PDHonline Course K110 (4 PDH)

Bioinformatics

Instructor: Warren T. Jones, Ph.D., PE

2012

PDH Online | PDH Center

5272 Meadow Estates Drive
Fairfax, VA 22030-6658
Phone & Fax: 703-988-0088
www.PDHonline.org
www.PDHcenter.com

An Approved Continuing Education Provider

Bioinformatics

Warren T. Jones, Ph.D., P.E.

Course Content

Module #1: Introduction

Motivation

Bioinformatics, what's in it for an engineer, particularly one with little background in biology? There are differing opinions on whether the opportunity is real in this new field for this type of person. Let me illustrate with an example from my own experience.

Several years ago it was my privilege to be the organizer of a tutorial session on bioinformatics at a major international conference. The leader of the tutorial was a highly qualified computer scientist who had recently completed a one year leave dedicated to building expertise in the molecular biology field. During the initial part of his tutorial, he cited the importance of deep expertise in both the problem domain of biology and the methodology domain of computational techniques. During the break, one of the attendees informally challenged the tutorial leader's premise with the following story. His bioinformatics team brought a new member on board from Russia who was an expert in a particular optimization methodology that was needed to solve an aspect of an important bioinformatics problem. The team described the problem to the new team member as an abstract optimization problem with associated parameters. After working on the problem for a time, he brought the solution and explained it to the team. At the end of the discussion, he asked, "By the way, what is DNA?"

The point was very clear. It is certainly possible to form effective teams in which all members are not equally expert in the problem domain at hand. One may argue that the above story is an extreme case, but it does serve to illustrate the nature of the team opportunities in bioinformatics. In more recent years, this opportunity is likely even more apparent in the modeling and simulation methodologies of systems biology, which now are beginning to build on the results of bioinformatics data analysis.

Molecular Biology Concepts

DNA (deoxyribonucleic acid) is a linear macromolecule made up of chemical components called nucleotides or bases. This DNA contains the information code for the

development of all living organisms and is made up of only four bases which are designated with the letters A, T, G and C. These bases can occur in any order and that order is a blueprint for its function. In the double strand helix structure, bases pair in specific ways: A to T and G to C. The entire DNA sequence for a given organism is called its genome. The human genome contains approximately 3×10^9 bases. A gene is a short segment of DNA bases with a specific function. The process of transcription of DNA to RNA together with the process of translation from RNA to protein has come to be known as the Molecular Biology Central Dogma.

Molecular Biology's Central Dogma

Through a process of transcription, the double stranded DNA produces single stranded polynucleotide RNA (ribonucleic acid).

RNA through a process of translation produces amino acid sequences that make a protein.

DNA → RNA → Protein

The RNA alphabet differs from DNA in only one character.

RNA Alphabet	DNA Alphabet
A	A
U	T
G	G
C	C

Proteins are made up of 20 component amino acids which can also occur in any order. Similarly, this primary structure determines its function and how the protein molecule folds. Each of the amino acids have different side chains which have different chemical properties and the three dimensional folding associated with these side chains, producing secondary and tertiary structures, is important to the function of the protein. The code that translates the RNA into protein is called the genetic code. In this code, three bases code for each amino acid in a protein sequence. These triples are called codons. It is interesting to note that a triple character code is the smallest combination which has the

capacity to produce a code for the 20 amino acids. A one character code produces four characters (4^1), a two character code produces 16 characters (4^2), and a three character code gives 64 characters (4^3). The genetic code translation table is given below. Each amino acid code in the center section of the table is the result of translation of the triple of bases defined by the first character in the left column, the middle character along the top and the third character in the right column.

Genetic Code

Beginning of Codon	Middle of Codon U	C	A	G	End of Codon
U	phe (UUU)	ser	tyr	cys	U
	phe	ser	tyr	cys	C
	leu	ser	termination	termination	A
	leu	ser	termination	trp	G
C	leu	pro	his	arg	U
	leu	pro	his	arg	C
	leu	pro	gln	arg	A
	leu	pro	gln	arg	A
	leu	pro	gln	arg	G
A	ile	thr	asn	ser	U
	ile	thr	asn	ser	C
	ile	thr	lys	arg	A
	met (and initiation)	thr	lys	arg	G
G	val	ala	asp	gly	U
	val	ala	asp	gly	C
	val	ala	glu	gly	A
	val	ala	glu	gly	G

Translation for a gene is initiatied in what is called an open reading frame (ORF) that begins with a start codon (see codon AUG in the table), proceeding through translation of a series of codons to corresponding amino acid components of a protein, and terminating in a stop codon (one of UAA, UAG or UGA). Organisms can be categorized into prokaryotic (bacteria) and eukaryotic (humans and others with complex genomes) groups. In the case of eukaryotic organisms, a splicing operation takes place after transcription and prior to translation in which sometimes large sections called introns are

spliced out and the adjacent exons retained and rejoined prior to translation. Initially introns were referred to as “junk DNA” because of their noncoding status. The task of gene prediction in an organism is one of identifying these ORF’s

The principle that sequence and structural homology (similarity) between molecules can be used to infer function, underlies many of the important bioinformatics techniques.

Module #2 Sequence Alignment

Pair-Wise Alignment

The alignment of sequences of DNA and proteins are foundational concepts of bioinformatics. The ability to perform rapid automated comparisons of sequences facilitates a variety of tasks such as gene function determination, for example. Not surprisingly, many biological functions are the same or very similar across organisms. These similar functions are associated with similar or identical genes. Therefore, with high confidence, one can often learn a great deal about a new organism's genome by finding these "conserved" genes in databases of other organisms by alignment algorithms, which are designed to provide a measure of similarity between two DNA or protein strings. Similarity is quantified as an alignment score with the highest score representing the best alignment. Global alignment is the task of finding the highest score between two complete strings, whereas local alignment is determining the highest score for a substring.

To illustrate, consider the following example of two DNA sequences:

X1: ATTCGGCATTCAGTGCTA
X2: ATTCGGCATTCAGTGCTA

This is the ideal case where two sequences of length 18 are identical. If our scoring assigned a value of 1 for each of the matched pairs, then the alignment score would be 18. Now consider a case in which some of the character pairs are not aligned.

X2: ATTCGGCATTCACTGCTA
X3: ATTCGGCATTCAGCTA

Here the score would be 12, since only 12 pairs are aligned. However, on closer inspection, one can observe that the last four characters are identical. By shifting these four characters in X3 to the right and by adding two gaps, we have the following:

X4: ATTCGGCATTCACTGCTA
X5: ATTCGGCATTCA_ _GCTA

Using the same scoring system, the score is now 16. However, since we want scoring systems that are biased toward alignments with small numbers of gaps, we include a gap penalty in the scoring system. Assuming a gap penalty of -1 for each gap, the score of 16 now becomes 14.

There are many scoring systems and all are based on the statistics of evolutionary events of deletions and insertions which cause the kind of mismatches we have seen in strings X4 and X5. Scoring matrices have been developed for both DNA and proteins based on

these biological statistics and are incorporated into alignment programs. Given all this natural variation, there is a large number of possible alignments for a given pair of strings. The goal is to find the optimal alignment consistent with the constraints of the scoring system chosen. Finding this optimal by computing all possible alignments and taking the one with the largest score is impractical, so automated techniques have been developed. Almost all of these automated techniques include some form of dynamic programming. The upside is that it provides a rigorous method which guarantees the optimal alignment. The downside is that it is very slow and memory intensive. Therefore, most alignment software makes use of approximations and heuristics to overcome these limitations. However, given its fundamental nature and importance to bioinformatics, we will discuss an example of global alignment using dynamic programming.

Dynamic programming solves optimization problems by solving subproblems and then combining these partial solutions to arrive at the global solution. The key to the application to sequence alignment is how the alignment problem can be decomposed into component problems. The Needleman-Wunsch algorithm is an example of dynamic programming used to solve the global alignment problem for two sequences. We demonstrate the essential aspects of this algorithm with an example. Consider the alignment problem below involving DNA sequences X6 and X7 below:

X6: GGTAC
X7: GTAG

a scoring matrix for the DNA string components A, C, G and T

	A	C	G	T
A	2	-1	1	-1
C	-1	2	-1	1
G	1	-1	2	-1
T	-1	1	-1	2

Scoring Matrix M

and a gap penalty of -2, find an optimal alignment. Note that, in general, this solution may not be unique.

The dynamic programming solution is calculated in two phases. The first phase is called the scoring phase and includes the computation of the elements of a table F which is of dimension length of string X7 by length of string X6, in this case 4 x 5. Each entry in F contains the partial alignment score for the optimal alignment up to that point. The second phase is a traceback process in F from lower right to upper left which identifies an optimal alignment. The first row and column are initialized with the gap penalties. Prior to these calculations, F appears as follows:

	1	2	3	4	5	6
		G	G	T	A	C
	0	-2	-4	-6	-8	-10
G	-2					
T	-4					
A	-6					
G	-8					

Table F

An alignment is equivalent to a path from the upper left to the lower right entry. A move up in the table represents a gap in the top sequence and a move to the right represents a gap in the left sequence. The first row and column represents a simple shifting of one string or the other and adding gaps. The value in each additional entry is a function of three previously calculated entries, as illustrated with the computation of table entry (2,2).

Entry (2,2) is the maximum value of the following three choices:

1. Upper left diagonal (1,1)
   Add entry (1,1) to the Scoring Matrix M entry M(G,G), representing an alignment of G and G

   0 + (2) = 2

2. Upper entry (1,2)
   Add entry (1,2) to gap penalty -2, representing a gap in the top sequence along the x axis.

   -2 + (-2) = -4

3. Left entry (2,1)
   Add entry (2,1) to gap penalty -2, representing a gap in the sequence along the y axis.

   -2 + (-2) = -4

Since Max(2, -4, -4) = 2, the table entry for (2,2) is 2.

In general, the calculations for each entry can be expressed as

F(i j) = Max{F(i-1, j-1) + M score for characters at table positions row i and column j ,
F(i-1, j) + gap penalty,
F(i, j-1) + gap penalty.

The table with all partial alignment scores is shown below:

		G	G	T	A	C
	0	-2	-4	-6	-8	-10
G	-2	2	0	-2	-4	-6
G	-4	0	4	2	0	-2
A	-6	-2	2	3	4	2
C	-8	-4	0	3	2	6

Completed Table F

The optimal alignment is found by a traceback beginning from the lower right table entry. Each successive table entry in the traceback is found by determining which of the table entry computations led to the value in that table entry. A vertical move along this path represents a gap in the sequence along the top and a horizontal move in the path represents a gap along the sequence on the left side. A diagonal move represents an alignment of the two characters at that position. The traceback path represents the optimal alignment. In this case there is only one traceback path. This is not always the case in general. The traceback in Table F is represented by orange entries and represents the following alignment:

GGTAC
GG_AC

Multiple Sequence Alignment

As in alignment of pairs of sequences, multiple sequence alignment of three or more sequences help define structural and functional domains in families of protein and new members of these families can then be identified by searching databases of these same domains. The pair-wise sequence alignment algorithm can be extended to the alignment of three sequences, but for more than three sequences, approximate methods are required.

One of the most common of these approximate methods is known as progressive alignment in which the most similar sequences are first aligned using dynamic programming and then building the multiple sequence alignment by progressively adding sequences. The major problem with progressive alignment methods is their sensitivity to the degree of relatedness of the initial sequences. Initial alignment errors are propagated through the process to the multiple sequence alignment.

More recently, genetic algorithms have been used successfully as a multiple sequence alignment method. The genetic algorithm is a machine learning technique that is not directly related to biology. See PDHOnline course *Artificial Intelligence: Smart Systems Design* for a general description of how it works. The basic idea is to produce many multiple sequence alignments for a given set of sequences by application of the genetic operators of recombination and mutation in order to find alignments with increasingly

higher scores among the candidate populations of alignments generated. SAGA (http://igs-server.cnrs-mrs.fr/~cnotred/Projects_home_page/saga_home_page.html) is an example software system using this approach. Genetic algorithms are computationally demanding, so it is not surprising that SAGA is slow for more than about 20 sequences.

Hidden Markov Models (HMMs) have also been successfully used for multiple sequence alignment. Prior to this application, they had been very successfully applied to the problem of speech recognition. HMMs are basically stochastic finite state machines. All possible combinations of matches, mismatches and gaps are used to generate a set of sequences. A set of 20 to 100 sequences is used as a training set to produce a model which then can be used to output the most probable multiple sequence alignment. As with other machine learning methods such as neural networks, the training set must be carefully selected to avoid over fitting the data to the model. Two common HMM programs in use are SAM (http://www.cse.ucsd.edu/research/compbio/sam.html) and HMMER (http://hmmer.wustl.edu).

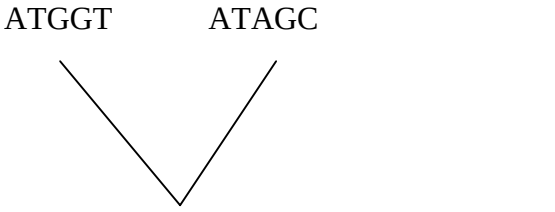
Module # 3 Phylogenetics

A phylogenetic analysis of a family of related DNA or protein sequences is a process to determine how that family might have been produced through evolutionary changes. Given that a multiple sequence alignment method has successfully aligned a group of sequences, phylogenetic analysis takes this alignment and produces a tree structure in which the outer branches represent the aligned sequences. The most similar sequences will be neighboring branches in this phylogenetic tree. The goal of phylogenetic analysis is to discover all of these branching relationships and associated branch lengths. Phylogenetic relationships can assist in the prediction of equivalent function among a family of genes. The challenges of producing this tree are linked to those of sequence alignment in that the more different the sequences, the more difficult the tree is to produce because of the many possible evolutionary paths that could have been followed to produce the variation in the observed sequences.

If the sequences of DNA or proteins from two different organisms are similar, then they are likely to be derived from a common ancestor sequence. A sequence alignment helps to determine which sequence positions have been conserved and which have diverged from a common ancestor. For example, consider the following two sequences. One can readily see that a common ancestor could be inferred by considering the possibility of changes in the dissimilar positions 3 and 5 as indicated in the diagram below.

Sequence 1: ATGGT

Sequence 2: ATAGC



The three primary phylogenetic methods are distance, maximum likelihood and parsimony. If there is clearly some sequence similarity present, then distance methods should be used. If there is strong sequence similarity, maximum parsimony methods should be applied. Finally, maximum likelihood should be used if the similarities are less clear. We illustrate with a distance method and provide discussion of the basic ideas involved in the remaining two.

One of the earliest of the distance methods is known as the unweighted-pair-group method with arithmetic mean (UPGMA). This method requires a distance matrix which is often represented in the following format:

Sequence	**A**	**B**	**C**
B	D_{AB}	-	-
C	D_{AC}	D_{BC}	-
D	D_{AD}	D_{BD}	D_{CD}

D_{AB} can be as simple as the count of non matching characters if A and B are of the same length and there are no gaps. We make this simplifying assumption in the following example to illustrate the method. In the first step, this method clusters the two species with the smallest pair-wise distances into an aggregate group. A new smaller distance matrix is then computed treating the new aggregate group as an entry. A clustering is again computed for the two entries with the smallest pair-wise distances followed by a new distance matrix. This process is continued until all entries are clustered. This final clustering is used to construct the tree. This process is illustrated with an example of five-way alignment of DNA sequences.

A: GTGCTGCACGG CTCAGTATA GCATTTACCC
B: AGGCTGCACGG CTCAGTGCG GTGGTTACCC
C: GTGCTGCACGG CTCGGCGCA GCATTTACCC
D: GTATCACACGA CTCAGCGCA GCATTTGCCC
E: GTATCACATAG CTCAGCGCA GCATTTGCCC

The pair-wise distance matrix for this example is as follows:

Species	**A**	**B**	**C**	**D**
B	8	-	-	-
C	4	8	-	-
D	9	13	7	-
E	10	14	8	3

The smallest pair-wise distance is observed to be D_{DE} which is 3. So we aggregate the two species D and E and compute a new matrix in which the aggregate DE is an entry. The distances between DE and the remaining entries are calculated as the average distance between the remaining species and DE. To illustrate with one example,

$$D_{(DE)A} = ½ (D_{AD} + D_{AE}) = ½ (9 + 10) = 9.5$$

The resulting new matrix is as follows:

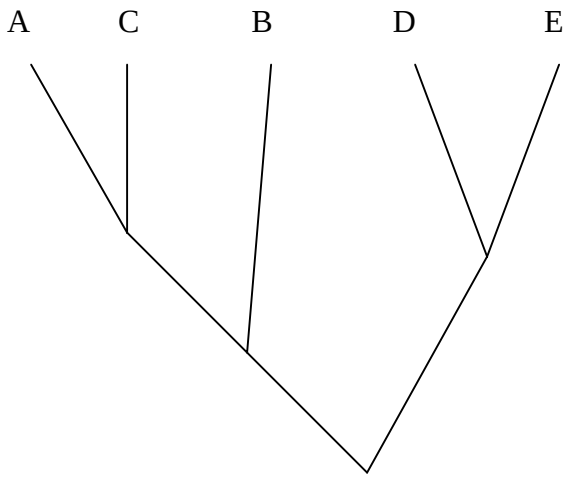
Species	A	B	C
B	8	-	-
C	4	8	-
DE	9.5	13.5	7.5

Now we observe that the smallest distance is $D_{AC} = 4$. We carry out a similar aggregation for AC and obtain the following matrix.

Species	B	AC
AC	6	-
DE	13.5	8.5

In this matrix we see that the smallest distance is $D_{((AC)B)} = 6$ and therefore a clustering of AC and B.

In summary, we have created a clustering of (DE) followed by (AC) and then (AC)B. The resulting complete tree can be represented in standard Newick format as (((AC)B)DE)) or diagrammatically as the following:



Numerical weights for the tree branches are used to represent the relative degree to which the sequences have diverged. In the case of the UPGMA method, the evolutionary rate of change is assumed to be constant across all species. Given this assumption, the nodes internal to the tree are assigned equal distances from each of the species, they give rise to in a binary tree. For example, in the above example, the branches A and C would be each assigned a weight of $D_{AC}/2 = 4/2 = 2$. Other tree construction methods are available for variable evolutionary rates.

Maximum Likelihood

This method brings a probabilistic approach to the problem of phylogenetic analysis. The method considers each individual position in the multiple sequence alignment and all possible trees are considered. Take this fact as a clue to the level of computer resources required. Let's just say extensive. However, this broad approach provides the opportunity to evaluate trees with variable rates of mutation and therefore can explore relationships among more diverse sequences. The tree with the highest aggregate probability is the most likely to represent the true phylogenetic tree.

Maximum Parsimony

This approach predicts the phylogenetic tree that minimizes the number of steps required to produce the observed variation in the aligned sequences. For each aligned position, the set of trees that require the smallest number of evolutionary changes to produce the

observed sequence changes are identified. This process is continued for every aligned position. Finally, the trees that produce the smallest number of changes globally for all sequence positions are computed. As mentioned, this method is best suited for small groups of sequences that are very similar.

Module #4 Bioinformatics Software Development Tools and Technologies

Perl, Java and Python

Although many bioinformatics software tools exist, there will be specific situations where crafting your own solution to the problem of filtering through the data to identify the information you need will be the right approach. When this happens you should be aware that the language Perl (Practical Extraction and Reporting Language) is the most popular language for this purpose. Why? Perl makes it especially easy to detect patterns in string data and the programming effort to do so is much smaller than in other languages such as Java, C or FORTRAN. There is also an extensive collection of Perl modules called bioperl (http://bio.perl.org) that facilitate the development of Perl scripts for bioinformatics applications. Bioperl is open source software that is still under active development.

Java and Python is also used in bioinformatics application development and there are similar open source bioinformatics libraries available at web sites **http://www.biojava.org** and **http://www.biopython.org** respectively.

MATLAB

As many engineers know, the basic MATLAB package is a software tool for numerical computation with matrices and vectors with graphical display capabilities. These capabilities have been extended by the introduction of additional Toolbox packages for various applications, including Bioinformatics. If you are already familiar with MATLAB, moving into bioinformatics using the Bioinformatics Toolbox **(http://www.mathworks.com/products/bioinfo)** would provide an entry point with some familiarity.

The Bioinformatics Toolbox provides an extensible environment in which to explore ideas, prototype new algorithms and build applications. It provides access to data formats, analysis techniques and specialized visualization tools for sequence and microarray analysis. The MATLAB language permits you to customize your algorithms or develop your own. MATLAB can be integrated with commonly used bioinformatics tools such as bioperl. If you opt to use MATLAB, you should be aware that there is a strong preference in the bioinformatics community for open-source tools. However, given that the code developed in MATLAB has an open architecture, it is comparable to an open-source tool.

Microarrays

DNA microarrays (sometimes called gene chips) represent a relatively new technology that has made it possible to quickly explore the patterns of expression of the genes of entire genomes. They can be used to help determine what genes are expressed in a specific cell type of an organism at a specific time and under specific conditions. Frequently comparisons are made between normal and diseased cells. A microarray functions by using the ability of a particular RNA molecule to bind (hybridize) to the DNA from which it originated. Using a computer, the amount of RNA bound to the spots on the microarray can be precisely measured and thus generating a profile of gene expression in the cell. DNA microarrays are small substrates, usually glass microscopic slides, silicon chips or nylon membranes, on which an array of spots of very small fragments of DNA are attached.

The Microarray Markup Language has been developed to provide a standard platform for submitting and analyzing the enormous amounts of microarray expression data being generated by different laboratories. More details on microarray technology are available on the web at **http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html.**

Systems Biology

The availability of DNA, RNA and protein data has given rise to data-driven computer simulations that are designed to move knowledge to the next level and model cellular interactions, providing insight into the next level of organs and tissues. This modeling enterprise is called "systems biology". Modeling teams may include biologists, computer scientists, engineers and mathematicians. Global sharing of these models, in the same spirit as GenBank® for sequence data, is being made possible through a website called BioModels at **http://www.ebi.ac.uk/biomodels** .

Models are developed in a standard model description language called the Systems Biology Markup Language (SBML) (**http://www.sbml.org**) that is XML based. The Systems Biology Workbench (SBW)(**http://sbw.sourceforge.net/the_project.shtml**) has been developed to enable the interaction of different tools. This framework supports tools written in different languages and executing on different platforms. The goal is to facilitate collaboration among developers of systems biology software.

Field Programmable Gate Arrays (FPGA)

With the exponential growth of genomic data and the need for increasingly complex models for systems biology, there is a clear need for continuously increasing levels of computing power for the algorithms and models. One approach is the use of the

increasingly powerful "general purpose supercomputing models" of the day, namely cluster and grid computing. This is certainly happening, and much is being said about it in the scholarly literature and trade journals. However, there is another approach to the computational bottleneck for certain bioinformatics problems.

FPGA's have been used very successfully as speedup strategies for bioinformatics algorithms that contain dynamic programming components and their potential for model components of system biology models is beginning to be realized. For example, there have been reports that FPGA's have performed Monte Carlo-based biochemical network simulations at least two orders of magnitude faster than a PC (p. 19 of *Genome Technology* May 2005). Another upside is that the cost of FPGA's is decreasing because of their extensive use in signal processing.

FPGA's have been commercially available since 1985. An FPGA is a logic chip composed of cells of logic blocks connected with software-configurable interconnections. This ability to reconfigure the hardware in the field is attractive since systems biology models are in an early state of development and require frequent changes. To be able to reprogram the speedup hardware associated with this modeling enterprise is a definite plus. A compiler-like system is needed to provide a language to bridge from the biochemical network model representations to the equivalent network of connections within the FPGA to add a degree of automation to the reconfiguration process. These systems are currently under development.

Module #5: The Life Science Revolution and University Education Opportunities

Industry Convergences

The impact of the Human Genome Project and the technology that made it possible will extend far beyond the medical field. In fact,

"Advances in genetic engineering will not only have dramatic implications for people and society, they will reshape vast sectors of the world economy. The boundaries between many once-distinct businesses, from agribusiness and chemicals to healthcare and pharmaceuticals to energy and computing will blur, and out of their convergence will emerge what promises to be the largest industry in the world: the life-science industry." [4].

An analogy is helpful to understand this amazing prediction. Consider the information technology industry. The digitization of information produced a common binary language for a host of industries such as telecommunications, computing, publishing, television and movies. Sharing a common language sets the stage for a common business and the converging process across these industries that we have witnessed and that continues as cell phone technology becomes more ubiquitous and multifunctional, for example.

A similar convergence is expected in the life sciences. The genetic code is also a type of language, a four letter code of A, T, C and G for DNA and all life as compared to the binary code of 0's and 1's for all digital information. Therefore, any industry that has to do with living organisms or organic compounds has a common language and thus a potential business convergence. Since the genetic code is also a form of information, the information technology industry can be expected to have a major role in these convergence processes. This role is already quite apparent in the form of major industry oriented conferences and expositions such as BioIT World (**http://www.bioitworldexpo.com/live/26/media//news/CC516349**). IT companies such as IBM have formed life sciences divisions to target the perceived major marketing opportunities. IBM's new supercomputer "Blue Gene" is designed specifically for life sciences research.

Nanotechnology and DNA

The remarkable properties of DNA can be used by modern techniques of biotechnology for nonbiological nanotechnology applications [9]. The DNA helix has a diameter of about two nanometers and the span of a full twist is about 3.5 nanometers. Small local areas of DNA have highly specific interactions with other chemicals and therefore can be

used to control the composition of material by acting as a catalyst and making it possible to accomplish the following:

- DNA can be programmed to self assemble into complex configurations.
- DNA can hold molecule-size devices and can also be utilized to construct materials with precise molecular arrangements.
- Movement of DNA nanomachines can be controlled by chemicals or special DNA strands.

DNA is a linear structure. Its success in the nanotechnology world will require the development of techniques for its use in three dimensions as well as in combination with metallic nanoparticles or carbon nanotubes. There have been some initial encouraging successes, such as DNA cubes, but major challenges remain. More detailed information is available at web site **http://seemanlab4.chem.nyu.edu** .

Expected Impact on Medicine

The $1,000 genome may be only about five years away. For this price one can know their own personal genetic code and hence predispositions to disease. The availability of this information will drive the healthcare industry toward personalized medicine and customized drugs.

DNA as digital storage technology

In the quest for greater capacity computer memory, recent research has shown that DNA may have the potential to function as a general purpose memory. This can be done with an organism by carefully defining a set of fixed length DNA sequences that do not exist in the organism but also have all the properties required of functioning DNA. This set of "foreign" sequence segments can be used as a code for the ASCII character set.

The capability of storing and retrieving information in a living host using such a code has been demonstrated [10]. The organisms selected for this demonstration were bacteria since they grow quickly and the embedded information can be inherited quickly and continuously. One of these bacteria is known to survive in extreme environments such as ultraviolet, desiccation, partial vacuum and ionization that exceeds fatal dosage for humans. These are interesting properties indeed for future computer memory applications. One could even speculate on the possibility of storing personal information such as medical data in the cells of one's own body.

University Bioinformatics Education Opportunities

In recent years an increasing number of universities have initiated short courses, specializations, certificate programs and even degree programs in bioinformatics at the B.S., M.S. and Ph.D. levels. A good resource for these educational opportunities is the web site for the International Society for Computational Biology. **http://www.iscb.org/univ_programs/program_board.php.**

Some bioinformatics programs, such as the one at the University of Alabama at Birmingham (UAB) that I helped develop, have collaborations with large medical research centers. UAB has a bioinformatics specialization in the M.S. program in computer science (see **http://www.cis.uab.edu** for more details), a graduate certificate in bioinformatics and courses which can lead to a non-credit continuing education certificate.

Biomedical engineering degree programs are well positioned for the life sciences revolution. However, it is becoming increasingly clear that the projected magnitude and breadth of this new life sciences industry will intersect most if not all other branches of engineering. Therefore, many university engineering degree programs are exploring curriculum options that will include content that will prepare future engineers for some level of career involvement in this new industry. This topic was an agenda item at the 2005 meeting of the Electrical and Computer Engineering Department Heads Association. One approach is to convert laboratory science requirements to required hours in biology. It is not easy to add a new content dimension to engineering curricula.

Module #6: Bioinformatics Databases and Software

Bioinformatics Databases

The Web is an indispensable source of bioinformatics information. Web resources permit the exchange of data as well as software within the global bioinformatics community. GenBank® and the Protein Data Bank (PDB) are two major Internet resources.

GenBank®

The National Center for Biological Information (NCBI) GenBank® provides a globally shared collection of the following:

- DNA sequence data
- PubMed (scientific literature access)
- A taxonomy database
- Access to protein sequence and structure data

GenBank® is part of the International Nucleotide Sequence Database Collaboration, which comprises the DNA DataBank® of Japan (DDBJ), the European Molecular Biology Laboratory (EMBL), and GenBank® at NCBI. These three organizations exchange data on a daily basis. A detailed overview is available at the following web site: http://www.ncbi.nlm.nih.gov/Genbank . GenBank® can be searched in two modes.

1. Search of the annotations associated with each DNA entry using a text-based query and
2. To compare a DNA or protein sequence itself, which one may be studying, to sequences in the database using software called BLAST with the objective of identifying similar sequences.

NCBI GenBank® sequences can be downloaded in one of three formats:

1. FASTA – a simple format that contains little more than sequence data itself.
2. GenBank – a flat file format used early in the history of GenBank®
3. ASN.1 – more recent generic data specification used for all datatypes of sequences, genomes, structure and literature at NCBI.

The NCBI Toolkit for molecular biology software development is based on ASN.1. A tutorial on the Toolkit is available at
http://www.ncbi.nlm.nih.gov/books/bv.fcgi?call=bv.View..ShowTOC&rid=toolkit.TOC

PDB

PDB was initially established by the Brookhaven National Laboratory to archive protein structure data. It is now maintained by the Research Collaboratory for Structural Bioinformatics, a consortium of university and public-agency researchers. PDB is accessible at web site **http://www.rcsb.org** and a comprehensive article [1] about this site is available at web site **http://nar.oxfordjournals.org/cgi/content/full/28/1/235** .

PDB data is available in two formats:

1. The original PDB data format
2. The more recent Macromolecular Crystallographic Information File (mmCIF) format

Bioinformatics Software

Bioinformatics software implements techniques from many disciplines, and developments in bioinformatics are widely distributed in the science and engineering literature. However, there are also a number of conferences and journals specializing in bioinformatics and they are listed at the end of this course. Free bioinformatics software package listings can be found at the websites for PDB (**http://www.rcsb.org**) NCBI (**http://www.ncbi.nlm.nih.gov**) as well as The Institute for Genomics Research (**http://www.tigr.org**). Web implementations are available for many of these packages. In the following table we summarize some of the most popular software packages as a sample of the landscape of bioinformatics software.

Optimization and pattern recognition techniques such as dynamic programming, neural networks and Hidden Markov Models often underlie the algorithms implemented in bioinformatics software packages. BLAST (Basic Local Alignment Search Tool) is the most popular software tool for searching sequence databases. In fact, it is so popular that the term “BLASTing a sequence” has become part of the language of the bioinformatics community.

Function	Software Packages
Identification of similar sequences by pairwise comparison (**sequence alignment**) using inexact heuristic methods that focus on local alignment properties rather than between entire sequences	**BLAST** **http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html**
Identification of protein coding regions (**gene finding**) in DNA. Rely on neural network and dynamic programming techniques.	**GrailEXP** **http://compbio.ornl.gov/grailexp/** **GENSCAN** **http://genes.mit.edu/GENSCAN.html**
Comparing more than two sequences at a time (**multiple sequence alignment**). Uses a process of progressive alignment of pairs of sequences based on phylogenetic tree generation followed by a dynamic programming process operating on branches of this tree	**ClustalW** **http://thr.cit.nih.gov/clustalw/clustalw.html**
Phylogenetic tree analysis. PHYLIP is one of the most popular of many phylogenetic analysis packages. It contains programs that are based on a variety of phylogenetic analysis algorithms including: parsimony searches, distance matrices, branch and bound search, maximum likelihood estimation and neighbor joining.	**PHYLIP** **http://evolution.genetics.washington.edu/phylip.html**
Motif identification involves the identification of any sequence pattern that is predictive of function. MEME (Multiple Expectation Maximum Elicitation) is based on a statistical procedure for predicting missing data values. HMMer (Hidden Markov Model) is basically a probabilistic finite state machine that creates profile	**MEME** **http://meme.sdsc.edu** **HMMer** **http://hmmer.wustl.edu**

HMM structures from sequence alignment.	
Protein secondary structure prediction, sometimes referred to the "holy grail" of computational biology. PHD uses winner-take-all methods applied to the outputs of a group of neural networks that make the prediction based on a combination of local and global sequence characteristics.	**PHD** **http://cubic.bioc.columbia.edu/papers/1996_phd/paper_txt.html**

Bioinformatics Resources

Bioinformatics Related Periodicals, Societies and Job Search Sites

A detailed overview of molecular biology concepts for technical people with no background in biology is accessible at the following website: **http://www.ebi.ac.uk/microarray/biology_intro.html**

IEEE/ACM Transactions on Computational Biology and Bioinformatics
http://www.computer.org/publications/index.htm#Transactions

Bioinformatics
http://bioinformatics.oxfordjournals.org/

Journal of Computational Biology
http://www.liebertpub.com/publication.aspx?pub_id=31&crit=computational%20biology

International Society for Computational Biology
http://www.iscb.org/

Bio-IT World
Bioinformatics commercial applications
http://www.bioitworldexpo.com/live/26/media//news/CC516349

Biohealthmatics
Bioinformatics career opportunities site
http://www.biohealthmatics.com/careers/biocareer.aspx

A more extensive list of journals can be accessed at this web site
http://www.iscb.org/journals.shtml

Bioinformatics Conferences

ACM Symposium on Applied Computing (SAC) – Bioinformatics Track

This Symposium is sponsored by the ACM Special Interest Group on Applied Computing (SIGAPP). It meets alternatively inside and outside the U.S. each year and is managed as a collection of independently administered tracks, each of which covers an application area of computing. I initiated the Bioinformatics Track at SAC several years ago and it has continued to be popular. This is a particularly attractive venue for bioinformatics people since there is also a wide variety of computer applications people in other Tracks with which to initiate collaborations.
http://www.acm.org/sigapp/SAC_Meetings.htm

Beyond Genome: The Future of Medicine
http://www.beyondgenome.com/

Bio-IT World Conference

A gathering place for the commercial side of bioinformatics.
http://www.bioitworldexpo.com/live/26/

Pacific Symposium on Biocomputing

Conference held in Hawaii each year. Proceedings of articles are available on-line.
http://psb.stanford.edu/

RECOMB

Founded as a forum for theoretical advances in molecular biology
http://www.broad.mit.edu/recomb2005/

IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)

http://ieee-cis.org/cibcb2005/

International Conference on Intelligent Systems for Molecular Biology

This conference is the annual meeting for the International Society for Computational Biology
http://www.iscb.org/ismb2005/

Critical Assessment of Microarray Data Analysis (CAMDA)

Held annually at the campus of Duke University.
http://www.camda.duke.edu/camda06

Selected Books and Articles

NOTE: An extensive collection of bioinformatics reference books is available at web site **http://www.iscb.org/bioinformaticsBooks.shtml**

[1] Berman, H. M. , J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne: "The Protein Bank", *Nucleic Acids Research*, 235-242 2000.

[2] Campbell, A. Malcolm and Laurie J. Heyer, *Genomics, Proteomics, and Bioinformatics*, Benjamin Cummings, 2003. (Textbook targeted to undergraduates)

[3] Chen, Yi-Ping Phoebe (Ed.), *Bioinformatics Technologies*, Springer-Verlag Berlin Heidelberg, 2005.

[4] Enriquez, Juan and Ray A. Goldberg, "Transforming Life, Transforming Business: The Life-Science Revolution", *Harvard Business Review*, March-April, 96-104, 2000.

[5] Heath, Lenwood S. and Naren Ramakrishnan, "The Emerging Landscape of Bioinformatics Software Systems", *Computer*, July, 41-45, 2002.

[6] Krane, Dan E. and Michael L. Raymer, *Fundamental Concepts of Bioinformatics*, Benjamin Cummings, 2003. (Textbook targeted to undergraduates)

[7] Krawetz, Stephen A. and David D. Womble, *Introduction to Bioinformatics: A theoretical and Practical Approach*, Humana Press, 2003.

[8] Mount, David W. *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press, 2001.

[9] Seeman, Nadrian C. , "Nanotechnology and the Double Helix", *Scientific American*, 290, 65-75, 2004.

[10] Wong, Pak Chung, Kwong-Kwok Wong and Harlan Foote, "Organic Data Memory Using DNA Approach", *Communications of the ACM*, 46, 95-98, 2003.