing to defend itself, etc. This became the mitochondria – creates energy.

The Basic Types of Lifeforms continued on next page...

2. **Archaea**: Look like bacteria; more closely related to us. Single-celled microorganisms. An example of a prokaryote.
3. **Prokaryotes**: Bacteria. All lack a nucleus; everything is inside of the cytoplasm. Good and bad ones exist: cheese and yogourt vs. Black Plague. Possess a smaller genome. Many live in exotic environments.

There is a large diversity of bacteria. They are very efficient, and their survival depends on how fast they can reproduce. Thus, they thrive on simplicity and streamlining of processes.

**NOTE** A lot of bioinformatic tools are intended for: (1) detection and (2) prediction.

A further diversity of life (which are not alive) are **viruses**. Viruses are simple, and behave like they're sentient. They use genetic encoding and can thus be treated like lifeforms. Furthermore, their sole purpose is to replicate and not do much more. This is typically accomplished by hijacking cells. If bacteria are optimizers, viruses are *super-optimizers*. They have even smaller genomes.

Finally, another self-replicative classification of entities are **prions**. These are *not* genetically encoded and these proteins can be folded into different configurations where abnormal configurations can self-catalyze their formation onto normal proteins when in contact, creating the illusion of replication.

## Genetic Encoding

### The Tree of Life

As a brief exercise, visit `http://tolweb.org`, the website for the Tree of Life. This allows you to go from a species and up through enclosing groups. For example, one could go from *Homo sapiens* up to the genus *Homo* and up the hierarchy of classification through to mammals and eukaryotes.

This is a tiny fraction of all biodiversity. All classification of this nature are done at a genetic level.

### Encoding: Translation and Transcription

Living cells encode genetic information of which are turned into effectors (proteins). A generic eukaruotic cell contains all of this genetic information within its **nucleus**. Just about everything important about a lifeform is contained inside this genetic information.

Genetic information can be represented as sequences of nucleotides of **Deoxyribose Nucleic Acid**, or DNA, which are what are transferred from one generation to another. In DNA, the nucleotides A and T and complimentary, as are C and G. This allows for a secure geometric shape where two complimentary strands twine together to form a double helix. This allows for protection against the environment, controlled redundancy, and mechanical protection among other things.

The encoding of this information is done in a number of steps.

**Step 1**: *Transcription from DNA to RNA.* RNA polymerase goes to unwound DNA and creates an RNA transcript copy of a strand. This is essentially a functional copy of the genetic information that can be used elsewhere in a cell.

**Step 2**: *Translation to Proteins.* This functional copy is read in words of 3 $\times$ characters, or what is known as a **codon**, to chains of amino acids that form a protein. Ribosome tRNA reads these words and attaches

Encoding: Translation and Transcription continued on next page...

these amino acids together. What is happening here is a 4 character alphabet is being translated to a 20 character one.

The interesting thing about the genetic code is that it is **universal** and **degenerated**. Different species have different codon bias. And there exists some redundancy in the translation stage to a protein such that the effects of mutations can be limited.

**NOTE** It's interesting to note that while DNA sequences can be handled as strings; one type of task that can occur in Bioinformatics is the identification of DNA palindromes which are known as a indication of possible cancer.

### Proteins

Proteins are chains of **amino acids**. Encoded by genes in DNA, these proteins go on to act as what was labelled previously as effectors. They bring together key amino acids to catalyze otherwise impossible reactions, carry messages, or assemble into molecular machines and superstructures. Most of the reactions in our body are impossible; they require enzymes, etc.

**Protein folding** is the chemical process of transferring these 2D sequences of amino acids into a unique 3D structure where each amino acid have different electrostatic charge, hydrphobicity, and chemical reactivity. Together, these define structural features within a protein.

It also defines a very interesting constrained optimization problem where repulsion has to be minimized, attraction has to be maximized, and hydrophobicity has to be managed. For instance, folding onto itself, amino acids in a protein can protect themselves from water. This is the protein folding problem.

## Molecules to a Text String

### Objective and Principles

Describing how the sequence in a molecule can be converted to a text string is the objective of this section.

The principle by which this is done is noting that DNA cannot be read using microscopy. It can be seperated, however, on the basis of length such that the DNA length can be determined very easily through **electrophoresis**. Smaller pieces of DNA will travel further than larger pieces through gel in electrophoresis. This allows seperation of specified genes for genetic engineering, along other things. Sequencing is now mostly automated. No more than a few thousand nucleotides can be read in one reaction.

A really simple algorithm can be defined for describing genetic sequencing (see slides). The algorithm involves the generation of a collection of all possible length DNA replicates, terminated by florescent dye nucleotide. This mixture of DNA replicates is separated by length; color of the DNA is read, and the order of color is reversed where color is replaced by their corresponding characters.

The copy mechanism has also been engineered to be used *in vitro* in test tubes. What is required is a template DNA, a DNA polymerase protein, a primer that is complementary to the template, and free nucleotides (building blocks). **A given sequence of DNA can be copied as long as we know the sequence that is just upstream of the sequence of interest**.

Molecules to a Text String continued on next page. . .

### FASTA Format

Most basic format to store sequence information only. This is what is usually downloaded from a database. Cannot really sequence proteins, but we can infer them.

Finding sequences can be done by looking at the central repository for sequences: the National Institute of Health (GenBank). Allows keyword search, geome browsing, structure searches.

## Factoids about the Human Genome

The first genome came from 2 random males and 2 random females from a pool of 20 each. The private project involves 5 individuals, 1 of which is likely to be that of the CEO of CELERA, the company involved. The first draft in 2001 was by TIGR and CELERA. We're still not at 100%. We are at about 92.3% as of 2003. Low-complicity regions are still unknown. The challenge now is to define nucleotide polymoprhisms between known ethnic groups.

The Clinton administration rejected the patenting of human sequences in 2000. Had it gone through, royalties would have to be paid for the production of medication, etc.

Other future ethical issues involve whether or not it is okay for an insurance company to require someone's genome. Or if it is better for someone to know that they are higher risks for conditions. Even moreso, should genetically modified food be tested and sold to third world countries where they are most needed? In the case of GM food where you want to test for secondary effects, who would you use as a non-control group?

It's also interesting to note that we can repair mutations. Proteins available to go around and scan for mismatches where it will remove and fill in correct sections. Error correction has a 50% chance of accidentally putting in the wrong section. But, only about 1% of our DNA is gene-encoding, so mutations are not a cataclysmic event always.

### What Is In A Genome?

If you look at the human genome, we can define a breakdown of what our genes are related to. There is almost no such thing as a Human-only gene. 22% are shared with verebrates. More than half of our makeup is shared with animals. The rest is shared between eukaryotes and prokaryotes. Interestingly, three's a tiny percentage of Prokaryote-only genes. From an inheritance point of view, this makes no sense. However, when we take in bacteria, there is a chance when they die that the information can become a part of our genome. Some of them may even be mitochondrial DNA.

## Sequencing a Genome

The objective of this section is to describe how large genomes are sequenced using the "shotgun" strategy. Often when you want to get at DNA, you have to clone it (organismal cloning is not what is referred to here).

### Vocabulary

- **Genes** are specific stretches of DNA or RNA.

Vocabulary continued on next page. . .

- **Plasmid** refers to an artificial construct used to manipulate sequences. Working plasmids can be cut at any location and be populated by genes of interest. These can be put into bacteria which will then begin to reproduce and create copies.
- **Cloning** refers to making a copy of a segment of DNA. It is a routine laboratory technique where a bacteria that is easy to grow can amplify pieces of DNA. Associating an antibiotic resistance gene to the cloning plasmid can provide artificial selective advantage to contain the cloned DNA. The end result is some micrograms of DNA.
- **BAC and YAC** are artificial chromosomes or plasmids from respectively bacteria or Yeast.
- **Library** refers to the extraction of whole cell DNA, cleaning it up, and breaking it into random fragments (150 kbp BAC, 0.15-1.5 YAC).

## Ways to Sequence

Generating primers from the 3' end of the newly generated sequences leaves no gaps. However, this is slow (sequential) and expensive and is known as **chromosome walking**. Furthermore, you cannot get the primer for your next reaction until the end of a given one.

**Shotgun sequencing** involves randomly breaking genomes into smaller fragments. Using random primers, sequencing DNA fragments are generated. Ultimately, you assemble the substrings on the basis of their overlap. The disadvantage of all of this is that there are merely regions which do not like to be sequenced in this fashion which are highly repetitive.

Coverage (C) refers to the expected number of times that a single position was sequenced. $C = nL/G$ where $n$ is the number of reads, $L$ is the average read length, and $G$ is the size of a genome. This has a diminishing return, however.

In practice, multiple copies of genomes are sheared into random fragments. By size, they can be isolated and from these clones, one at a time can be taken to generate a read. Finally, these reads are assembled into contigs (continuous regions) and then the order is determined of these contigs, categorizing them into scaffolds.

This was expensive for humans, whereas it was easy for bacteria. There are even larger genomes than humans'.

*Next-generation sequencing* involves **pyrosequencing**, or 454 sequencing, which uses a different and far more complex method. This is extremely cheap and fast (250× faster). The result is the production of very small reads. Furthermore, these are more difficult to assemble.

So, ultimately, the most efficient method to cover a genome is to randomly sample sub-sequences until most of the genome is covered at least once. The next step is purely computational.

## Algorithms to Assemble Genomes

Given a genome G and a given set of reads, you can control, up to a certain level of confidence, whether the ends match between strings of nucleotides.

The **greedy** algorithm outputs the shortest possible sequence which overlaps all N reads. The $n^2$ problem involved is nothing compared to the fact that it typically does not work. Theoretically, you cannot backtrack and consider alternatives if a wrong decision was made earlier. This will most likely happen – you may just end up stopping and having free-floating reads which may be quite important. Practically, this algorithm is

Algorithms to Assemble Genomes continued on next page...

$n^2$ and thus impractical when $n$ grows very large. Repeat sequences also choke the algorithm and removes pieces in-between repeats. Be aware that between two strings, a score can be achieved.

The **overlap-layout-consensus** algorithm outputs the same thing. Unlike the greedy approach where you do merging, this involves graph theory and looking for the Hamiltonian path. Edges of this graph are merge events. Deleting matching substrings is essential because not doing so would result in circles in the graph. Moreso, we did not lose pieces between repeats. The presence of gaps would also correspond to the presence of separate components in the graph. The downside of this is still the same as the greedy approach – it is slow. The overlap can be of variable length between two strings. Be aware that between two strings, a score and p-value can be determined.

The **de Bruijn** graph approach is the best method. While the OLC approach requires an all versus all alignment step (and is not a symmetric graph) to initially determine where edges are (an equally problematic consequence of the greedy approach), the de Bruijn graph solves this problem by running in linear time. It is an Eulerian path solution (go through every edge) and is a word-based method also involving graph theory. One pitfall to this is that you have to create as many vertices as there are nucleotides. Each edge in the graph is an explicit connection between two characters. The meaning of an edge in these graphs is simply the indication there is an indication of overlap.

**NOTE** The $k$ value is picked as defined in the paper on Moodle.

The reality is that 25% of human genes are present in at least 2 copies. Repeats increase with age (a reason why Dolly the sheep procured old-age diseases early in teenagehood since the cells knew how old they were) So even the OLC and this solution will have great deals of trouble getting a correct solution.

Artifacts that can occur in the de Bruijn graph due to this include spurs, bubbles (common), and frayed-ends.

**Scaffolding** allows the use of external data to delete edges in a graph, for instance. If you know a certain part of a genome has to be before another, you can now begin to remove edges. Furthermore, due to knowing two given reads are from the same clone and thus are part of a pair, you can infer they must belong to the same path. This will simplify the path and resolve graph complexity from the creation of these constraints.

So overall, random reads are taken of a given genome, and then a shortest path of some sort is taken in order to collect the genome sequence. Both the OLC and de Bruijn approaches use graphs in order to achieve this.

**NOTE Metagenomics** involves the gathering of genetic material from an environmental sample. For example, one could go out to the ocean and get a huge bucket of water. Sequences of this material involves a sequencing of different sources of genetic information where one has no idea where the specific sources are. Much of it will come from bacteria.

## Article Study: *De Novo* Genome Assembly

The web location for this document is `http://www.biomedcentral.com/content/pdf/1471-2105-13-S6-S1.pdf`. Recall that there are two major algorithms for genome sequencing:

1. *OLC – Overlap-Layout-Consensus*: A vertex is a read and an edge is a significant overlap. Has a time-complexity issue.
2. *de Bruijn*: A vertex is a k-mer and an edge is a relationship between k-mers or a nucleotide overlap.

Article Study: *De Novo* Genome Assembly continued on next page...

The scenario that occurs in OLC (and de Bruijn) typically is a high series of overlapping reads where a significant amount of redundancy is present.

On page 3, in panel (a) on the figure, before any changes, a de Bruijn graph is described with an image where edges are not explicitly shown. Simply shows that the substring can be followed by a $G$ or $C$. The edge is not stored but can just store it in a dictionary and do hashing in constant time. The next vertex can be inferred from this.

But there are only four nucleotides! Can only allocate 2 bits per character and for the possible following characters, simply apply a 4-bit map (0 or 1 for each character). Even further than this, we can have $g$ number of these "tags" at the end of a k-mer.

By mapping the end of a k-mer to a longer set of tags, a single pair of vertices can be stored (with the only disadvantage being you are storing a larger tag).

## Molecular Evolution

**Evolution** refers to the idea of "survival of the fittest" — fittest here not necessarily refer to strength but to the fact that those that survive did not die. It is the process upon which information is transmitted from one generation to the next and how this subsequent information changes over time. This is a concept central to bioinformatics because it is the process that defines the relationships within all data in a dataset. It is *not* the driving force of making things better, but this can happen by accident.

### Divergence

Molecules provide characters to compare evolving things. Over time, sequences are changing. Comparing two sequences imply some understanding of the process by which this happened. **Evolutionary divergence** is a **stochastic** (random) process where evolution is the consequence. As sequences are randomly mutated over time, two sequences sharing a common ancestry will *necessarily diverge.*

**NOTE Convergence** can also occur if two unrelated sequences evolved to a similar solution because they are climbing the same hill. This is thought to be rare for sequences (but not as much for structures in the macro-world). An example is bat wings and bird wings.

The relationship between biological entities, therefore, assuming only vertical descent, is *tree-like and is directed.* The **root** is the last common ancestor to all leaf nodes. **Internal nodes** are notional ancestors common to all ascendent nodes. **Leaf nodes** are observed entities. And finally, **branch lengths** are quantities of evolutionary distance.

Today, the tree of life has *no root.* Not enough mathematical power yet, but we can infer it is somewhere between two locations in the middle. The closest thing to animals on the tree are fungi. There is also this line of thought where there is no root at all and somehow a pool of organisms coalesced into what we know as life.

### Selection and Mutation

**Natural selection** is typically thought of by the positive selection: the favouring of differential reproduction of 'good' solutions. A good example of this is an epidemic of the influenza virus which may wipe out millions of people. A copy of the virus with the right mutation to be immune to the antibody measure could go on and spread to cause another epidemic. This is surprisingly the most rare type of selection. Negative selection is referring to the elimination of bad solutions – a purifying selection.

The **neutralist hypothesis** is referring to the idea that mutations far outpace phenotypic changes. Most genetic change is not subject to selection. Mutations that have no detrimental effect will remain – most mutations are neutral and do not do anything of significance. The main purpose of the force of diversification is that when the rules *do* change, there will be surviving entities. These will be the ones that diversify again in turn. So evolution did not make things better – it merely made a fan big enough such that a few can luckily survive.

**Diversity** refers to the range of diverging solutions to an ecological situation. In a population divergence is important because of **adaptation**. Adaptation is negative selection that occurs within the current diversity when the "fitness function" changes (an environmental change occurs) according to the neutral hypothesis. Without diversity, fewer instances in a population will survive dramatic changes in the environment.

So what is a **mutation**? It is the root causes of diversity. The most common one is a point change – a change at a single nucleotide location which may change the amino acid in a protein. Natural mutation occurs when we replicate our DNA (5' to 3') where the DNA polymerase has an intrinsic error rate. Base pairing is not perfect and since copying DNA relies on this, errors can occur.

Because viruses replicate so quickly, they adopted a measure of reproduction with a high error rate. Eukaryotes have a lower error rate. If the mutation rate is too slow, there will not be enough genetic variation that may affect adaptability in cases of positive selection. If it was too fast, you get an accumulation of deleterious mutations.

Point mutations occur as described above, UV light can cause mutations, and chemicals can cause mutations (chemicals that look like nucleotides). Only one of two DNAs are mutated, so only 50% of descendants will carry a mutation ( 25% for humans because of error correction).

A mutation must be passed to the subsequent generations to have any impact on evolution. In bacteria, all mutations are passed. In eukaryotes, only mutation in the germline cells – most of our cells are somatic (not reproductive) and will not pass their information to the next generation. **Cancer** is an example of mutation in the somatic cells – not to be confused with the genetic predisposition to cancer, which is passed to subsequent generations.

**NOTE** See slide on silent and missense/nonsense mutations. Most mutations are the latter two.

Mutations become **substitutions** when a mutation spreads such that it becomes the "new sequence" in a whole population (fixation). This is through the selection process, and not all mutation become substitutions. From a bioinformatics perspective, substitutions are what is referred to here.

In absence of selection, we are thus expected a certain proportion of silent and non-silent substitutions. A statistically significant excess of one indicate negative or positive selection.

One reason why we cannot place a root in the tree of life is because the genetic information can decay over time. Our ability to determine distances decreases with time. Identical proteins may be encoded by significantly different sequences and species tend to have bias in codon usage. DNA sequences better resolve distances between sequences in a short time scale, protein sequences are better for more distant sequences (more characters), and finally, protein 3D structures are the most persistent signal that can be used to measure distance between "sequences". However, our ability to play with 3D structures vs. strings is much different.

Molecular Evolution continued on next page...

## Gene Duplication

**Gene Duplication** refers to the pasting of DNA back in itself. The mechanism for it involve unequal crossover where two DNA strands are close to each other and the repair mechanism cuts and fuses genes by accident. Chimera genes are more common than a simple fusing and twisting of pieces of chromatids. There is also the possibility of two chromosomes with no affinity to crossover by accident.

Whole genome duplication (polyploidy) has been documented, whole chromosomes (polysomy), gene duplication, and parts of a gene (recombination) can be duplicated. Most of the time, genes or parts of a gene are duplicated (shuffling and pasting). The effects of this on the genome is bringing in of redundancy, divergence, and diversity.

This redundancy is useful. *Pseudogenes* refer to genes that were duplicated that are not functional as it is a duplicate. They will both do a number of functions, but one of them may get less efficient at one of them where the other will be better at it. Their expression levels may change over time due to mutations – two homologues are essentially created. At one point in history, these two genes have had a common ancestor.

## Homology

**Homology** refers to the relation from divergence to a common ancestor. For example, humans and monkeys are homologues. The classic example of homology was Darwin's finches. He observed that from island to island, plants were different and finches also had different-shaped beaks to adapt to the different environments.

Homology cannot be observed – unless there is a history of the evolution (virus), it must be inferred from similarity using *characters*. These traits in bioinformatics are quantifiable characters such as sequence similarity and sequence identity. They can be assessed for statistical significance.

**NOTE** Do not mistake homology with **analogy**, which refers to non-homology but still similar things.

Homologs are further refined into three categories:

1. *Paralogs*: When a specimen shares common ancestors but do slightly different things (such as in the pseudogenes example above).
2. *Orthologs*: When a specimen is duplicated and both has a component that still does the same thing after time has passed. Some genes could be incorrectly identified as homologs and the evolutionary distances that result would be incongruent, causing flaws in trees.
3. *Xenologs*: When a gene transfers from one species to another. Genetic modification. Not passed on through vertical descent (horizontal transfer). This completely trashes the tree representation of life.

What occurs as a result are **protein superfamilies** where genes can be grouped into a limited number of categories and three-dimensional shapes.

## Evolution and the Skin Colour of Humans

The following is based on a reading found on http://humanorigins.si.edu/evidence/genetics/skin-color.

One of the arguments given is the fact that all ethnic groups have all come from some variant of dark skin resulting from Melanin, that it must be all humans come from Africa.

Evolution and the Skin Colour of Humans continued on next page...

Another is a discussion of the Mitochondria where there is evidence they possess a reduced genome which could be used as a genetic signature or ”surname”. Therefore, from the maternal side purely, this signature can be used as a way to derive a clean and clear path of maternal ancestry.

Every so often, a mutation may happen in the Mitochondrial DNA where mutations could have occurred after the exodus outside of the African population (in Africa, the M, N, and R sub-types are non-existent for the most part). There is also a great deal more of diversity in the DNA in Africa.

It is also important to note that another driving force of diverging ethnic groups is that populations generally coalesce into taking on mates people that look similar.

A positive selection could be occurring for those with white skin while a negative one could be occurring with those with dark skin. UV rays might not have had a great effect on evolution, but a lack of Vitamin D would. Both of these disappear for the Inuit who get a sufficient amount of Vitamin D and thus have no reason to really change skin colour. A neutral model came out of this where a dark Inuit was not giving a fitness advantage over a lighter one.

The evidence there that genetics is not the main explanation for the difference between ethnic groups is that over 80% of total modern human genetic diversity is across individuals and not across ethnic groups.

Some research can actually show that there are at least seven genes directly affecting skin colour. Darker skinned people have a great deal of cells generating a great deal of Melanin. One of these genes determines how easily these cells can colonize skin with Melanin. Other ones can be viewed on the Wikipedia page for human skin colour.

## Information Theory and DNA

### Open Reading Frames (ORFs)

Since codons are words of 3 characters, finding a START (ATG) and one of three possible STOP codons (TAA,TGA,TAG) constitutes a reading frame. The latter are stop codons because there are no tRNA that can match these STOP codons and so reading stops.

There are *six possible frames* for a protein to be read in – some organisms even manage to overlap genes using two reading frames. How does the mRNA know what the correct strand to copy? A signal sequence is located upstream such that the control sequence is not located on the wrong strand at that location (but at the opposite location on the opposite strand).

Be aware that ORFs are not genes. In bacteria it is OK to assume that all genes are ORFs (but not all ORFs are genes). Find the longest possible sequence beginning with an ATG and terminating by a STOP codon. Real genes will have a regulatory region upstream of ORFs and are typically 100 to 500 codons long. They also tend to have a bias in the composition of their characters.

**NOTE** Information is a deviation from randomness. Sequence logos show a plot of amount of information present.

**Regulatory sequences** promote the translation of a gene – it is a sequence found in 5’ of gene. **Introns** refer to discontinuities in the sequence of a gene where coding DNAs (exons) are interspaced with introns.

The naive way to find ORFs is located on the slides in pseudocode. Simply scanning for STARTS and then next STOP. The problem with that is that while all genes are ORFs, not all ORFs are genes.

Open Reading Frames (ORFs) continued on next page...

What is the probability that a given ORF is random or not? We need to derive a cutoff for long ORFs. $P(stop) = 3/64$ and $P(code) = 61/64$. Let $n = len(s)$ in codons where $s$ is our sequence. Therefore, $P(n) = P(code)^{n-1} \times P(stop)$. This is the probability of seeing a string of $n$ codons between START and STOP. Therefore, in a 500 bp sequence ( 166 codons), there is a 0.03% chance for an accidental ORF.

Given two ORFs of exactly the same length, one is a gene while the other is an ”accident”. To tell them apart:

- There can be a difference in signal.
- This difference will be genome-dependent.
- The classifier needs to learn what is this difference.
- This must be done without assuming the ORF is known.

When given a fragment of sequences, it can be in either 6 frames, or NOT in a gene. The classification is thus a 7-way classification problem.

To address the issue of a training set, we must set our training set as a dataset for which we already know the classification – assumed to be a uniform sample of the data to be classified in the future. These include (1) BLAST results of known genes (prior knowledge), or a (2) sample from long ORFs.

## Markov Processes

**GeneMark** is a **Markov Model** based approach. Does not require the knowledge of gene boundaries (ORF). A computer scientist may prefer the term k-mer approach.

A Markov process is a stochastic model – something that appears random to us. We have something that is linear (time, sequences, etc.) where at some point we have a current observation (current state) and immediately after there is a next state. What happened in the past is inconsequential. The only information you can infer from a Markov process is what is going to be the next state with the current state in mind.

The simplest Markov model is cycling traffic lights – you can infer the next state from knowledge of the current state. This is resting on the idea that there is some real probability associated with what is going to be coming next.

There are many orders of Markov processes. See slide. A first-order Markov process is where we define the current state as a single character (would be very close to the frequencies of the characters). A second-order Markov process involves two characters, and so on. With increasing order you get increasing specificity (but decreasing sensitivity). That means there is a fine balance somewhere here.

In GeneMark training, the principle involves storing the probability of all possible hexamers (6-mers) in each possible frame and in non-coding regions. Classification involves looking at 12-mers. You can go through a training set with all possible 5-mers and look for all possible following characters and count their frequencies. This gives us a notion of likelihood. (See pseudocode).

The notation for all likelihoods can be described as so. Given $P_x(\text{C}|\text{AAATA}) = 0.4$, $P_x$ is the probability/likelihood in frame $x$ for observing C if it is preceded by AAATA.

Bayesian Posterior Probability (that the segment is in frame 3, given sequence s) is:

$$P(3|s) = \frac{P(s|3) \times P(3)}{P(s|nc) \times P(nc) + \sum_{i=1}^{6} P(s|i) \times P(i)}$$

Where $P(3|s)$ is the posterior probability of being in frame 3, $P(s|3)$ is the likelihood, $P(3)$ is the prior probability, $P(s|nc)$ is the likelihood non-coding, and $P(nc)$ is the prior non-coding probability.

Markov Processes continued on next page...

Glimmer 3 is the state-of-the-art interpolated Markov model where training set is based on long ORFs and all k-values from 2 to 8 are considered. It has 95% accuracy.

**NOTE** In a codon, the first two characters are typically the coding ones; the last one does not matter nearly as much. The idea with the GeneMark and Glimmer approaches is that if you're in the right frame, you will be fitting the pattern of a given genome.

## Gene Finding in Eukaryotes

Because of introns, almost none of the genes are ORFs on the genomic sequence, but become ORF later on after post-processing. Journalists like to use the term "junk DNA" – "not in a gene, and we don't know what it does". Some of it is pseudo-random, but because of a great density of introns, the probability of finding a gene begins to shrink.

Genes are scattered on the genome in eukaryotes. There is no direct way to detect introns. Many are created by viruses that integrated themselves into genomes, but this is outside the scope of this course. Signals of slices of genes are drowned by intron signals. But beware, *exons are not mini-genes.* Genes are made up of exons and introns. Exons will typically finish with an $AG$ but the frequency of this is extremely high and so would be an ineffective way to detect them.

But you can assign scores (see sequence logos) and see how often they are donors or acceptors. Scores from **position-specific scoring matrices** (PSSM) can be used to evaluate, rank, and compare a number of possible sites. This score is quite noisy, however.

**GRAIL** uses the interplay between a number of weak factors to predict possible promoters when fed into a neural network. None of the signals themselves are very strong, but when a number of them together are passed through a neutral network, the results can be very beneficial.

See the slides for a *implementation example* which uses a different kind of a Markov model — GeneScout. All possible exons are enumerated (where if the entire gene sequence is an exon it must be a non-intron-possessing bacterial sample). All acceptors (starts of exons) and donors (ends of exons) are set as vertices in a graph and all edges correspond to "codeability" and a path from START to STOP which maximizes this will be the goal. *Those edges from STOP to START represent introns while those from START to STOP correspond to exons.* The constraints would be that:

1. Sum of length of all exons must be a multiple of 3.
2. No stop codons in any exons.
3. START and STOP must be included in the solution.

Recall that the probability of seeing a STOP codon in a random sequence is fairly high. See the slide with the formula for how to find this best path — note there is recursion that occurs in calculating this score.

So for any vertex $V$, there is a score or weight $w(V_i)$ corresponding to an adjacent vertex and the edge which has its own score $w(V_i, V)$. The way these scores comes about is not important – assume there is some way to find them.

Going through all possible scores, you can find the best path by looking for the best score. This "magic path" that goes through some vertices from START to STOP provides the biologically significant donors and acceptors, inferring a gene. Weak signals are discarded and the best signal is kept.

Gene Finding in Eukaryotes continued on next page...

**NOTE** Ten-fold cross-validation is a method used when evaluating a classification method. The training set is broken into 10 chunks where each chunk is successively withdrawn and the other remaining chunks are tested on.

Recall gene transcription and translation. In a cell, as RNA are splicing themselves (the protein has a binding affinity to the coding regions), exactly the gene without intervening introns can come out as a result — this is known as **cDNA** and can be sequenced if extracted from a cell. From this, exons and introns can be mapped – this is thus a way to determine the ”true” path.

**GeneScan** is one of the best methods for gene prediction. It uses a **hidden markov model** to model genomic DNA (looking at the transition between states of intron and exon) and predict non-overlapping genes in all 6 frames. GeneScan does not look ahead. Codability, PSSMs for splice sites are used to the compute the likelihood through the model.

**NOTE** A HIDDEN MARKOV MODEL can be explained by looking at a coin. When flipped, there is a 50% chance to get heads, and a 50% chance to get tails. But say you have a coin that has a 10% chance to give heads and a 90% chance to get tails. Let's consider these coins as *hidden states*. If given a string of results, we can assume they are generated by one of these two coins: the fair coin or the unfair one.

A likelihood could be calculated to see how likely a certain state was used. But how certain can we be only one coin was used? Corresponding likelihoods can be calculated and a *transition probability* is also considered when states are changed. One sequence of transitions will maximize the likelihood. For genetic sequences, each state will have four biases (for each character).

The input to GeneScan is a collection of possible splice sites, promoters, etc. The output is the set of intron-exons boundaries.

## Performance Evaluation

**The Confusion Matrix** refers to the idea that any classification can be organized into one of four kinds of results: true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

**Sensitivity** refers to the proportion of real codons that correctly classified as such. **Specificity** refers to the proportion of predicts codons that are real ORF boundaries (real START codons, for instance). The best way of comparing classifiers is to the use the **F-score** measure, which is the harmonic mean of sensitivity and specificity. Both measures must be good for this to be an attractive score.

## Comparing Sequences

What makes two sequences different? Are conserved characters more important? How far apart are two sequences in evolutionary time? Can we cluster them into meaningful classes? Consider that at this point in history, we may dispose of *more sequence information than actual biological knowledge* at the molecular level.

As an example, we have a given gene, we can retrieve a number of homologs, compare these sequences, and infer a phylogenetic tree that optimally represents the data. On this vein, it is important that the knowledge two sequences are **homologous** allows one to predict what is known about one sequence to the other – at one point, they have a common ancestor.

Sequences can be compared based on their composition:

Comparing Sequences continued on next page. . .

1. Length.
2. Character content.

But these features ignore that a sequence is *more* than a bag of characters. Character-based sequence comparison is much more powerful (rather than looking at it as a bag of them). The analogy here is that of scribes and sacred texts – all books were written by scribes and a certain number of copies were requested. This is how genetic information is translated – RNA polymerase makes copies character-per-character of DNA. But over time, scribes may change words to match the verbs, style, etc. of the current time and these changes may propagate to succeeding generations.

This is the idea behind **alignment**. Homologous sequences need to be properly aligned in order to be compared (using characters) in a positional manner. Homology must be established between respective characters and the resulting operation is called an alignment.

**NOTE** Characters are homologous if they have a common ancestry.

The **hamming distance** between two strings refers to the number of edits that must be done to transform one string into the other. An alignment is a mapping of characters which minimizes the distance between sequences. This is the source for an optimization problem.

We know about the underlying *substitution process* by mutation. Point substitutions cause mismatches between characters, but there is more than just that at play. We can have insertions and deletions, as well. The alignment is thus a description of what you need to do to a sequence to change it into another.

**NOTE** If sequences are left for too long a time, signal decay will prevent meaningful comparison of sequences.

A very simple way to compare sequences is known as a **dot plot**. It is a visualization that is qualitative and is groundwork for quantitative alignments. Two sequences are encoded as a matrix and each matrix entry is a pair of characters between the two sequences that match. Parallel duplicate strands can be inferred as sites of genes where gene duplication occurred. The path crossing the most dots from the upper left corner to the lower right will be a potential alignment.

The issue with dot plots is clutter due to pairing occurring randomly between two sequences. Furthermore, binary treatment of pairing can occur – it's the same character or it's not and no gradation can occur in-between. This is mostly why dot plots are not used in the real world.

This brings us to the **Needleman-Wunch** algorithm, an application of dynamic programming — three-step algorithms which will solve for the best alignment without the requirements of manual intervention. Ultimately, there is a scoring of the matrix of pairwise comparisons, then you accumulate the scores and trace the highest scoring path from top-left to bottom-right of the matrix.

*Scoring pairs of characters* can occur in many ways – identity is essentially like a dot plot, similarity is more powerful for protein sequences (two characters which can reasonably be considered to be equivalent to a certain extent). Getting the scoring wrong means that alternative alignments may score better than the true alignment. An alternative alignment will lead us to believe in erroneous homologous relationships. Other scoring functions include genetic similarity, physical properties, and empirical evidence. The latter will be emphasized here and tends to work the best – derived from substitution observed in trusted alignments.

*Accumulating the matrix* involves starting from the bottom-right corner and applying a formula (see the slide) to each matrix cell where $g$ is a gap penalty and $m > 1$. Essentially, you add to each element the

Comparing Sequences continued on next page. . .

largest number from the row or column to the lower right of each element proceeding right to left, bottom to top.

*Tracing the path* could be a **greedy climb** starting at the bottom-right which can be converted into an alignment. However, you could result in multiple alignments with the same score.

One issue with the Needleman-Wunch algorithm is that it seeks the **global alignment** – an alignment which maximizes the score from the beginning to the end of two sequences. See slide for a situation where the end of a gene inverts.

Another algorithm is the **Smith-Waterman** alignment method – it is a local alignment method and is procedurally similar. Instead, alignment is stopped in regions of accumulated negative scores.

## Information Theory

**Information** is the difference between an observation and what is expected by observing a random arrangement of the data. If there is information, it means there is an *underlying process organizing the data.* Often, a scoring scheme can be derived from this, and information can be quantified and applied to bioinformatics problems.

Telling the difference between noise and signal can be done using a chi-square test, by evaluating uncertainty and information.

Say we're given a multiple sequence alignment. There is an assumption there is a positional homology across each character. If the characters along the strings are not permuted or randomly shuffled, this would typically be information. But how do we quantify this in terms of bits? How many bits of information is there?

Random does *not* mean that everything has an equal probability. Keep this in mind when we discuss uncertainty and information. Even in absence of the theoretical model, the probability will be approximately equal to the frequency. If the selection of people is purely random, one should expect a certain number of them. See slides for statistics.

Furthering on what was said before, treating information as a quantity is done by considering it as a *loss of uncertainty.* Random signal is just noise, but if there is some level of structure, it deviates from random date and that deviation is called information. It is quantified from 0 to some maximum value where the maximum is dependent on the system and its entropy. A primer on Information Theory in Biology can be found at `www.lecb.ncifcrf.gob/~toms/paper/primer/`.

Extending this to genetic sequences involves looking at an alphabet of characters with 4 possible values – this involves using 2 bits if we assume all characters are *equiprobable.* Using different bases involves different units of information (see slide). **Suprisal** is a value that can be calculated which should be defined before looking at uncertainties.

$$u_i = -\log_2 P(i)$$

**Uncertainty** is simply the mean suprisal over all possibilities. Intuitively, the more surprises there are, the less certain we are of observing a given outcome.

$$H = -\sum_{i=1}^{N} P(i) \times \log P(i)$$

Information Theory continued on next page. . .

Where $H$ is commonly known as **entropy**. Uncertainty is maximal when *the probability of all states, or characters, is equal*. There would be no point in using a loaded dice otherwise. Information can be thus quantified where $g$ refers to ground state:

$$R = H_g - H_o$$
$$H_g = -\sum_{i=1}^{N} P(i)^{all} \times \log P(i)^{all}$$
$$H_o = -\sum_{i=1}^{N} P(i)^{obs} \times \log P(i)^{obs}$$

**NOTE** When a promoter is found while translating a DNA sequence, a set of proteins will bind to the promoter region and allow a setup for the RNA polymerase to start translating.

In the Sequence LOGO on the next slide, note the excessive amount of information at every other character. This is because of the spiral and double helix structure of DNA where one strand and every other nucleotide is facing the RNA polymerase while transcription is occurring. The other nucleotides in-between are not nearly as important. Furthermore, as is, sequence LOGOs are phylogenetically oblivious. This could be corrected by considering the phylogenetic distance of the sequences.

Another application for information theory is in the specificity of protein/DNA binding. Most DNA binding proteins make contact to the edges of bases to achieve specificity. The binding site accordingly must main some level of conservation to be a suitable binding side (where conservation means preserving information over being a random sequence). This information is the sum of the information of each site in the alignment that are part of a binding site.

To be unique in a genome, a binding site must thus contain a certain quantity of information. This information must be equal or larger than the suprisal of finding a unique site in a genome. This value is 22.9 bits for *E. coli.*. If a genome doubles in size, a whole bit should be added to make sure that the larger genome does not contain fortuitous bindings sites. Genome site variations minimally impact the specificity of individual binding sites.

**NOTE** The more information you have contained in a binding site, the more that has to be maintained and therefore the more vulnerable it is to mutation.

## Similarity Searches

There are problems which grow up so fast that no money in the world would buy a computer powerful enough to solve them. Taking a short sequence that is sequenced from an obscure organism and comparing it to everything in GENEBANK would be one of them if we would use Smith-Waterman.

However, ”BLASTing” a sequence takes hardly under 1 minute on busy days. How is this possible?

FLASTA is a heuristic method. It will find a good solution faster than a complete method, but it is not guaranteed to find the *best* solution. Essentially what happens is both sequences are broken into *k-tuples*. For protein alphabets, k is around 3 while with nucleotides it is around 11. Small k's will make FASTA more sensitive but slower, while large ones will be faster but requires longer stretches to be exactly the same. Computing a *k-tuple* score means getting the pairwise log odds score for each matched character in two

Similarity Searches continued on next page...

strings of length k. Therefore, similarity scores are found using a scoring matrix such as BLOSUM62 or PAM250. Using DP, it finds which of the best 10 segments give the best score over the length of matching regions.

What results is a tentative alignment after applying a joining procedure. Using **Banded Smith-Waterman**, the matrix is scored and accumulated, but only on a band that is centred on the highest scoring segment in the path. So, only a certain range of the subset of the solution is treated – the assumption made is that things inside this range *must* belong to the alignment. This turns the $n^2$-time algorithm into an $n$-time algorithm because this is linear with the query.

The cost of doing this heuristic however is that small regions will be cut off because of this linear approximation. Furthermore, longer *k-tuples* will miss most distantly related subsequences. The alignment score also has a bias for rare amino acids. Picking the 10 most similar is another parameter where a mistake could crop up giving an incorrect answer. Should we worry about this? Well, it still beats not getting any results on the same day!

**BLAST** is a similar procedure. For each position of the query, a list of *k-tuples* are generated which have a similarity larger than $T$ (neighbours list). Matches are found on the database to either of the neighbours for all positions. All hits are extended until overall similarity scores fall below a threshold (*High Scoring Pairs* HSP).

The first step is enumerating every possible *k-word* for the query. This is identical to the FASTA procedure. Where it changes is what follows. For each *k-mer*, you enumerate every possible neighbouring words of length k that score over T in a pairwise alignment. Scores are found in a scoring matrix (BLOSUM62, PAM200 – they provide different purposes).

The next step involves performing extract string matches for each word in the collection of neighbour *k-mers*. After this, for each string match between a neighbour and a database sequence, an alignment extension is performed until the alignment score falls under a certain threshold. This creates a larger chunk of alignment and these become seeds for creating a more accurate alignment. All alignments which score above an empirically determined threshold are filtered out.

Finally, the significance of getting an alignment score $S$ is evaluated for a query of length $n$. The distribution of Smith-Waterman alignment scores has been demonstrated to fit an extreme value distribution. The *extreme value distribution* has a fatter right tail than a normal distribution – different strategies are used to define this distribution.

The relationship between the score $S$ and the *p-value* is:

$$p(S) = 1 - e^{-Kmne^{-\lambda S}}$$

$K$ and $\lambda$ depend on the similarity matrix, $m$ and $n$ are the size of the compared sequences, and $S$ is the score we are trying to determine whether is significant.

BLAST usually expresses this as an E-value.

$$E = Kmne^{-\lambda S}$$

Intuitively, the E-value is the number of spurious matches, *expected to be found on average*, with an equal or better score than the match reported. You want this to be as small as possible.

Ungapped hits mean that one similar subsequence with an insertion will return *two* HSPs. NCBI-BLAST2 attempts to connect the hits using a **gapped alignment procedure**. However, if two hits are within a

Similarity Searches continued on next page...

certain distance $A$, they will be consolidated. A Smith-Waterman is performed, but only for as long as the alignment score isn't dropping below a threshold. Even more computation complexity is diminished here – because you know there is a pair, you do not set a band width but compute the score until you fall off a certain threshold.

## Scoring Schemes

The theoretical and empirical methods for scoring pairs of alignment characters are numerous. Not all substitutions are observed with an equal probability. **Substitution matrices** for nucleotides usually are concerned with transitions and transversions.

The **Genetic code matrix** is a matrix that relates amino acids to each other by the number of DNA bases needed to change to observe a substitution. It assumes that all DNA substitutions are independent and thus that the more substitutions required, the less probable an AA substitution will be observed. This *ignores* the role of selection, though.

On the neutralist model, once a protein has attained a certain level of functionality, most mutations are either deleterious or neutral. So based on a knowledge-based method we can count observed substitutions over a large collection of alignments and derive log-odd ratios for each substitution.

If we think of loaded dice, we can consider loaded dice as an example of evolutionary bias and fair dice as frequencies of any two characters. Intuitively, very rare amino acids are more likely to be evolutionary related than pulled out of a bag. The **log-odd ratio** is used because very small or large numbers tend to lose precision. Additions are, additionally, faster to perform.

The **PAM** matrix is a point-accepted mutation matrix. The principle is that the mutation observed in highly similar (1% identity) are going to be "neutral". Therefore, this is an assumption to quantify character equivalence in single-character substitutions. The **mutability** is the frequency at which a substitution was observed.

We should note that the probability of observed substitutions change over time. Qualitatively, this means that less likely substitutions are getting increasingly more probably observed over a long period of evolutionary time. Just more likely due to accumulation of time. PAM assumes this. Therefore, higher PAM values are intended to model distantly related sequences.

The **BLOSUM** matrices were built using substitution counts in *blocks* of multiple sequence alignment. Blocks are segments of multiple sequence alignments for which the author were confident the alignment was correct.