A quick guide to the main concepts used in this course
(compiled by Somdutta Dhir and Sándor Pongor)

This tutorial is a brief collection of biological and informatics concepts that you need to understand before you come to the course. We present them so those of you who studied one or the other field will find large parts of this very familiar. What may be new is the way we link these concepts, and in a way this is the first message of this tutorial: knowledge is very often in the new links between known things. And this is the very essence of bioinformatics: we use informatics tools to gather, to analyze and to interlink biological data. In a broad sense, bioinformatics is using computers for understanding biological problems. This course is centered around the analysis of biological sequences, but the main steps of analysis are very similar in aoo other fields of bioinformatics.

Bioinformatics at a glance

Bioionformatics deals with all aspects of biological knowledge, so it is not possible to define or summarize it in a single phrase. The biological objects that we will deal with are

Genes

Proteins

Genomes

These are the traditional objects of molecular biology and in fact bioinformatics was originally born as a part of molecular biology.

We have a few basic tools to represent genes, proteins and genomes

Sequences

Molecular structures

Network descriptions

Scientific publications

Bioinformatics, at a first glance, deals with these data. What we do with these descriptions is

Analyze/Visualize

Compare

Classify

In one word, we try to extract knowledge from these data, we try to understand them. For this aim, we have to store the data in an organized form, in

Biological databases

A bioinformatics database is not only a collection of data. It is a special collection designed to help better understanding. This is best achieved by linking together the associated items in various databases, and providing them with analysis tools so that the user could view certain properties that are not stored in the database. Such a complex assemblies of databases, tools and links are called

Bioinformatics resources

In a way, bioinformatics resources are doing the same as libraries: they store knowledge in an organized and searchable form, but at the same time they are like encyclopedias and handbooks: they point to further books and libraries. In addition, they also have active analysis tools. Bioinformatics was perhaps the first among the other sciences to create such complex tools of analysis.

This is bioinformatics at a glance, and we limited this snapshot to the fundamental static concepts. Many things are missing here, like tools of proteomics, functional genomics, etc. These fields are related to special data collection techniques and to more advanced, dynamic descriptions like the changes of gene expression during the cell cycle. A few of these fields will be mentioned during the course, and also here below in this tutorial.

- - -

Introduction to the subjects

I. Molecular models

We will talk about two kinds of concepts, human and mathematical.

Our main subjects are molecules, cells, various chemical and biological entities. Our knowledge on them is stored in human concepts, such as atoms, bonds, protein-domains, chromosomal regions, and the relations between them (a is close to b, etc.). In summary, we call these models: these are mental representations of real-life objects, knowledge on the real world (and not the real world itself).

We can not give a whole background on all of molecular models, but those interested can look at the following links:

Amino Acids, Proteins
http://pps00.cryst.bbk.ac.uk/course/
(This is the Brikbeck tutorial; it is very good but includes lot of links)

http://www.bmb.uga.edu/wampler/tutorial/prot0.html
(This is a nice tutorial on peptides and protein structure)

Nucleic Acids, DNA, RNA
http://www.cem.msu.edu/~reusch/VirtualText/nucacids.htm

For Basics of Molecular Biology
http://www.coe.uncc.edu/~hhilger/EB_I_F_06/web_links_on_basics_of_molecular_biology.htm

2. Structural descriptions

We have a few basic tools to represent genes, proteins and genomes

Sequences

Molecular structures

Network descriptions

Scientific publications

All these descriptions have a structure. The common structure of all of them is that they all contain entities (subunits) and relations between them. The entities can be atoms, amino acids, protein domains, chromosomes etc. The relations can be chemical bonds, structural vicinities, chemical reactions etc – these are like links between the entities.

Sequences

Structural model: chain-like covalent structure of biomolecules (proteins, DNA, RNA).

Description: series of character

Sequences only contain the basic structural information. Protein sequences are comprised of amino acids, represented in a one-letter alphabet consisting of the 20 characters denoting the 20 natural amino acids. In a sequence we simply write these letters next to each other, starting from the N-terminus (and finishing at the C-terminus.

MALWMRLLPLLALLALWGPDPAAAFVNQHLCGSHLVEALYLVCGERGFFYTPKTRREA
EDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCTSICSLYQLENYCN

Fig: Amino acid sequence for Human Insulin

DNA sequences are the same but we use the 4-letter alphabet of DNA nucleotides (ACGT), RNA sequences are comprised of the 4 RNA nucleotides (ACGU). The sequences are written from the 5' to the 3' direction. Even though DNA is double stranded, we usually write down only the sequence of one of the strands.

CTCGAGGGGCCTAGACATTGCCCTCCAGAGAGAGCACCCAACACCCTCCAGGCTTGA
CCGGCCAGGGTGTCCCCTTCCTACCTTGGAGAGAGCAGCCCCAGGGCATCCTGCAGG
GGGTGCTGGGACACCAGCTGGCCTTCAAGGTCTCTGCCTCCCTCCAGCCACCCCACT
ACACGCTGCTGGGATCCTGGATCTCAGCTCCCTGGCCGACAACACTGGCAAACTCCT
ACTCATCCACGAAGGCCCTCCTGGGCATGGTGGTCCTTCCCAGCCTGGCAGTCTGTT
CCTCACACACCTTGTTAGTGCCCAGCCCCTGAGGTTGCAGCTGGGGGTGTCTCTGAA
GGGCTGTGAGCCCCCAGGAAGCCCTGGGGAAGTGCCTGCCTTGCCTCCCCCCGGCC
CTGCCAGCGCCTGGCTCTGCCCTCCTACCTGGGCTCCCCCCATCCAGCCTCCCTCCC
TACACACTCCTCTCAAGGAGGCACCCATGTCCTCTCCAGCTGCCGGGCCTCAGAGCA
CTGTGGCGTCCTGGGGCAGCCACCGCATGTCCTGCTGTGGCATGGCTCAGGGTGGAA
AGGGCGGAAGGGAGGGGTCCTGCAGATAGCTGGTGCCCACTACCAAACCCGCTCGG

Fig: Portion of DNA sequence of Human gene for preproinsulin, from chromosome 11.

Simplified sequence representations are often used to explain differences between various segments of a molecule. Like domain cartoons, exon-intron-structure of a gene.

Fig: Domain Cartoon

Extended representations add info that we want to explain, e.g. the structure of disulfide bonds.

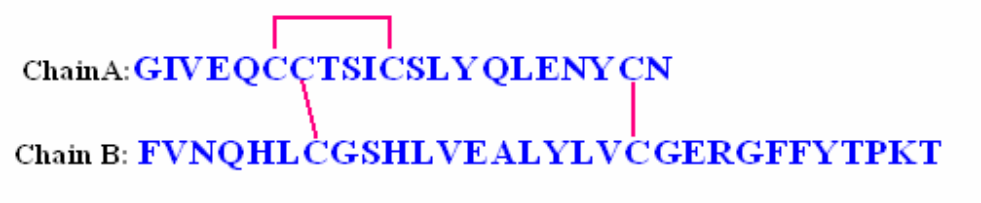

Fig: Disulphide bonds (shown in pink) between the chains of the mature insulin (without the intervening C-peptide that is cleaved off)

3D structural representations

Structural models: full covalent structure in 3D

Description: List of 3D coordinates (explicit) + atomic connectivites (implicit)

Molecular structures, in the first approximation, consist of atoms and chemical bonds. Since they are represented in 3D, each atom will have a position in the 3D space, described by x,y,z coordinates.

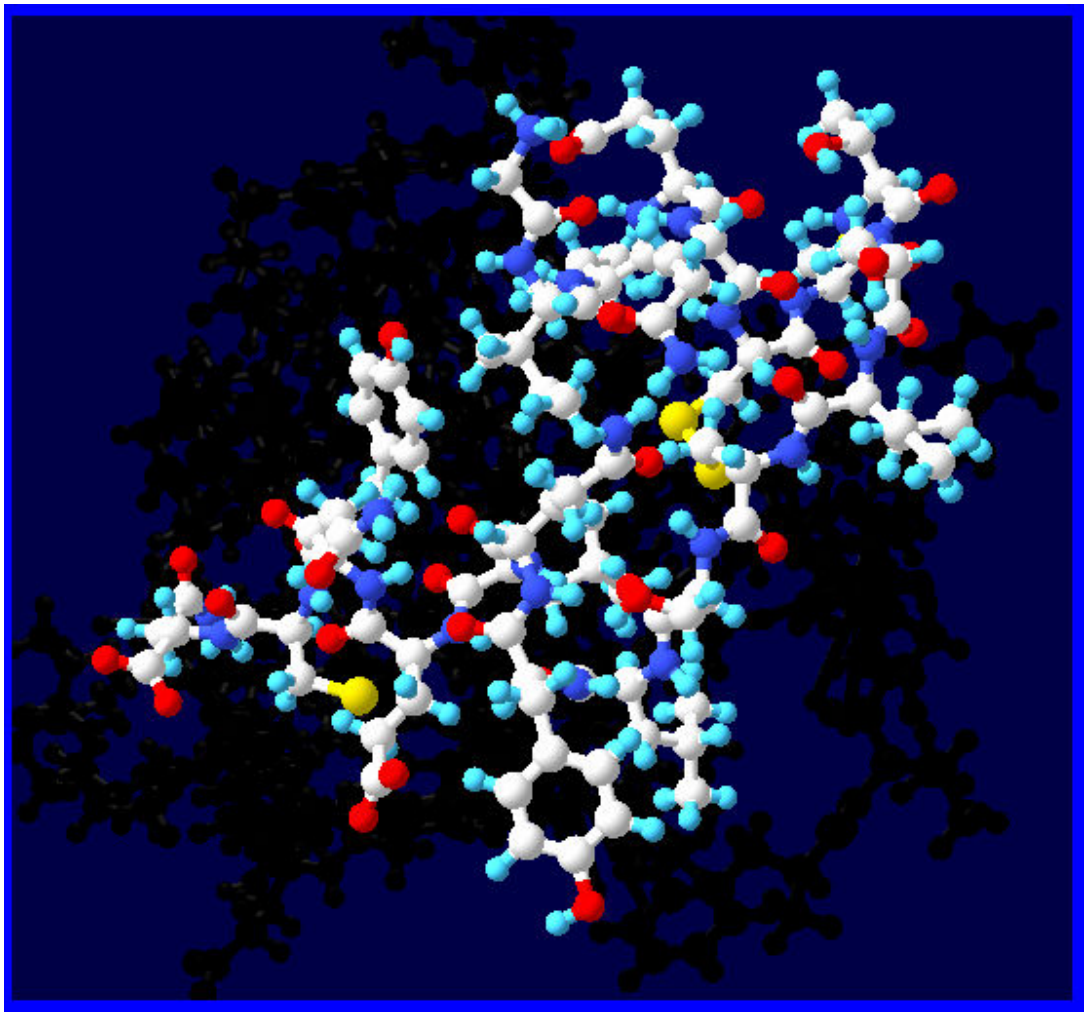

Fig: Atomic structure of Insulin monomer.

In order to represent a molecule in 3D, we need to give the name of all atoms, their x,y,z coordinates, plus a list of connectivities that list which atom is denoted with which other atom. For N atoms, we have a list of N lines that contain the coordinates, and in principle, a connectivity matrix of N x N, that contains the bonds between the atoms. For biological

molecules, we normally store only the coordinates, and the connectivities are stored separately, not as a matrix (that would be too big…), rather than as a form of standardized descriptions for the amino acids (building blocks) that build up a protein or another biomolecules.

The simplest description of a 3D structure is thus a list of lines that contain the names (identifiers) and the coordinates of the atoms, which looks approximately like this:

```
HEADER    HORMONE/GROWTH FACTOR                   19-SEP-02   1MSO
TITLE     T6 HUMAN INSULIN AT 1.0 A RESOLUTION
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: INSULIN A-CHAIN;
SOURCE    MOL_ID: 1;
SOURCE   2 SYNTHETIC: YES;
KEYWDS    T6 CONFORMATION
EXPDTA    X-RAY DIFFRACTION
AUTHOR    G.D.SMITH,W.A.PANGBORN,R.H.BLESSING
REVDAT   1   04-MAR-03 1MSO    0
JRNL        AUTH   G.D.SMITH,W.A.PANGBORN,R.H.BLESSING
REMARK   1
REMARK   2
REMARK   2 RESOLUTION. 1.00 ANGSTROMS.
REMARK   3
REMARK   3 REFINEMENT.
REMARK   3   PROGRAM     : CNS_SOLVE 1.1
REMARK   3   AUTHORS     : BRUNGER,ADAMS,CLORE,DELANO,GROS,GROSSE-
REMARK   3               : KUNSTLEVE,JIANG,KUSZEWSKI,NILGES, PANNU,
DBREF  1MSO A    1    21  SWS    P01308   INS_HUMAN       90    110
DBREF  1MSO C    1    21  SWS    P01308   INS_HUMAN       90    110
SEQRES   1 A   21  GLY ILE VAL GLU GLN CYS CYS THR SER ILE CYS SER LEU
SEQRES   2 A   21  TYR GLN LEU GLU ASN TYR CYS ASN
HET     ZN  D 501       1
FORMUL   5   ZN    2(ZN1 2+)
HELIX    1   1 GLY A    1  SER A    9  1                                   9
HELIX    2   2 SER A   12  ASN A   18  1                                   7
SHEET    2   A 2 PHE D  24  TYR D  26 -1  0  TYR D  26   N  PHE B  24
SSBOND   1 CYS A    6    CYS A   11
SSBOND   2 CYS A    7    CYS B    7
ORIGX1      1.000000  0.000000  0.000000        0.00000
SCALE1      0.012302  0.007103  0.000000        0.00000
SCALE2      0.000000  0.014205  0.000000        0.00000
ATOM      1  N   GLY A   1      -8.785  16.891  14.314  1.00 10.71           N
ATOM      2  CA  GLY A   1      -9.382  16.940  12.938  1.00 11.99           C
ATOM      3  C   GLY A   1      -9.621  15.536  12.448  1.00  9.73           C
ATOM      4  O   GLY A   1      -9.497  14.563  13.228  1.00  9.25           O
ATOM      5 1H   GLY A   1      -8.562  17.871  14.559  1.00 13.45           H
ATOM      6 2H   GLY A   1      -9.544  16.534  14.911  1.00  8.83           H
ATOM      7 3H   GLY A   1      -7.939  16.294  14.400  1.00  9.16           H
ATOM      8 1HA  GLY A   1     -10.303  17.481  12.939  1.00  9.14           H
ATOM      9 2HA  GLY A   1      -8.645  17.402  12.307  1.00  8.02           H
ATOM     10  N   ILE A   2      -9.958  15.423  11.162  1.00  9.38           N
ATOM     11  CA  ILE A   2     -10.280  14.136  10.572  1.00  9.42           C
ATOM     12  C   ILE A   2      -9.116  13.192  10.627  1.00  7.71           C
```

Fig: A portion of the pdb file for the 3D structure of the protein Insulin (1MSO)

The conectivities (i.e. how are the atoms connected in Alanine 131, and is this different from Alanine 212) are not explicitly stored, they are part of the programs that read such files.

Macromolecular structures consist of several hundred to thousand atoms, they a typically too big to be shown in a single picture. So to start with, we need simplified representations, or in other terms: sophisticated visualization.

Typical simplified representations are those showing only the backbones (like "folds") or only the surface. There is a whole set of conventional rules for coloring atoms – for sequences we have no such rules.

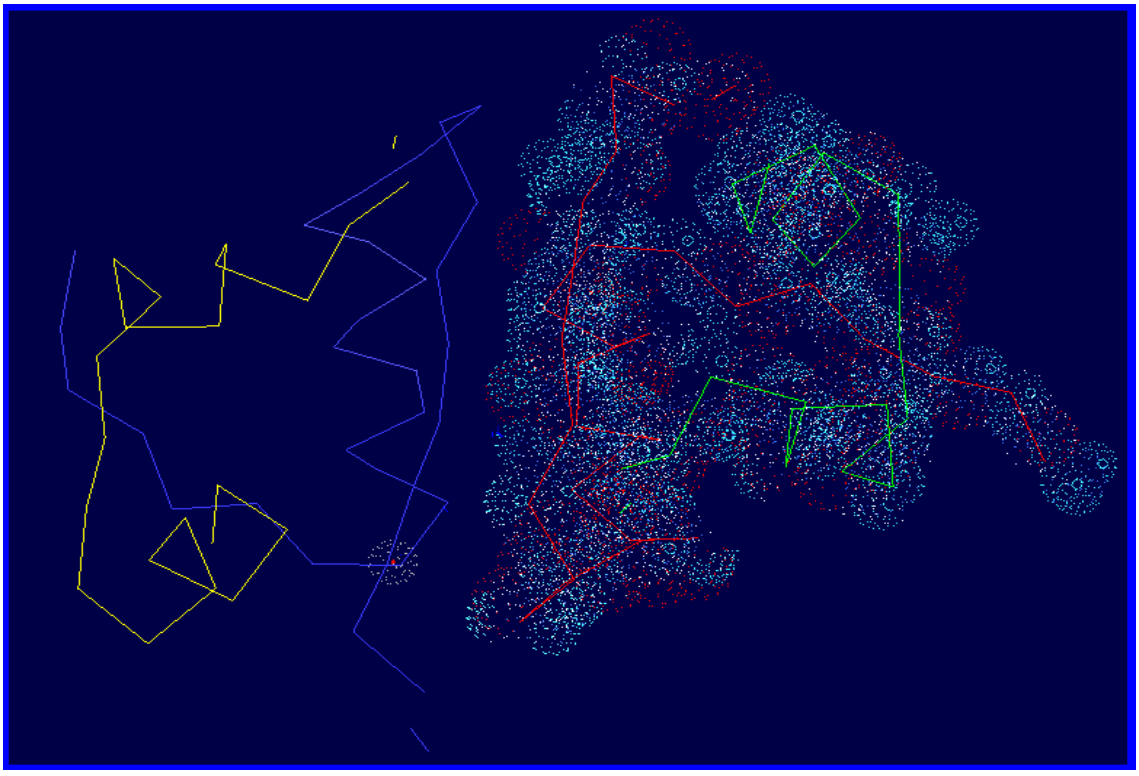

Fig: Backbone representation of Insulin showing the four chains in four different colors. Chains A and B (represented in yellow and blue respectively) show the alpha-carbon backbone, whereas cloud of dots around the chains of C and D (depicted green and red) show the surface representation.

There are also extended 3D representations, typically such are the pictures in textbooks.

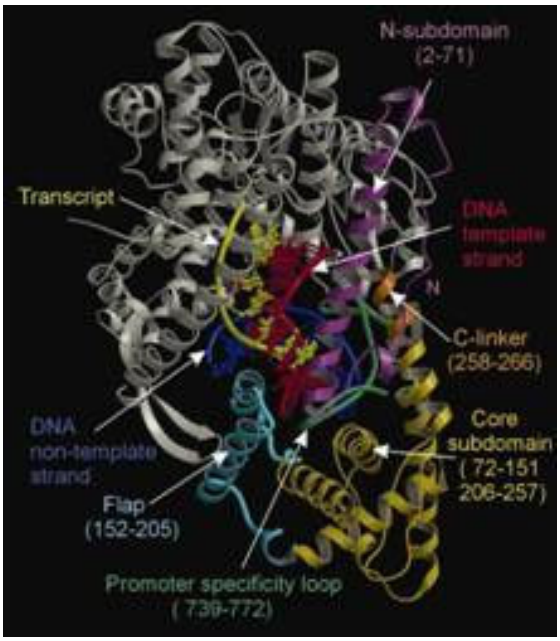

Fig: Detailed structure of T7 RNA Polymerase Elongation

Network representations

Structural model: Ad hoc list of entities and relationships (pathways, regulatory nets, coauthor-networks etc).

Descriptions: any (mathematical) graph representation

In network representations we can show any set of entities (proteins, genes, substrates etc) and any set of relations (structural, functional etc.), as a network lying in the plane of the paper. From the mathematical point of view, such networks are graphs, composed of nodes (points, vertices) and edges (links). Such networks can be directed, for instance the lines are arrows that express that one gene regulates the other one. Or they can be undirected in which a line simply expresses that two nodes (say cells) interact.

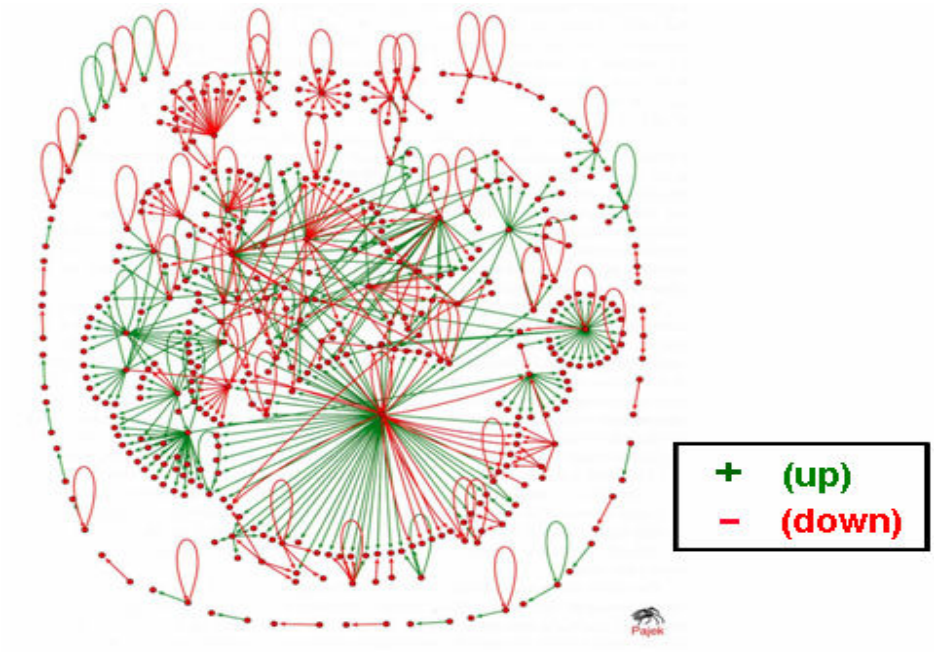

Fig: Gene-regulatory network of *E. coli*

From the visualization's point of view, we have 2D diagrams that can be easily extended with additional information, comments etc.

While sequences and 3D representations are strongly linked to specific fields (molecular and structural biology, respectively) Network representations are more general, and they include the former two as special cases.

Scientific publications, literature abstracts.

Structural model: (?)
Description: Structured text-files

Even though much less standardized than molecular structures, scientific publications are also structured descriptions as they consist of more or less standard parts such as title, abstract, introduction, methods, etc. Of course, there is a great difference, all titles are different and express a different message, while carbon atoms or amino acids are supposed to be identical (by the way, are they?).

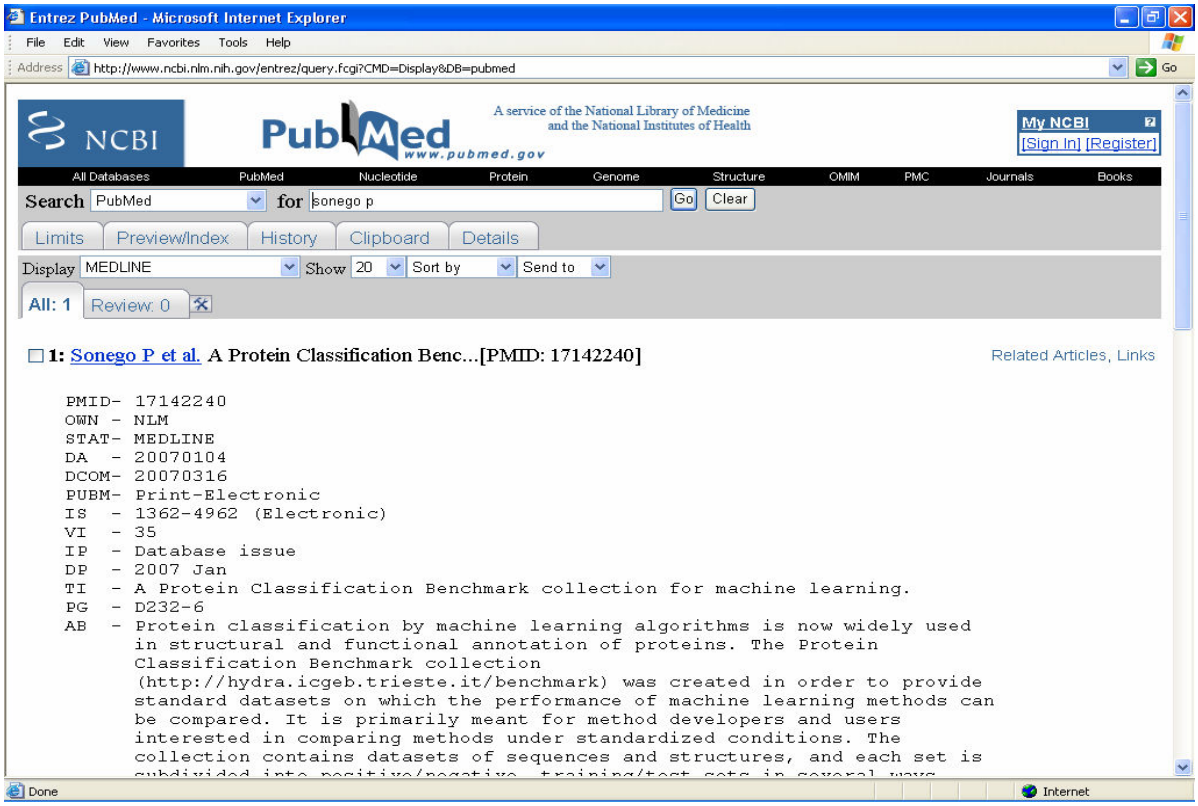

Fig: Example of a Medline Abstract

The real medium of scientific publications is scientific language, which is though more standardized than common or literary language, it is still a natural language. So the analysis of scientific publications is a typical case of natural language processing which is outside our scope. On the other hand there is a strong tendency towards structuring scientific communications so as to make them more computer-readable. The organization of the articles, the standardization of author names, affiliation, and especially the definition and organization of keywords are all clear signs of this tendency. In addition, there is vast challenge in linking molecular data to the literature - text analysis is a perhaps the fastest growing field within bioinformatics.

A note on unstructured and mixed (compositional) descriptions

In bioinformatics everything seems to be structured. What is an unstructured description? Any numerical value, text attribute which is assigned to an object without making assumptions about its internal structure, can be regarded as an unstructured description. If we characterize an apple by its weight, we have such a description. Or we can assign to it a list of features, weight, diameter, color, etc. If we assign numbers only, we get a feature vector. Feature vectors are not "nice" since they often incorporate completely incompatible data. But vectors are easily handled by computers and there are fast and efficient methods (for instance those used in chemical informatics) that can use such data for classification.

There is a simple way to produce vectorial descriptions from structured data. For example, a protein sequence can be turned into a (20-dimensional) vector of amino acid composition, or a (20 x 20 = 400 dimensional) vector of dipeptide compositions. In sequences analysis these are called n-grams, or words. Then, one can work with these vectors, using the tools of standard tools in other fields. These vectors are a mixed representation as they incorporate structural info (an amino acid is a structure) as well as a composition-like, essentially unstructured info (we do not see, where the amino acids are, just how many they are…).

There are many fields where unstructured or compositional descriptions are important, For example, when we say that a region of the genome is GC-rich, or a protein domain is hydrophobic, we use compositional descriptions. Even simpler descriptions are used in some text mining applications: presence/absence of words ("bag of words" approach).

Databases, Ontologies

In the previous sections we talked about concepts (models) and descriptions (sequences, 3D structures etc.). Now we go one (Meta) level higher, we will look at a) Databases that store the descriptions (the contents) and b) onthologies that formally define the descriptions.

Databases

Descriptions are stored in databases, each data item is in a record and each record is divided into fields. In the simplest case, a record is a series of fields, and a database is a database is a series of records. In the context of our bioinformatics course, a typical record corresponds to a protein or to a gene.

A typical record is comprised of two main parts, one is a structural description, such as a sequence or a list of 3D coordinates, and the other one is the so-called annotation part that contains the information in a human or human-readable language.

The information contained in the annotation is partly quite simple: the identifiers of the records, the name of the molecule, etc.

Some parts of this info is quite complex. For example, we have cross-references to other database records, e.g. a protein sequence is cross-referenced to its gene or to its 3D structure

record. We have functional descriptions which are again a type of cross-reference, since it links a protein to other proteins.

Other types of info refer to parts of the sequence. A typical example is a feature-list or feature-table which assigns certain features (such as "signal peptide") to a part of the sequence. A feature-table is thus a duplication of the structure, i.e. a list of labels that can be assigned to various parts of the structure.

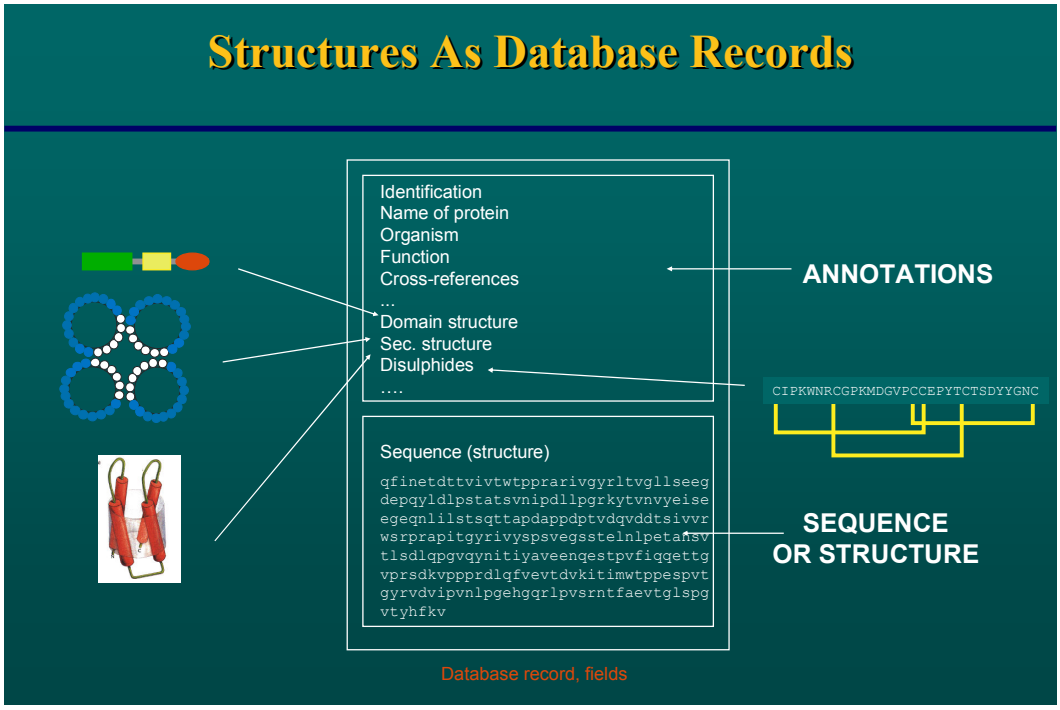

Fig: A cartoon showing the structure of a database record.

It is thus apparent that annotations contain global descriptors (referring to the entire molecule) as well as local descriptors (that refer only to a given part of it).

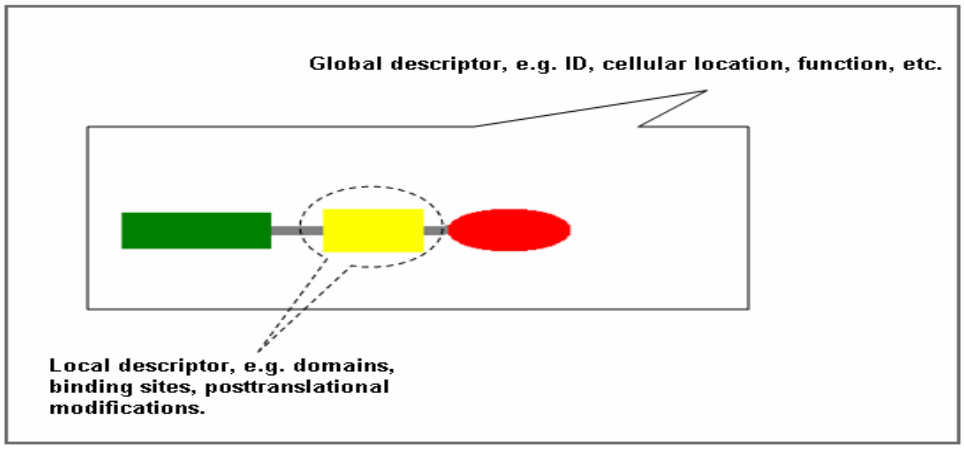

Fig: A cartoon showing Global and Local descriptors

Database design is an important branch of informatics. As compared with other fields, the databases used in bioinformatics are neither too large, nor too complicated (as a comparison, the database of air ticket reservations is accessed and updated by many thousand operators and queried by millions of users).

Ontologies

Database records are generalized structures that are made to store descriptions. Ontologies, on the other hand, are generalized tools to store concepts. Ontologies are created so as to standardize the language of a field. They define the concepts that can be used as well as the relations that are allowed between the concepts. If we describe our database using an ontology, we can use computers to check the integrity of the data and to filter out the logical inconsistencies. Moreover, we can analyze the data in a much deeper way.

The ontologies can be of many types, for example… In bioinformatics we have the Gene Ontology that is an effort to standardize the genes and their descriptors. We have a separate ontology for 3D structures, etc.

A note on systems biology.

Much of current bioinformatics research concentrates on "system biology", a very broad concept that is perhaps more of a paradign than an ordinary field of resarch. The bottom-up approach to sytems biology is data driven, it attampts to make a use of all possible data availble on a biolgical system. The top-down approach, on the other hand tries to build up abstract, wholistic models of orgnism, cells and other biolgoical systems, and tries to establish the logis at this abstract levels.

3. Comparison

Comparing and classifying proteins and genes is perhaps the most fundamental exercise in bioinformatics, which is grounded in a very human ability of establishing similarities between real-life objects. In real life, similarities can be either structural – related to the nature of the object itself – or contextual – related to the environment of the objects. For example, two paintings can be similar because they both show flowers. Or they can be similar in the sense that they are both in the same museum, or painted by the same hand. In bioinformatics, molecules can be either structurally similar – in terms of their sequence, their 3D shape – or functionally similar, since function is by definition understood in the context of a system, such as a pathway, a human cell etc.

Here we are more concerned with the mathematical concepts of comparison.

Comparing unstructured descriptions

The simpler case is the comparison of unstructured descriptions, i.e. comparison of vectors. Vectors are compared in terms of distances, which are numerical values. So one can average

them, make statistics on them, etc., compare one object with a group of objects etc. In fact, the mathematics of comparison is largely based on the concepts developed for vectors.

The **Euclidean distance** between two points $P = (p_1, p_2, ...p_n)$ and $Q = (q_1, q_2, ..., q_n)$ in Euclidean *n*-space, is defined as:

$$D = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$$

Or

$$D = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

A distance is a measure of dissimilarity of difference, i.e. it is zero for identity and nonzero if two objects are different. Sometimes have the properties of metrics, i.e.…

A typical measure of similarity is the dot product of vectors, which is maximal if the vectors are identical and minimal, if they are different.

The dot product also known as inner product of two vectors $A = [a_1, a_2, ..., a_n]$ and $B = [b_1, b_2, ..., b_n]$ is by definition:

$$A.B = a_1 b_1 + a_2 b_2 + ... + a_n b_n \qquad \textbf{Or} \qquad A.B = \sum_{i=1}^{n} a_i b_i$$

Comparing structured descriptions: alignments and their scoring

When we compare unstructured descriptions, such as vectors, we only carry out a simple calculation. When we compare structured descriptions, we first need to align them.

Simply put, alignment means matching the parts of one structure to parts with another structure. In real life, this is instinctive. When we compare cars, we know that the engine of one corresponds to the engine of the other one. When matching molecules, we are not guided by intuition, we need computational procedures that are fast for simple representations such as sequences, but slow for more complex descriptions.

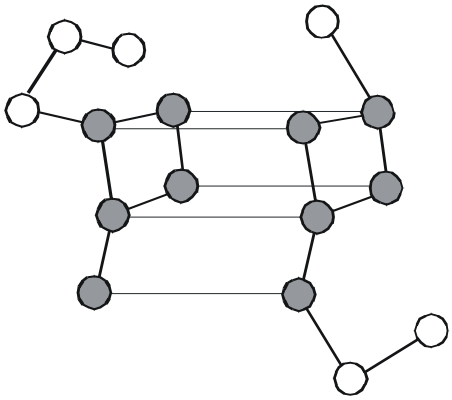

Fig: Alignment means to discover and link the equivalent parts in two structures. This process is instinctive and quasi automated for the human mind. It is computationally feasible only for simple structures, like sequences and graphs.

When calculating a measure of similarity or dissimilarity between sequences, we first need to align them then calculate a score. An alignment looks as follows:

Global Alignment

```
--T--CC-C-AGT--TATGT-CAGGGGACACG-A-GCATGCAGA-GAC
  |   || |   ||   | | | |||     || | | |   | ||||    |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG-T-CAGAT--C
```

Local Alignment

```
                    tccCAGTTATGTCAGgggacacgagcatgcagagac
                       ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

A score can then be calculated by assigning weights to the characters that face each other in the two sequences, and adding them up. We can assign scores for creating a gap in either of the sequences, and we may say that identical characters have a weight of one, non-identical ones have a weight of zero. This is a very simple, but for protein sequences we have more elaborate possibilities. According to a method started by Margaret Dayhoff, the weights are calculated from the number of times one amino acid is found to replace another one in a database of homologous genes. This is determined from a large number of alignments. Given these weights we can assign scores to each alignment and the weights used in practice are defined in such a way that the alignment scores are similarity measures. The first such matrix for amino acids was the PAM matrix, the most popular one used today is the family of BLOSUM matrices).

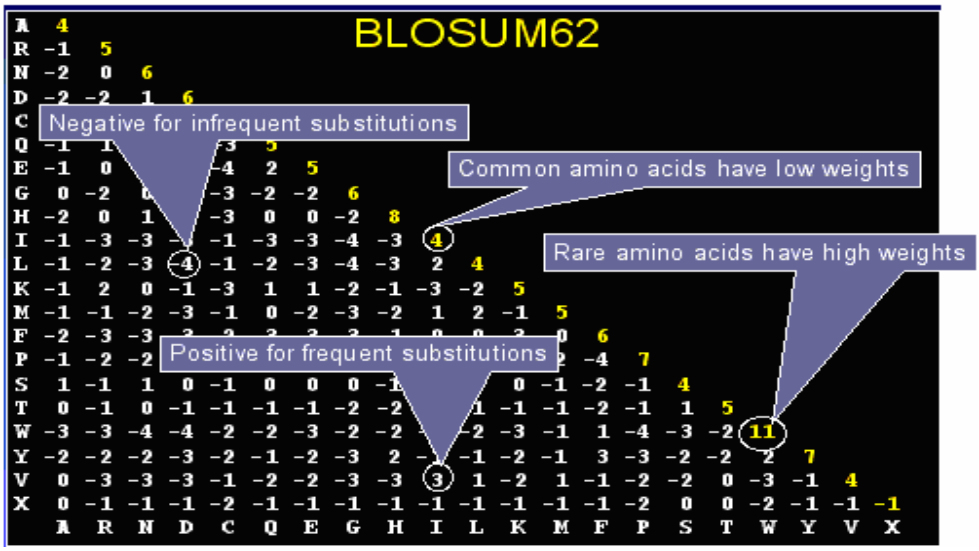

Fig: Example of BLOSUM62

The complications come from the fact that the range of alignment has to be found before we calculate the score. This is not straightforward, and string matching algorithms, such as Smith-Waterman or the popular BLAST algorithm make just that.

4. Group representations: multiple alignments, trees, regular expressions

Structural models: Similarity groups, alignments, trees
Description: distance matrices, regular expressions, frequency matrices, profiles etc.

Representations for object groups are characteristic tools of bioinformatics. Give a set of sequences, we can carry out a multiple alignment by hand that looks as follows:

```
                    B-domain                                    A-domain

IGF1    --------GPETLCGAELVDALQFVCGDRGFYFN       15 ----GIVDECCFRSCDLRRLEMYCA
IGF2    -----AYRPSETLCGGELVDTLQFVCGDRGFYFSRPA     8 ----GIVEECCFRSCDLALLETYCA
INS     -------FVNQHLCGSHLVEALYLVCGERGFFYTPKTRR 33 ----GIVEQCCTSICSLYQLENYCN
RLN2    ----SWMEEVIKLCGRELVRAQIAICGMSTWSKR      106 QLYSALANKCCHVGCTKRSLARFC
INSL3   LGPAPTPEMREKLCGHHFVRALVRVCGGPRWSTEARR    48 AAATNPARYCCLSGCTQQDLLTLCPY
INSL4   --------AELRGCGPRFGKHLLSYCPMPEKTFTTTPGG 58 SGRHRFDPFCCEVICDDGTSVKLCT
INSL5   -------KESVRLCGLEYIRTVIYICASSRWRR        66 ---QDLQTLCCTDGCSMTDLSALC
hINSL6  -RELSDISSARKLCGRYLVKEIEKLCGHANWSQFR     118 ----GYSEKCCLTGCTKEELSIACLPYIDF
rINSL6  ---QEEVTSPTKLCGRDLLVEVIKLCGQNDWSR       110 ----GFADKCCAIGCSKEELAVACLPFVDF
consensus           hCG  hh     hC                      CC   C    h  hC
```

Fig: Multiple alignment of insulin gene family members with human and rat INSL6 over the B- and A-domains

In order to summarize an alignment in a concise way, we can use the tools of regular expressions, e.g.

```
[RK]-x(2)-[DE]-x(3)-Y
```

Fig: This regular expression suggests that the pattern should contain an arginine or lysine residue, followed by two residues of any type, then an aspartic or glutamic acid, followed by 3 residues of any type and a tyrosine residue.

Regular expressions are well known in the computer sciences allow in fact any set of characters at given positions, but they do NOT say what characters occur there more frequently than others. If we need that information, we construct a frequency matrix that contains the occurrence of each letter at each position.

Regular expressions and frequency matrices are especially good for one thing: we can scan a database with them so as to find sequences that are similar to the group. Regular expressions are very fast in this respect but they may detect a large number of unrelated hits (because they allow to match any character at certain position). On the other hand, frequency matrices (in fact their transformed versions) can be used also by fast algorithms similar to BLAST – these searches are much more selective.

The representation of an object group in computer science is a distance matrix (or similarity matrix). We get such a matrix by numerically comparing the objects with each other and then writing a distance or similarity value into a matrix.

	Chimpanzee	Sheep	Rattlesnake	Carp	Snail	Moth	Yeast	Cauliflower	Parsnip
Human	0	10	14	18	29	31	44	44	43
Chimpanzee		10	14	18	29	31	44	44	43
Sheep			20	11	24	27	44	46	46
Rattlesnake				26	28	33	47	45	43
Carp					26	26	44	47	46
Garden snail						28	48	51	50
Tobacco hornworm moth							44	44	41
Baker's yeast (iso-1)								47	47
Cauliflower									13

Fig: Distance Matrix showing degree of identity between the sequences of cytochrome c from a variety of species.

We can use such a matrix to build a phylogenetic tree, which is perhaps the oldest way to a group of biological objects. Of the many methods one is to use a distance matrix and then to successively join the nearest neighbors to each other.

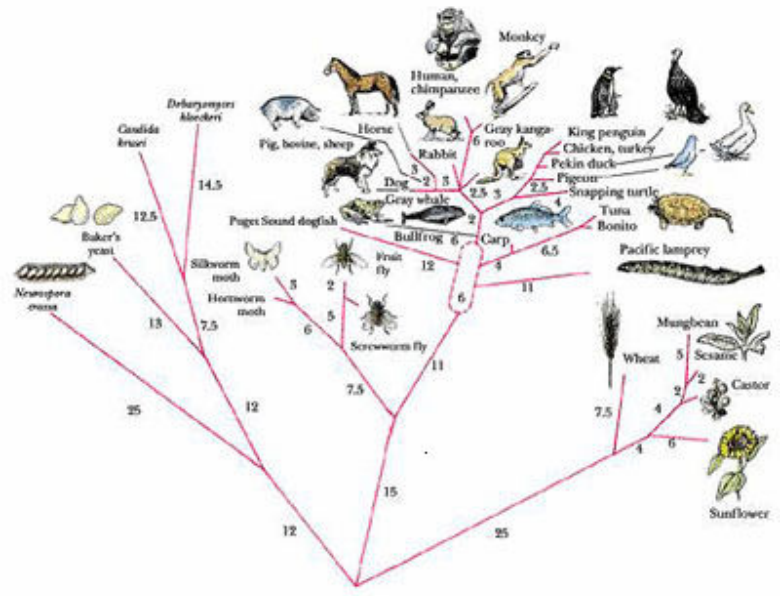

Fig: Phylogenetic tree based on the above distance matrix for cytochrome c

5. Classification

Classification is one of the best studied subjects of computer science, and its basic methods deal with unstructured descriptions (vectors) that can be used in almost any field. Here we only give a brief outline on the example taken from genomics, and we will use structured descriptions (sequences, structures) that are characteristic of bioinformatics.

There are two kinds of classification problems:

i) In unsupervised classification we have a group of unknown objects (e.g. sequences) that we try to group by similarity. We can do that by comparing them with each other and then group them into clusters. This is the area of cluster analysis, a sub-science of its own. Many of its standard techniques are not applicable to very large databases, it is not since very long that wholesale clustering of all proteins can be carried out.

We can compare the resulting clusters with outside information we might gather. For example if proteins of a known family consistently cluster into two groups, we may try to define two subfamilies. Especially if those two groups correspond to groups of phylogenetically distinct species.

ii) Supervised classification is a situation where we already know the classes (from external knowledge or from clustering), and want to determine if new objects, e.g. sequences fall into the one of them. Here we build "classifiers" using the known classes and then evaluate the unknown objects. This is the area of machine learning, and landmark algorithms like neural networks, Hidden Markov Models etc. fall into this category.

In the simplest case we deal with two classes, + and -. Given an unknown query, the response of a classifier is then one of four possibilities:

In the practice of sequence classifications there are 4 general methods to do this classification:

1) Pairwise comparison. Here the classifier if a human operator who looks at the top list and makes a decision whether or not to assign the unknown query to one of the top-ranking classes.

2) Generative models. We can build a model of the protein class. Regular expressions, frequency matrices, profiles, neural networks are all kinds of generative models. Such a model can tell if a new object is member or not. In the simplest case, like with regular expressions, the answer is + or -. For more advanced models we get a numeric value that has to pass a certain threshold so as to be +, otherwise it is -. So we detect members of one class vs. outliers.

In sequence annotation we may have many thousand classes, and we need a classifier for each of them. Luckily, a given protein will only be detected by very few, and then we can assign the protein to the classifier that gives the largest signal.

3) Discriminative models

In the next generation of methods we try to explicitly represent the objects (protein groups) in a multidimensional space, and determine the demarcation line ("decision surface") separating protein groups. The separating lines can be relatively easily computed by turning proteins into vectors, and one of the methods is to use a distance matrix of proteins as a vector representation. This is the principle of kernel methods. There are various methods to calculate separating surfaces, the most popular of them are Support Vector Machines. These again give a + or – answer depending on which side of the decision surface the query is found.

4) Network models

This is the last cry of current fashion greatly inspired by such algorithms as those used by Google. The principle is to use the entire information stored into the entire network of similarities (and not simplify it into class labels or probabilities…). Here we take a query protein, compare it with the database (e.g. using BLAST) and instead of simply looking at the ranking, we start to modifying it using the precoputed similarities between all members of the database.

These precomputed values are the similarity network, and according to the Google principle we can imagine the proteins in the database passing messages through the query. The message is a number characterizing e.g. how many and how similar neighbors they have, etc. In this way, the original ranking can be updated, and in the optimal case, the relevant similarities may be more conspicuous in the top list.

6. Simple programs

There are a few straightforward but general programming principles that are outside the general scope of our course, so we briefly mention them here. Many of the simple plotting tools are available at bioinformatics resource pages such as Swiss-Prot. DNA sequence plotting and visualization tools are available at the ICGEBnet homepage (DNA-tools)

Visualization tools

Visualization of molecules is a field in itself, it consists on mapping (color or other symbolic) features on (theoretical or realistic) structures.

Simple sequence plots

The chainlike structure of macromolecules makes it plausible to picture them as a straight line, and then plot some variable that is characteristic of a sequence position.

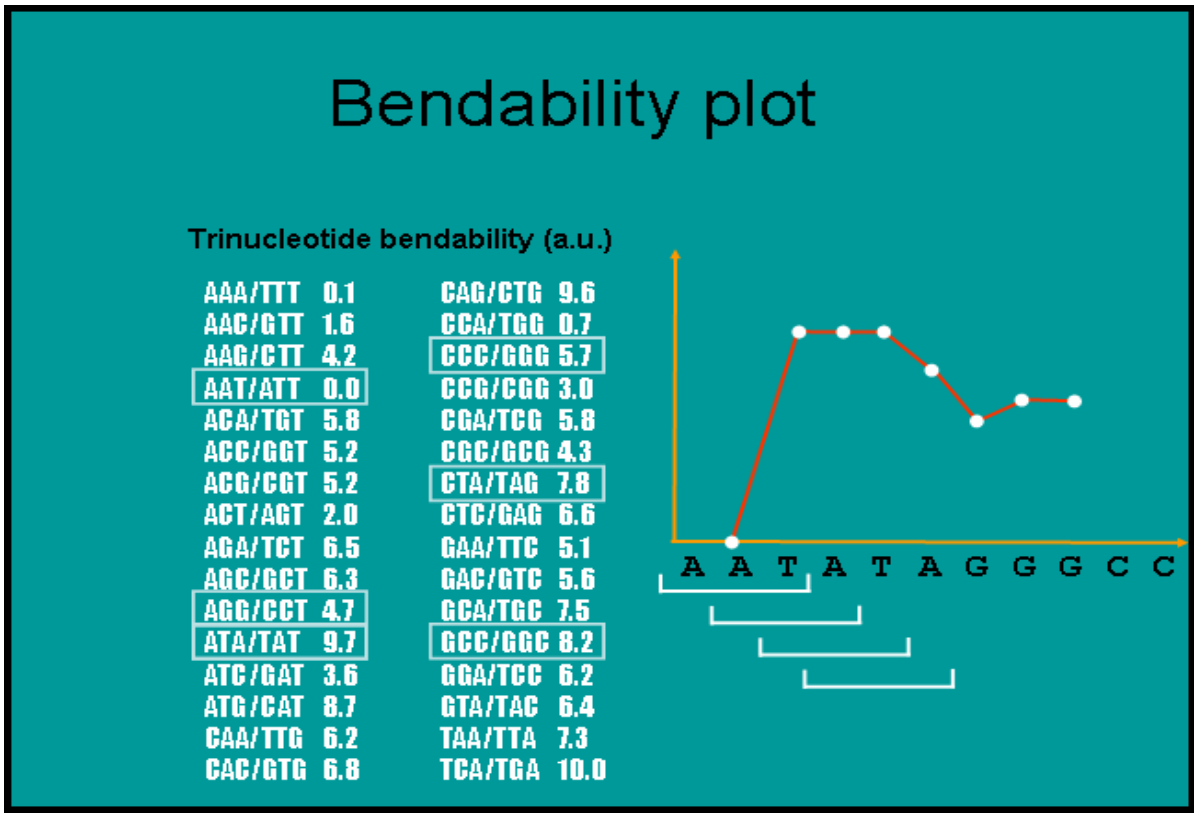

Fig: The DNA bendability plot based on the trinucleotide bendability values.

In protein sequence analysis, we can plot for example hydrophobicity – this is an experimentally determined and scaled estimate of how much an amino acid side-chain prefers a hydrophobic environment, such as a membrane, as opposed to water. Plotting this value along the sequence is a noisy plot that has to be smoothed so that we see peaks and valley. In

membrane-spanning receptors we normally see a series of peaks and valleys that makes it easier to recognize these proteins – one can not easily recognize them from the sequences.

There are two kinds of parameters that we can plot: a) Experimentally or theoretically determined precomputed values that are stored in lookup tables; b) Values computed from the sequence or from the structure. These need to be computed for each sequence we analyze.

Some methods for DNA are listed in the following table:

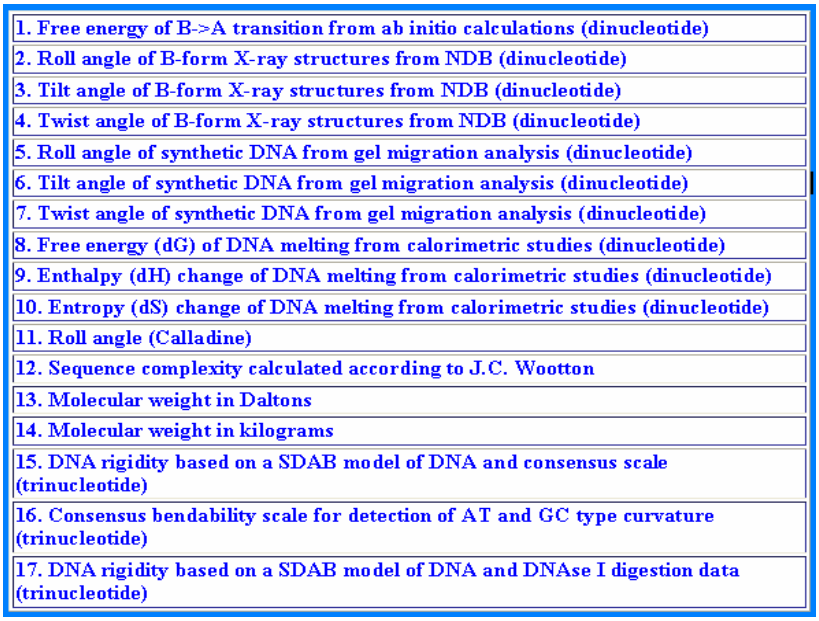

Fig: List of few methods for DNA taken from plot.it® Server

We can easily construct 2D and 3D plots from these data – using Excel or other standard software.

Symbolic structures: helical wheels, helical nets,

In many cases we can learn a lot by putting our sequence data on a symbolic 3D structure, such as an alpha-helix. Amphiphilic helices like to lie on the surface of a membrane or on the outside of a protein, so that one side faces the hydrophobic interior, the other one the hydrophilic solvent. These can be easily recognized by coloring the amino acids by hidrophobicity…

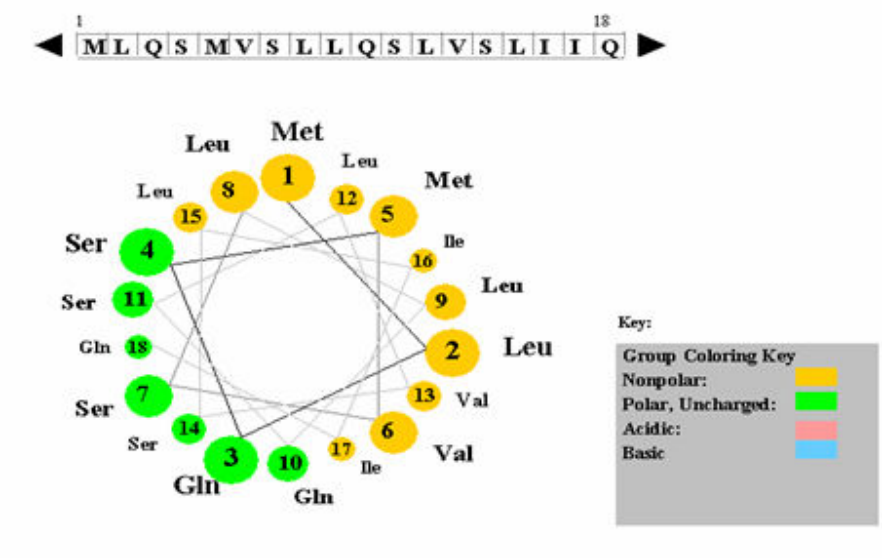

Fig: An example of an amino acid sequence plotted on a helical wheel. This figure is a snapshot of a Java Applet written by Edward K. O'Neil and Charles M. Grisham (University of Virginia in Charlottesville, Virginia). It can be also plotted with the helical wheel program from the EMBOSS suite known as PEPWHEEL.

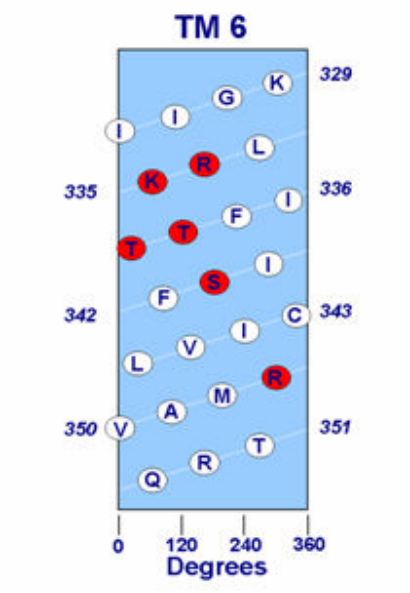

Fig: An example of amino acid sequence of TM6 of CFTR plotted as a helical net. The sequence is displayed as if the cylinder of the helix has been sliced open along its long axis, which allows visualization of the displacement of adjacent positions by 100 degrees. Helical nets can be plotted with program from the EMBOSS suite known as PEPNET.

Visualization of 3D structure

This is not in the scope of our course, even for a simple field like proteins there are various conventions, rules etc. There is a good compendium of these rules at:

Chemical representation
http://www.cscs.ch/~mvalle/ChemViz/representations/index.html

Graphic representation of Molecular Structure
http://cmm.cit.nih.gov/modeling/guide_documents/graphics_representations.html

World Index of Bio Molecular Visualization Resources
http://molvis.sdsc.edu/visres/molvisfw/titles.jsp

Links to Free Viewer and Plug-in Softwares
http://www.indiana.edu/~cheminfo/ca_mvts.html#plug

Most of the common visualization tricks are in standard programs such as SwissPdbviewer…. These programs usually use the PDB files for producing pictures. However if you want to make an individual representation and are willing to do some simple programming, you can modify the PDB files.