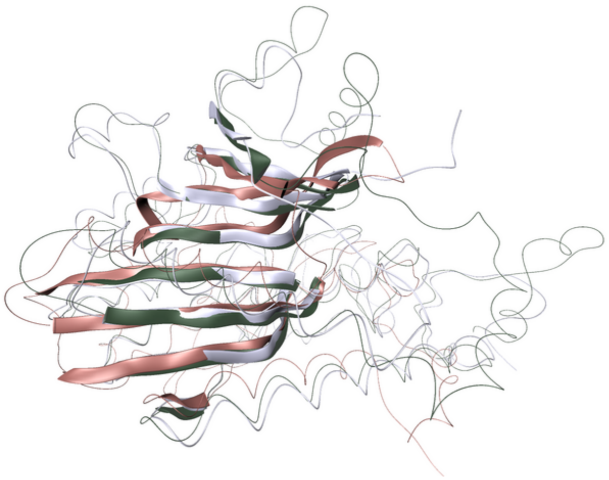
University of Illinois at Urbana-Champaign
Luthey-Schulten Group
Beckman Institute for Advanced Science and Technology
Theoretical and Computational Biophysics Group
San Francisco Workshop

# Bioinformatics and Sequence Alignment



Felix Autenrieth

Barry Isralewitz

Zaida Luthey-Schulten

Anurag Sethi

Taras Pogorelov

June 2005

# Contents

# Introduction

The sequencing of the genomes from several organisms, and high-throughput X-ray structure analysis, have brought to the scientific community a large amount of data about the sequences and structures of several thousand proteins. This information can effectively be used for medical and biological research only if one can extract functional insight from it. Two main computational techniques exist to help us reach this goal: a *bioinformatics* approach, and full atom *molecular dynamics* simulations. Bioinformatics uses the statistical analysis of protein sequences and structures to help annotate the genome, to understand their function, and to predict structures when only sequence information is available. Molecular modeling and molecular dynamics simulations use the principles from physics and physical chemistry to study the function and folding of proteins. Bioinformatics methods are among the most powerful technologies available in life sciences today. They are used in fundamental research on theories of evolution and in more practical considerations of protein design. Algorithms and approaches used in these studies range from sequence and structure alignments, secondary structure prediction, functional classification of proteins, threading and modeling of distantly-related homologous proteins to modeling the progress of protein expression through a cell's life cycle.

In this tutorial you will begin with classical pairwise sequence alignment methods using the Needleman-Wunsch algorithm, and end with the multiple sequence alignment available through CLUSTAL W. You will start out only with sequence and biological information of class II aminoacyl-tRNA synthetases, key players in the translational mechanism of cell. Then you will classify protein domains and align the catalytic domains. If structural alignments are considered to be the true alignments, you will see that simple pair sequence alignment of two proteins with low sequence identity has serious limitations. Finally you will determine the phylogenetic relationship of class II tRNA synthetases with a dendrogram algorithm. You will carry out the exercises with the program MATLAB and the Needleman-Wunsch alignment program provided by A. Sethi. Many of the tools of the field can be freely accessed by any person with a web browser; a listing of our favorite bioinformatics tools and resources is provided in Section 8.

# Getting Started

*The entire tutorial takes about 2 hours to complete. Aside from this document, there are several necessary programs and files to run this tutorial.*

Files required for this tutorial are available for download at:
http://www.scs.uiuc.edu/∼schulten/bioinformatics.html

There are certain programs required to complete this tutorial successfully:
MATLAB
CLUSTAL W (Download: ftp://ftp-igbmc.u-strasbg.fr/pub/ClustalW/)
Phylip (Download: http://evolution.genetics.washington.edu/phylip/getme.html)
Pair (Provided in the tutorial file package)
Multiple (Provided in the tutorial file package)
SeqQR (Provided in the tutorial file package, but installation is required. Look in Bioinformatics/PheRS/seqqr for instructions)

You also have the option of running CLUSTAL W and Phylip on the web:
CLUSTAL W: http://bioweb.pasteur.fr/seqanal/interfaces/clustalw.html
Phylip: http://bioweb.pasteur.fr/seqanal/phylogeny/phylip-fr.html

# 1 Biology of class II aminoacyl-tRNA Synthetases

The aminoacyl-tRNA synthetases (AARSs) are key proteins involved in the translation machinery in living organisms; it is not surprising, therefore, that these enzymes are found in all three domains of life. There are twenty specific tRNA synthetases, one for each amino acid, although not all organisms contain the full set. Studying the function, structure, and evolution of these proteins remains an area of intense interest as, in addition to being a major constituent of the translation process, these proteins are also believed to contain vital information spanning the evolution of life from the ancient "RNA world" to the modern form of life.

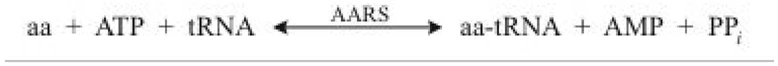


Figure 1: The overall reaction catalyzed by the aminoacyl tRNA synthetases.

The AARSs are responsible for loading the twenty different amino acids (aa) onto their cognate tRNA during protein synthesis (see Figure 1 and Table 1). Each AARS is a multidomain protein consisting of (at least) a catalytic domain and an anticodon binding domain. In all known cases, the synthetases divide into class I or class II types; class I AARSs have a basic Rossmann fold, while class II AARSs have a fold that is unique to them and the biotin synthetase holoenzyme. Additionally, some of the AARSs, for example aspartyl-tRNA synthetase, have an "insert domain" within their catalytic domain (see Figure 2). Recognition of the tRNA molecule is performed both on the anticodon domain and the catalytic domain, which interacts with the acceptor arm and the so-called discriminator base. These molecular machines operate with remarkable precision, making only one mistake in every 10,000 translations. The intricate architecture of specific tRNA synthetases helps to discriminate against miscoding.

| Amino Acid | Single Letter | Three Letter |
|---|---|---|
| Alanine | A | Ala |
| Arginine | R | Arg |
| Asparagine | N | Asn |
| Aspartic acid | D | Asp |
| Asparagine or aspartic acid | B | Asx |
| Cysteine | C | Cys |
| Glutamine | Q | Gln |
| Glutamic acid | E | Glu |
| Glutamine or glutamic acid | Z | Glx |
| Glycine | G | Gly |
| Histidine | H | His |
| Isoleucine | I | Ile |
| Leucine | L | Leu |
| Lysine | K | Lys |
| Methionine | M | Met |
| Phenylalanine | F | Phe |
| Proline | P | Pro |
| Serine | S | Ser |
| Threonine | T | Thr |
| Tryptophan | W | Trp |
| Tyrosine | Y | Tyr |
| Valine | V | Val |

Table 1: Amino acids names and letter codes

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

Table 2: The genetic code. Some species have slightly different codes. $^*$This codon also specifies the initiator tRNA$^{\mathrm{fMet}}$.

## 2 AARSs in *Archaeoglobus fulgidus*

*In this section, you will find all the tRNA synthetases in an organism. The* A. fulgidus *genome, with NCBI accession number* `NC_000917`*, has been completely sequenced, so we can perform comprehensive searches through all of its genes.*

- Open a web browser.
- Access the NCBI database (`http://www.ncbi.nlm.nih.gov`).
- Type the *A.fulgidus* accession number `NC_000917` into the Search box, with Genome selected as the search type, click on the single result, and you will reach the site with the complete annotated sequence of *A.fulgidus*.
- The best way to find Class II tRNA synthetases in this genome is via the listing of proteins organized by COG (Cluster of Orthologous Groups) functional categories. Click on the COG functional categories link located about halfway down the page as part of the text "Gene Classifaction based on **COG functional categories**" (note: this is *not* the COGs link, located near the top of the page, just beneath "BLAST protein homologs"). You will reach a catalog of *A.fulgidus* proteins organized by functional annotation.
- Choose the link which is most likely to provide you with information about class II tRNA synthetases. (Hint: tRNA synthetase function is part of *translation*.) Click on this link and scroll to the search fields at the bottom of the resulting page.
- Type the string `tRNA synthetase` in the text search box at the bottom of the page and you will receive a summary of all tRNA synthetases in *A.fulgidus*. Consult the genetic code in Table 2 for help in answering the following questions.

**Questions.** How many tRNA synthetases are in *A. fulgidus*? Among these, are there any that bind the same amino acid? Why might this be the case? How many tRNA synthetases are minimally required to synthesize all proteins in one organism? What is "codon usage" and how can information about codon usage be applied to distinguish genomes from two organisms?

# 3 Domain structure of class II tRNA synthetases

*In this section you will study the similarity between functional domains in tRNA synthetases from two domains of life.*

In the following steps, we will perform a search for the tRNA synthetases from *E. coli* and *M. jannaschii* in a database that can perform searches based on similar domain structure, the NCBI Entrez Structure/MMDB/3D Domain tools. Sample output is shown in Figure 2.

- At the home page of the NCBI database (`http://www.ncbi.nlm.nih.gov`), set the search-type pop-up menu to Structure and search for `tRNA synthetase`.

- In the search result lists, click on the links for the individual tRNA synthetases (e.g. 1NJ5), to access information about the domain structure of individual class II tRNA synthetases. Click on individual domains in the 3D Domain to see a display of structural neighbors: domains with similar structure to your target domain are shown aligned to your target domain. (Warning: if you receive the error "Vast neighbor data for this domain are not yet available.", just use the Back button to return to the search results and try again with another structure.)

- The online web version of this tutorial includes a color figure of the domains of tRNA synthetase, color-coded by domain as Figure 2. The NCBI domain results are colored by the same code as in this figure: magenta= Catalytic, blue= Insert I, orange= Insert II, green = Anticodon.

- Look among the entires in your search results to find tRNA synthetases from two or three different species. Open separate browser windows to show the domain organization of each and compare their 1-D organization.

- Look among the entires in your search results to find two or three tRNA synthetases from the same species. Open separate browser windows to show the domain organization of each and compare their 1-D organization.

**Question.** What is the biological function of protein domains in one of your chosen tRNA synthetases? Which is more similar: the domain structures of different tRNA synthetases within one organism (paralogs), or the domain structures of tRNA synthetases from different organisms (orthologs)?

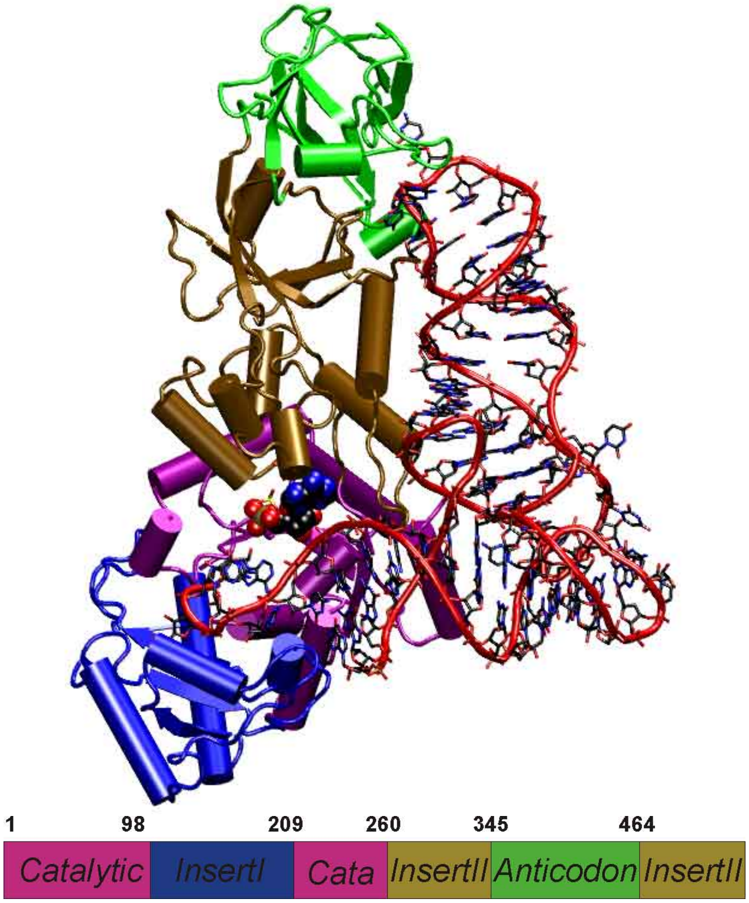


Figure 2: Domains of Glutamine aminoacyl-tRNA synthetase. The bound tRNA is shown with red backbone on the right side of the figure.

# 4 SCOP fold classification

As with all multi-domain proteins, the properties of Class II tRNA synthetases depend on their component domains. The schematic domain structure of aspartyl class II tRNA synthetase from Eubacteria is depicted in Figure 2.

Only one domain is common to all class II tRNA synthetases studied so far: the catalytic domain, which carries out the amino acid loading reaction. In later sections of this tutorial, you will align the catalytic domains of class II tRNA synthetases using Needleman-Wunsch sequence alignment, and with structure-based alignment methods. The fourth column of Table 6 provide you with the ASTRAL database accession codes [2]. The ASTRAL database ( tt http://astral.stanford.edu) is a compendium of protein domain structures derived from the PDB database [3].

In the following exercise you will classify one of your chosen catalytic domains in folds, superfamilies and families with help of the database SCOP (Structural Classification Of Proteins) [4].

- Point your web browser to the SCOP server (`http://scop.berkeley.edu/index.html`).
- You are going to browse a hierarchy of protein structural classifications. Go to the top of the hierarchy: Click on the link top of the hierarchy. This places you at the root of all the protein classes deposited in SCOP.
- Click on one of the Classes links (e.g. Alpha and Beta proteins); you should reach the next hierarchy level of folds. Note how lineage records your path through the hierarchy.
- Some of the 3D structure renderings of an example of the protein class entries have been prepared in advance. Click on the purple and white buttons to the right of a few entries to see an example view of the chosen protein class.
- We will now search for entries relevant to a particular domain of interest. Now enter one of the ASTRAL database catalytic domain codes from Table 6 (e.g. `d1asza2`) in the Search field you find at the bottom of the page.
- You should reach a window showing the SCOP lineage and a summary of all relevant PDB entry domains. For subsequent exercises we have provided you with catalytic domain coordinates of Class II tRNA synthetases obtained from the ASTRAL database. The ASTRAL coordinate files are produced by extracting domains from PDB coordinate files.

**Questions.** What is the lineage of your chosen tRNA synthetase domain? What are some other members in the SCOP family and superfamily of the catalytic domain of Class II tRNA synthetase? What is the most abundant fold in SCOP's "Alpha and Beta proteins" fold class ?

# 5 Sequence Alignment Algorithms

*In this section you will optimally align two short protein sequences using pen and paper, then search for homologous proteins by using a computer program to align several, much longer, sequences.*

Dynamic programming algorithms are recursive algorithms modified to store intermediate results, which improves efficiency for certain problems. The Smith-Waterman (Needleman-Wunsch) algorithm uses a dynamic programming algorithm to find the optimal local (global) alignment of two sequences — $a$ and $b$. The alignment algorithm is based on finding the elements of a matrix $H$ where the element $H_{i,j}$ is the optimal score for aligning the sequence ($a_1$,$a_2$,...,$a_i$) with ($b_1$,$b_2$,.....,$b_j$). Two similar amino acids (e.g. arginine and lysine) receive a high score, two dissimilar amino acids (e.g. arginine and glycine) receive a low score. The higher the score of a path through the matrix, the better the alignment. The matrix $H$ is found by progressively finding the matrix elements, starting at $H_{1,1}$ and proceeding in the directions of increasing $i$ and $j$. Each element is set according to:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + S_{i,j} \\ H_{i-1,j} - d \\ H_{i,j-1} - d \end{cases}$$

Where $S_{i,j}$ is the similarity score of comparing amino acid $a_i$ to amino acid $b_j$ (obtained here from the BLOSUM40 similarity table) and $d$ is the penalty for a single gap. The matrix is initialized with $H_{0,0} = 0$. When obtaining the local Smith-Waterman alignment, $H_{i,j}$ is modified:

$$H_{i,j} = \max \begin{cases} 0 \\ H_{i-1,j-1} + S_{i,j} \\ H_{i-1,j} - d \\ H_{i,j-1} - d \end{cases}$$

The gap penalty can be modified, for instance, $d$ can be replaced by $(d \times k)$, where $d$ is the penalty for a single gap and $k$ is the number of consecutive gaps.

Once the optimal alignment score is found, the "traceback" through $H$ along the optimal path is found, which corresponds to the the optimal sequence alignment for the score. In the next set of exercises you will manually implement the Needleman-Wunsch alignment for a pair of short sequences, then perform global sequence alignments with a computer program developed by Anurag Sethi, which is based on the Needleman-Wunsch algorithm with an affine gap penalty, $d + e(k-1)$, where $e$ is the extension gap penalty. The output file will be in the GCG format, one of the two standard formats in bioinformatics for storing sequence information (the other standard format is FASTA).

## 5.1 Manually perform a Needleman-Wunsch alignment

In the first exercise you will test the Smith-Waterman algorithm on a short sequence parts of hemoglobin (PDB code `1AOW`) and myoglobin 1 (PDB code `1AZI`).

- Here you will align the sequence `HGSAQVKGHG` to the sequence `KTEAEMKASEDLKKHGT`.
- The two sequences are arranged in a matrix in Table 3. The sequences start at the upper right corner, the initial gap penalties are listed at each offset starting position. With each move from the start position, the initial penalty increase by our single gap penalty of 8.

| | | H | G | S | A | Q | V | K | G | H | G |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | -8 | -16 | -24 | -32 | -40 | -48 | -56 | -64 | -72 | -80 |
| K | -8 | | | | | | | | | | |
| T | -16 | | | | | | | | | | |
| E | -24 | | | | | | | | | | |
| A | -32 | | | | | | | | | | |
| E | -40 | | | | | | | | | | |
| M | -48 | | | | | | | | | | |
| K | -56 | | | | | | | | | | |
| A | -64 | | | | | | | | | | |
| S | -72 | | | | | | | | | | |
| E | -80 | | | | | | | | | | |
| D | -88 | | | | | | | | | | |
| L | -96 | | | | | | | | | | |
| K | -104 | | | | | | | | | | |
| K | -112 | | | | | | | | | | |
| H | -120 | | | | | | | | | | |
| G | -128 | | | | | | | | | | |
| T | -136 | | | | | | | | | | |

Table 3: The empty matrix with initial gap penalties.

- The first step is to fill in the similarity scores $S_{i,j}$ from looking up the matches in the BLOSUM40 table, shown here labeled with 1-letter amino acid codes:

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 5 | | | | | | | | | | | | | | | | | | | | | | |
| R | -2 | 9 | | | | | | | | | | | | | | | | | | | | | |
| N | -1 | 0 | 8 | | | | | | | | | | | | | | | | | | | | |
| D | -1 | -1 | 2 | 9 | | | | | | | | | | | | | | | | | | | |
| C | -2 | -3 | -2 | -2 | 16 | | | | | | | | | | | | | | | | | | |
| Q | 0 | 2 | 1 | -1 | -4 | 8 | | | | | | | | | | | | | | | | | |
| E | -1 | -1 | -1 | 2 | -2 | 2 | 7 | | | | | | | | | | | | | | | | |
| G | 1 | -3 | 0 | -2 | -3 | -2 | -3 | 8 | | | | | | | | | | | | | | | |
| H | -2 | 0 | 1 | 0 | -4 | 0 | 0 | -2 | 13 | | | | | | | | | | | | | | |
| I | -1 | -3 | -2 | -4 | -4 | -3 | -4 | -4 | -3 | 6 | | | | | | | | | | | | | |
| L | -2 | -2 | -3 | -3 | -2 | -2 | -2 | -4 | -2 | 2 | 6 | | | | | | | | | | | | |
| K | -1 | 3 | 0 | 0 | -3 | 1 | 1 | -2 | -1 | -3 | -2 | 6 | | | | | | | | | | | |
| M | -1 | -1 | -2 | -3 | -3 | -1 | -2 | -2 | 1 | 1 | 3 | -1 | 7 | | | | | | | | | | |
| F | -3 | -2 | -3 | -4 | -2 | -4 | -3 | -3 | -2 | 1 | 2 | -3 | 0 | 9 | | | | | | | | | |
| P | -2 | -3 | -2 | -2 | -5 | -2 | 0 | -1 | -2 | -2 | -4 | -1 | -2 | -4 | 11 | | | | | | | | |
| S | 1 | -1 | 1 | 0 | -1 | 1 | 0 | 0 | -1 | -2 | -3 | 0 | -2 | -2 | -1 | 5 | | | | | | | |
| T | 0 | -2 | 0 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | 0 | -1 | -1 | 0 | 2 | 6 | | | | | | |
| W | -3 | -2 | -4 | -5 | -6 | -1 | -2 | -2 | -5 | -3 | -1 | -2 | -2 | 1 | -4 | -5 | -4 | 19 | | | | | |
| Y | -2 | -1 | -2 | -3 | -4 | -1 | -2 | -3 | 2 | 0 | 0 | -1 | 1 | 4 | -3 | -2 | -1 | 3 | 9 | | | | |
| V | 0 | -2 | -3 | -3 | -2 | -3 | -3 | -4 | -4 | 4 | 2 | -2 | 1 | 0 | -3 | -1 | 1 | -3 | -1 | 5 | | | |
| B | -1 | -1 | 4 | 6 | -2 | 0 | 1 | -1 | 0 | -3 | -3 | 0 | -3 | -3 | -2 | 0 | 0 | -4 | -3 | -3 | 5 | | |
| Z | -1 | 0 | 0 | 1 | -3 | 4 | 5 | -2 | 0 | -4 | -2 | 1 | -2 | -4 | -1 | 0 | -1 | -2 | -2 | -3 | 2 | 5 | |
| X | 0 | -1 | -1 | -1 | -2 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 0 | -1 | -2 | 0 | 0 | -2 | -1 | -1 | -1 | -1 | -1 |
| | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V | B | Z | X |

- We fill in the BLOSUM40 similarity scores for you in Table 4.
- To turn this $S$ matrix intro the dynamic programming $H$ matrix requires calculation of the contents of all 170 boxes. We've calculated the first 4 here, and encourage you to calculate the contents of at least 4 more. The practice will come in handy in the next steps. As described above, a matrix square cannot be filled with its dynamic programming value until the squares above, to the left, and to the above-left diagonal are computed. The value of a square is

  $$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + S_{i,j} \\ H_{i-1,j} - d \\ H_{i,j-1} - d \end{cases}$$

  using the convention that $H$ values appear in the top part of a square in large print, and $S$ values appear in the bottom part of a square in small print. Our gap penalty $d$ is 8.
- Example: In the upper left square in Table 4, square (1,1), the similarity score $S_{1,1}$ is -1, the number in small type at the bottom of the box. The value to assign as $H_{1,1}$ will be the greatest ("max") of these three values: $(H_{0,0}+S_{1,1}), (H_{0,1}-d), (H_{1,0}-d))$. That is, the greatest of: $(0+-1), (-8-8), (-8-8)$ which just means the greatest of: -1, -16, and -16. This is -1, so we write -1 as the value of $H_{1,1}$ (the larger number in the top part of the box). The same reasoning in square (2,1) leads us to set $H_{2,1}$ as -9, and so on.

  *Note: we consider $H_{0,0}$ to be the "predecessor" of $H_{1,1}$, since it helped decided $H_{1,1}$'s value. Later, predecessors will qualify to be on the traceback path.*
- Again, just fill in 4 or 5 boxes in Table 4 until you get a feel for gap penalties and similarity scores S vs. alignment scores H. In the next step, we provide the matrix with all values filled in as Table 5. Check that your 4 or 5 calculations match.
- Now we move to Table 5, with all 170 $H_{i,j}$ values are shown, to do the "alignment traceback". To find the alignment requires one to trace the path through from the end of the sequence (the lower right box) to the start of the sequence (the upper left box). This job looks complicated, but should only take about 5 –7 minutes.
- We are tracing a path in Table 5, from the lower right box to the upper left box. You can only move to a square if it could have been a "predecessor" of your current square – that is, when the matrix was being filled with $H_{i,j}$ values, the move from the predecessor square to your current square would have followed the mathematical rules we used to find H$_{i,j}$ above. Circle each square you move to along your path.

- Example: we start at the lower right square (10,17), where $H_{10,17}$ is -21 and $S_{10,17}$ is -2. We need to test for 3 possible directions of movement: diagonal (up + left), up, and left. The condition for diagonal movement given above is: $H_{i,j} = H_{i-1,j-1} + S_{i,j}$, so for the diagonal box (9,16) to have contributed to (10,17), $H_{9,16} + S_{10,17}$ would have to equal the H value of our box, -21. Since (-29 + -2) does not equal -21, the diagonal box is not a "predecessor", so we can't move in that direction. We try the rule for the box to the left: $H_{i,j} = H_{i-1,j} - d$ Since -37 - 8 does not equal -21, we also can't move left. Our last chance is moving up. We test $H_{i,j} = H_{i,j-1} - d$. Since -21 = (-13 - 8) we can move up! Draw an arrow from the lower right box, $(H_{10,17} = -21, S_{10,17} = -2)$ to the box just above it, $(H_{10,16} = -13, S_{10,16} = 8)$ .

- Continue moving squares, drawing arrows, and circling each new square you land on, until you have reached the upper right corner of the matrix If the path branches, follow both branches.

- Write down the alignment(s) that corresponds to your path(s) by writing the the letter codes on the margins of each position along your circled path. Aligned pairs are at the boxes at which the path exits via the upper-left corner. When there are horizontal or vertical movements movements along your path, there will be a gap (write as a dash, "`-`") in your sequence.

- Now to check your results against a computer program. We have prepared a pairwise Needleman-Wunsch alignment program, `pair`, which you will apply to the same sequences which you have just manually aligned.

- Go to /Workshop/bioinformatics-tutorial/bioinformatics, or download `Bioinformatics.tar` from the following website:

  http://www.scs.uiuc.edu/∼schulten/bioinformatics.html

- Unzip it using the following command at a UNIX prompt:
  ```
  >tar zxvf Bioinformatics.tar
  ```

- Change your directory by typing at the Unix prompt:
  ```
  cd ~/Bioinformatics/pairData
  ```
  Use the `ls` command to view the contents of your directory.
  Notice that there are three different `pair` files, distinguished by operating system (i.e. pair-linux, pair-macosx, and pair-sgi).
  Then start the pair alignment appropriate for your operating system by typing one of the following commands:
  ```
  pair-linux targlist
  pair-macosx targlist
  pair-sgi targlist
  ```

All alignments will be carried out using the BLOSUM40 matrix, with a gap penalty of 8. The paths to the input files and the BLOSUM40 matrix used are defined in the file `targlist`; the BLOSUM40 matrix is the first 25 lines of the file `blosum40`. (Other substitution matrices can be found at the NCBI/Blast website.)

*Note: In some installations, the* `pair` *executable is in* `~/Bioinformatics/pairData` *and here you must type either* `./pair-linux targlist`*,* `./pair-macosx targlist`*, or* `./pair-sgi targlist` *to run it.*
*If you cannot access the* `pair` *executable at all, you can see the output from this step in* `~/Bioinformatics/pairData/example_output/`

- After executing the program you will generate three output files namely `align`, `scorematrix` and `stats`. View the alignment in GCG format by typing `less align`. The file `scorematrix` is the 17x10 $H$ matrix. If there are multiple paths along the traceback matrix, the program `pair` will choose only one path, by following this precedence rule for existing potential traceback directions, listed in decreasing precedence: diagonal (left and up), up, left. In the file `stats` you will find the optimal alignment score and the percent identity of the alignment.

**Questions.** Compare your manual alignment to the the output of the pair program. Do the alignments match?

## 5.2 Finding homologous pairs of ClassII tRNA synthetases

Homologous proteins are proteins derived from a common ancestral gene. In this exercise with the Needleman-Wunsch algorithm you will study the sequence identity of several class II tRNA synthetases, which are either from Eucarya, Eubacteria or Archaea, or differ in the kind of aminoacylation reaction which they catalyze. Table 6 summarizes the reaction type, the organism and the PDB accession code and chain name of the employed Class II tRNA synthetase domains.

- We have have prepared a computer program `multiple` which will align multiple pairs of proteins.

- Change your directory by typing at the Unix prompt:
  `cd ~/Bioinformatics/multipleData`
  then start the alignment executable for your operating system by typing one of the following:
  `multiple-linux targlist`
  `multiple-macosx targlist`

```
multiple-sgi targlist
```

*Note: In some installations, the* `multiple` *executable is in* `~/Bioinformatics/multipleData` *and here you must type* `./multiple-linux targlist`*,* `./multiple-macosx targlist`, or `./multiple-sgi targlist` to run it.
If you cannot access the `multiple` executable at all, you can see the output from this step in `~/Bioinformatics/multipleData/example_output/`

- In the `align` and `stats` files you will find all combinatorial possible pairs of the provided sequences. On a piece of paper, write the names of the the proteins, grouped by ther domain of life, as listed in Table 6. Compare sequence identities of aligned proteins from the same domain of a life, and of aligned proteins from different domains of life, to help answer the questions below.

- To study the evolution, look at multiple alignments of sequence and structure. These alignments are more accurate than pairwise alignments. Use CLUSTALW for closely related proteins (30–40% identity); STAMP should be used for more distantly related proteins. Structure and sequence give the same tree for certain regions, creating congruence of structure and sequence-based phylogenies. We will learn sequence-based phylogenetic mehods for $\alpha$-chain PheRS later, and structure-based for AARS in the *Evolution of Protein Structure* tutorial.

**Questions.** What criteria do you use in order to determine if two proteins are homologous? Can you find a pattern when you evaluate percent identities between the pairs of class II tRNA synthetases? Which is the most evolutionarily related pair, and which is the most evolutionarily divergent pair according to the sequence identity?

| | | **H** | **G** | **S** | **A** | **Q** | **V** | **K** | **G** | **H** | **G** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | -8 | -16 | -24 | -32 | -40 | -48 | -56 | -64 | -72 | -80 |
| **K** | -8 | **−1**<br>−1 | **−9**<br>−2 | 0 | −1 | 1 | −2 | 6 | −2 | −1 | −2 |
| **T** | -16 | **−9**<br>−2 | **−3**<br>−2 | 2 | 0 | −1 | 1 | 0 | −2 | −2 | −2 |
| **E** | -24 | 0 | −3 | 0 | −1 | 2 | −3 | 1 | −3 | 0 | −3 |
| **A** | -32 | −2 | 1 | 1 | 5 | 0 | 0 | −1 | 1 | −2 | 1 |
| **E** | -40 | 0 | −3 | 0 | −1 | 2 | −3 | 1 | −3 | 0 | −3 |
| **M** | -48 | 1 | −2 | −2 | −1 | −1 | 1 | −1 | −2 | 1 | −2 |
| **K** | -56 | −1 | −2 | 0 | −1 | 1 | −2 | 6 | −2 | −1 | −2 |
| **A** | -64 | −2 | 1 | 1 | 5 | 0 | 0 | −1 | 1 | −2 | 1 |
| **S** | -72 | −1 | 0 | 5 | 1 | 1 | −1 | 0 | 0 | −1 | 0 |
| **E** | -80 | 0 | −3 | 0 | −1 | 2 | −3 | 1 | −3 | 0 | −3 |
| **D** | -88 | 0 | −2 | 0 | −1 | −1 | −3 | 0 | −2 | 0 | −2 |
| **L** | -96 | −2 | −4 | −3 | −2 | −2 | 2 | −2 | −4 | −2 | −4 |
| **K** | -104 | −1 | −2 | 0 | −1 | 1 | −2 | 6 | −2 | −1 | −2 |
| **K** | -112 | −1 | −2 | 0 | −1 | 1 | −2 | 6 | −2 | −1 | −2 |
| **H** | -120 | 13 | −2 | −1 | −2 | 0 | −4 | −1 | −2 | 13 | −2 |
| **G** | -128 | −2 | 8 | 0 | 1 | −2 | −4 | −2 | 8 | −2 | 8 |
| **T** | -136 | −2 | −2 | 2 | 0 | −1 | 1 | 0 | −2 | −2 | −2 |

Table 4: Alignment score worksheet. In all alignment boxes, the similarity score $S_{i,j}$ from the BLOSUM40 matrix lookup is supplied (small text, bottom of square). Four alignment scores are provided as examples (large text, top of square), try and calculate at least four more, following the direction provided in the text for calculating $H_{i,j}$.

| | | **H** | **G** | **S** | **A** | **Q** | **V** | **K** | **G** | **H** | **G** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | -8 | -16 | -24 | -32 | -40 | -48 | -56 | -64 | -72 | -80 |
| **K** | -8 | **−1**<br>−1 | **−9**<br>−2 | **−16**<br>0 | **−24**<br>−1 | **−31**<br>1 | **−39**<br>−2 | **−42**<br>6 | **−50**<br>−2 | **−58**<br>−1 | **−66**<br>−2 |
| **T** | -16 | **−9**<br>−2 | **−3**<br>−2 | **−7**<br>2 | **−15**<br>0 | **−23**<br>−1 | **−30**<br>1 | **−38**<br>0 | **−44**<br>−2 | **−52**<br>−2 | **−60**<br>−2 |
| **E** | -24 | **−16**<br>0 | **−11**<br>−3 | **−3**<br>0 | **−8**<br>−1 | **−13**<br>2 | **−21**<br>−3 | **−29**<br>1 | **−37**<br>−3 | **−44**<br>0 | **−52**<br>−3 |
| **A** | -32 | **−24**<br>−2 | **−15**<br>1 | **−10**<br>1 | **2**<br>5 | **−6**<br>0 | **−13**<br>0 | **−21**<br>−1 | **−28**<br>1 | **−36**<br>−2 | **−43**<br>1 |
| **E** | -40 | **−32**<br>0 | **−23**<br>−3 | **−15**<br>0 | **−6**<br>−1 | **4**<br>2 | **−4**<br>−3 | **−12**<br>1 | **−20**<br>−3 | **−28**<br>0 | **−36**<br>−3 |
| **M** | -48 | **−39**<br>1 | **−31**<br>−2 | **−23**<br>−2 | **−14**<br>−1 | **−4**<br>−1 | **5**<br>1 | **−3**<br>−1 | **−11**<br>−2 | **−19**<br>1 | **−27**<br>−2 |
| **K** | -56 | **−47**<br>−1 | **−39**<br>−2 | **−31**<br>0 | **−22**<br>−1 | **−12**<br>1 | **−3**<br>−2 | **11**<br>6 | **3**<br>−2 | **−5**<br>−1 | **−13**<br>−2 |
| **A** | -64 | **−55**<br>−2 | **−46**<br>1 | **−38**<br>1 | **−26**<br>5 | **−20**<br>0 | **−11**<br>0 | **3**<br>−1 | **12**<br>1 | **4**<br>−2 | **−4**<br>1 |
| **S** | -72 | **−63**<br>−1 | **−54**<br>0 | **−41**<br>5 | **−34**<br>1 | **−25**<br>1 | **−19**<br>−1 | **−5**<br>0 | **4**<br>0 | **11**<br>−1 | **4**<br>0 |
| **E** | -80 | **−71**<br>0 | **−62**<br>−3 | **−49**<br>0 | **−42**<br>−1 | **−32**<br>2 | **−27**<br>−3 | **−13**<br>1 | **−4**<br>−3 | **4**<br>0 | **8**<br>−3 |
| **D** | -88 | **−79**<br>0 | **−70**<br>−2 | **−57**<br>0 | **−50**<br>−1 | **−40**<br>−1 | **−35**<br>−3 | **−21**<br>0 | **−12**<br>−2 | **−4**<br>0 | **2**<br>−2 |
| **L** | -96 | **−87**<br>−2 | **−78**<br>−4 | **−65**<br>−3 | **−58**<br>−2 | **−48**<br>−2 | **−38**<br>2 | **−29**<br>−2 | **−20**<br>−4 | **−12**<br>−2 | **−6**<br>−4 |
| **K** | -104 | **−95**<br>−1 | **−86**<br>−2 | **−73**<br>0 | **−66**<br>−1 | **−56**<br>1 | **−46**<br>−2 | **−32**<br>6 | **−28**<br>−2 | **−20**<br>−1 | **−14**<br>−2 |
| **K** | -112 | **−103**<br>−1 | **−94**<br>−2 | **−81**<br>0 | **−74**<br>−1 | **−64**<br>1 | **−54**<br>−2 | **−40**<br>6 | **−34**<br>−2 | **−28**<br>−1 | **−22**<br>−2 |
| **H** | -120 | **−99**<br>13 | **−102**<br>−2 | **−89**<br>−1 | **−82**<br>−2 | **−72**<br>0 | **−62**<br>−4 | **−48**<br>−1 | **−42**<br>−2 | **−21**<br>13 | **−29**<br>−2 |
| **G** | -128 | **−107**<br>−2 | **−91**<br>8 | **−97**<br>0 | **−88**<br>1 | **−80**<br>−2 | **−70**<br>−4 | **−56**<br>−2 | **−40**<br>8 | **−29**<br>−2 | **−13**<br>8 |
| **T** | -136 | **−115**<br>−2 | **−99**<br>−2 | **−89**<br>2 | **−96**<br>0 | **−88**<br>−1 | **−78**<br>1 | **−64**<br>0 | **−48**<br>−2 | **−37**<br>−2 | **−21**<br>−2 |

Table 5: Traceback worksheet. The completed alignment score matrix $H$ (large text, top of each square) with the BLOSUM40 lookup scores $\mathrm{S}_{i,j}$ (small text, bottom of each square). To find the alignment, trace back starting from the lower right (T vs G, score -21) and proceed diagonally (to the left and up), left, or up. Only proceed, however, if the square in that direction could have been a predecessor, according to the conditions described in the text.

| Specificity | Organism | PDB code:chain | ASTRAL catalytic domain |
|---|---|---|---|
| Aspartyl | Eubacteria | 1EQR:B | d1eqrb3 |
| Aspartyl | Archaea | 1B8A:A | d1b8aa2 |
| Aspartyl | Eukarya | 1ASZ:A | d1asza2 |
| Glycl | Archaea | 1ATI:A | d1atia2 |
| Histidyl | Eubacteria | 1ADJ:C | d1adjc2 |
| Lysl | Eubacteria | 1BBW:A | d1bbwa2 |
| Aspartyl | Eubacteria | 1EFW:A | d1efwa3 |

Table 6: Domain types, origins, and accession codes

# 6 Molecular phylogenetic tree.

*In this section you will plot a dendrogram displaying the measured similarities between the seven proteins which you pairwise aligned in Section 5. You will compare there relative position in the dendrogram to their relative position in the phylogenetic tree of life.*

Below is the pairwise alignment scores from the 21 pairs aligned in section 5. The information in ~/Bioinformatics/multipleData/stats is assembled into a symmetric matrix:

| | 1EQR | 1ATI | 1ADJ | 1EFW | 1ASZ | 1B8A | 1BBW |
|---|---|---|---|---|---|---|---|
| 1EQR | 0.0 | 0.21 | 0.23 | 0.55 | 0.28 | 0.31 | 0.31 |
| 1ATI | 0.21 | 0.0 | 0.25 | 0.24 | 0.24 | 0.18 | 0.21 |
| 1ADJ | 0.23 | 0.25 | 0.0 | 0.24 | 0.24 | 0.21 | 0.23 |
| 1EFW | 0.55 | 0.24 | 0.24 | 0.0 | 0.34 | 0.36 | 0.30 |
| 1ASZ | 0.28 | 0.24 | 0.24 | 0.34 | 0.0 | 0.41 | 0.27 |
| 1B8A | 0.31 | 0.18 | 0.21 | 0.36 | 0.41 | 0.0 | 0.28 |
| 1BBW | 0.31 | 0.21 | 0.23 | 0.30 | 0.27 | 0.28 | 0.0 |

The commands in the following Matlab session are all in the Matlab script `Dendro.m`. The commands can be run all at once simply by typing `Dendro` at the Matlab command line when Matlab's current directory contains `Dendro.m`.

Now we will use the clustering algorithms in the Statistics toolbox of Matlab to draw a dendrogram of the relatedness of the domains. Here we use the above scores derived from sequence alignment, but structure alignment scores could be used as well[5].

- Move to the directory for this exercise with `cd ~/Bioinformatics/matlabData`
- Start Matlab by typing at the UNIX console: `matlab`.
- The commands in the following Matlab session are all in the Matlab script `Dendro.m`. The commands can be run all at once simply by typing `Dendro` at the Matlab command line, as long as Matlab's current directory contains `Dendro.m and distM.dat`. If you like, type in the below, or paste lines into the Matlab command line from `Dendro.m` or the web-based version of this tutorial. (To see the numerical result of a calculation, leave of the semicolon from the end of the line. To see the value of a variable, enter its name alone on the Matlab command line.)
- First, we read in the above distance matrix of sequence similarity for 7 proteins.

  ```
  load distM.dat;
  ```

- We make a new matrix by subtracting the sequence similarity values from 1, so that longer distances in our dendrogram will correspond to greater evolutionary distance.

```
dM=1-distM;
```

- Its important to keep track of names of the proteins...

```
l={’1eqr’,’1ati’,’1adj’,’1efw’,’1asz’,’1b8a’,’1bbw’};
```

- To use the ‘linkage’ command of Matlab, one needs to form a column vector of the $((n)(n-1)/2)$ non-redundant elements above the main diagonal of the $n \times n$ distance matrix; our $7 \times 7$ matrix produces a 21-element vector:

```
d=[dM(2:7,1);dM(3:7,2);dM(4:7,3);dM(5:7,4);dM(6:7,5);dM(7:7,6)];
```

- Use the `linkage` command to make a hierarchical cluster tree using average distance between cluster elements:

```
z1=linkage(d’,’average’);
```

- For more options in constructing the cluster tree, type `help linkage` at the Matlab command line, also see a modeling text such as Leach [6].
- We display the dendrogram of the clusters in `z1`:

```
h101=figure(101);
dendrogram(z1);
```

- And, finally, paste in some magic to place the labels correctly:

```
hx=get(get(h101,’CurrentAxes’),’XTickLabel’);
for i=1:size(hx,1)
  hx(i)=str2double(hx(i));
end
set(get(h101,’CurrentAxes’),’XTickLabel’,[l(hx(1)), ...
      l(hx(2)),l(hx(3)),l(hx(4)),l(hx(5)),l(hx(6)),l(hx(7))])
figure(h101);
title(’Molecular Phylogenetic Tree’);
xlabel(’Protein (pdb code)’)
ylabel(’1-Similarity (%)’)
```

- Print out the dendrogram, or copy it down on paper, along with the names of the proteins. Refer to Table 6 to write, under each name, the domain of life each protein originates from.

**Questions.** What is the pair with the closest evolutionary relation? What is the pair with the most distant relation? Is the arrangement of the proteins in the dendrogram consistent with what we know about the evolution of the three domains of life?

# 7 Phylogenetic tree of $\alpha$-chain PheRS

In this section, you will study the evolutionary history of the $\alpha$-chain PheRS proteins. After this, you will choose a non-rendundant set of proteins that represents the phylogenetic structure of the $\alpha$-chain PheRS.

In the first step, we will find all the $\alpha$-chain PheRS in the Swiss-Prot database [7]. The Swiss-Prot database is a highly annotated database of protein sequences.

1. Use a web browser to go to the Swiss-Prot website which is located at the url http://us.expasy.org/sprot/.
2. In the text-box, search for the protein SYFA_ECOLI. This is the $\alpha$-chain PheRS from *E. Coli*.
3. Examine the information given about the protein and click on the FASTA format link at the bottom of the page. At this stage, we have the sequence for the $\alpha$-chain PheRS from *E. Coli*.

In order to study the evolutionary history of the $\alpha$-chain PheRS, we will have to compare the sequences of $\alpha$-chain PheRS from a number of organisms. To find the $\alpha$-chain PheRS from other organisms, we will use BLAST.

1. Copy the sequence of the protein and perform a BLAST search of the protein in NCBI at the url http://www.ncbi.nlm.nih.gov/BLAST/Blast.cgi? over the Swiss-Prot database.
2. Paste the sequence information that you copied in Swiss-Prot onto the text box in BLAST.
3. Choose the Swiss-Prot database. Change expectation value to 1E-05 and perform the search.
4. All the sequences found in this search can be downloaded from NCBI. Use the file Phe.fasta with abbreviated organism names we have provided along with the tutorial files. Move to the directory with the file using the command `cd ~/Bioinformatics/PheRS`.

Note that the sequences contain the whole protein and not just the catalytic domain. An AARS protein is a multi-domain protein with a catalytic domain and a anticodon binding domain. The last sequence in this search "YG60_METJA" was shown to be a putative class II CysRS and not a PheRS recently [8]. Hence, in the phylogenetic tree, we will use this sequence as an outgroup.

Before, you can compare the sequences, you will have to make a multiple alignment of the sequences. This can be done using CLUSTAL W.

1. To perform a multiple alignment, open the CLUSTAL W program. In UNIX, type the following commands at the promp within the appropriate directory:

   ```
   >clustalw
   ```

2. The CLUSTAL W menu comes up. Select option 1 and enter the file name of the downloaded sequences.
3. You will be back at the main menu. Now, align the sequences by choosing option 2 and then choosing option 1 inside the multiple alignment menu.

These multiple sequence alignments are only approximate and to perform an accurate phylogenetic tree, one would have to improve the alignment manually. However, for this excercise, we will use the alignment from CLUSTAL W to get the phylogenetic tree. After entering the output file names, you will be ready to create a phylogenetic tree from the multiple sequence alignment.

We will use Phylip to get the phylogenetic tree of the $\alpha$-chain PheRS proteins. We can use CLUSTAL W to create a phylip format file.

1. Go to main menu of CLUSTAL W and choose option 4 for Phylogenetic trees. Choose option 4 inside the phylogenetic tree menu.
2. After choosing output file name, execute the draw tree command.
3. Now, exit from CLUSTAL W by going to main menu and choosing the option X.
4. To view the phylogenetic tree, we will need to choose the retree program from the Phylip package. On a unix prompt, type the command

   ```
   >retree
   ```

5. Inside retree, type Y and give the tree file from CLUSTAL W as input. Use ? to find all the options.
6. Use page up and page down to find the node number of the sequence YG60_METJA and choose this node as the outgroup.
7. Exit retree after writing the tree with the new root. Now, you are ready to draw the tree using the dragram commandon the UNIX prompt.

   ```
   >drawgram
   ```

8. The program will ask you for the input file name. Give the filename outtree. Enter the filename of the tree file from CLUSTAL W.
9. It will then ask you for a font file name. You should have the font file font1 among the tutorial files.
10. Give the path and file name of font1 and Phylip will take you to a series of drawgram menus. Choose L so that you can view the postscript of the phylogenetic tree. Choose N so that the tree is not previewed.

11. In the main drawgram menu, choose 1, choose phylogram, and then give option P and choose 4 and then give an angle of 90 degrees to get the standard format for phylogenetic trees.

12. The output of the phylogenetic tree is given in plotfile in ps format. You can view the phylogenetic tree using the command ghostview. Take a print out of the phylogenetic tree.

**Question:.** What kind of phylogenetic pattern is shown by the $\alpha$-chain PheRS? Is there any horizontal gene transfer? Are there more proteins from some domains of life than from others? Which domains of life have more proteins than others? To view the organism information for each protein, you can use Swiss-Prot.

As you can see, the databases have bias towards certain domains of life. To get a nonredundant set of sequences that provide the phylogenetic structure of the PheRSs, we will use the newly developed Sequence QR algorithm. The theory for the multidimensional QR algorithm and it's applications to sequences and structures is provided in [9, 10, 8]. To run Sequence QR, we will have to give the following command on the UNIX prompt:

```
>clustalw -infile=<alignment_file.aln> -convert
-output=fasta -outfile=<alignmentfile.fasta>
>seqqr -p 0.28 <alignmentfile.fasta> <QRalignment.fasta>
```

The first command converts the alignment file of the proteins you found in BLAST into FASTA format. After executing the first command, make sure to remove the YG60_METJA protein from the sequence alignment as this protein is not a PheRS. The second command finds the QR representative set for the proteins. The -p option specifies the percentage identity threshold for the nonredundant set. In this case, 28% gives a protein from the Eucarya, Bacteria and Archaea domains of life and one from the mitochrondria of a eucarya. As you increase the the percentage threshold, the redundancy in the QR set increase.

**Question:.** Draw the phylogenetic tree of the QR non-redundant set. Does the non-redundant set represent the major evolutionary changes in the $\alpha$-chain PheRS equally? Does the non-redundant set reproduce the structure of the phylogenetic tree? What happens as you increase the percentage identity threshold for the QR algorithm?

**Question:.** Look at the alignment of all the PheRS and the non-redundant set of PheRS. Are there any conserved positions in the alignment? In [9], it was shown that both classes of AARSs have some conserved elements. Is PheRS a class I or class II AARS? Mark the conserved regions in the non-redundant set of PheRS and submit the alignment along with the homework.

# 8 Other bioinformatics tools

So far in this tutorial, you have made use of only a small selection of bioinformatics techniques and tools. In the last exercise we invite you to explore additional tools and resources by yourself. Results of aligning sequences can be improved by systematically building up profiles from multiple sequences.

- Try using the multisequence alignment servers such as ClustalW or servers employing Hidden Markov methods to build a profile from the four aspartyl AARSs sequences.
- Align the histidyl AARS to the profile.
- Check if you can obtain an alignment closer to the structure-based alignment you will see in VMD next week.

*Tools, resources, and link collections:*

**ClustalW** (`http://bioweb.pasteur.fr/seqanal/interfaces/clustalw.html`) Perform a multi-sequence or profile-profile alignment with the program ClustalW. Just access the website directly and paste in all or a selection of your Class II tRNA synthetases in order to execute the program. ClustalW is the most widely used tool in bioinformatics for carrying out multi-sequence alignments.

**Psipred** (`http://bioinf.cs.ucl.ac.uk/psipred/`) Predict the secondary structure of one of your Class II tRNA synthetases with the Psipred Protein Structure Prediction Server. Paste your sequence in the input sequence window, provide your email address and you will receive after a few minutes a secondary structure prediction of your chosen tRNA synthetase. Sequence and structural alignments as well as secondary predictions form the framework for a successful modeling project.

**3D PSSM** (`http://www.sbg.bio.ic.ac.uk/~3dpssm/`) A web-based method method for protein fold recognition using sequence profiles coupled with secondary structure.

**TMpred** (`http://www.ch.embnet.org/software/TMPRED_form.html`) A database scoring-based method to predict the transmembrane portions of membrane proteins.

**TMHMM** (`http://www.cbs.dtu.dk/services/TMHMM-2.0/`) A hidden Markov method to predict the transmembrane portions of membrane proteins.

**European Bioinformatics Inst.** (`http://www.ebi.ac.uk/services/index.html`) An up-to-date and well-organized collection of links to bioinformatics tools, databases, and resources. The site provides advice as to the best or most popular tools in a category, and provides short descriptions of all entries.

**ExPASy Molecular Biology Server** (`http://ca.expasy.org/`) Another well-organized directory of online analysis tools, databases, and other resources, with a greater focus on proteins. “The ExPASy (Expert Protein Analysis System) proteomics server of the Swiss Institute of Bioinformatics (SIB) is dedicated to the analysis of protein sequences and structures...” With this server you can start your own homology modeling project of an unknown class II tRNA synthetase, namely Alanyl-tRNA synthetase. You can obtain the sequence in FASTA format from the SwissProt database which can be accessed directly from the ExPASy server with the accession number SYA_ECOLI. As structural template choose one of the provided catalytic domain structures of class II tRNA synthetases. You can also model the other domains for which you need to find an appropriate template from the provided PDB structures.

**SwissModel** (`http://swissmodel.expasy.org/`)
For model generation use SwissModel, where you can thread your sequence upon one or several of your chosen templates. SwissModel provides you with an online tutorial and will perform refinements on initial models you submit to its server.

**Dynamic Programming in Java**
(`http://www.dkfz-heidelberg.de/tbi/bioinfo/PracticalSection/AliApplet/index.html`)
This is an alternative Smith-Waterman tutorial which will provide you with a web-based interface for dynamic programming, an animated version of the paper-and-pencil exercise in section 5.

**Biology WorkBench** (`http://workbench.sdsc.edu`)
This website allows you to search popular protein and nucleic acid sequence databases. Sequence retrieval is integrated with access to a variety of analysis tools as for example the multi-sequence alignment program ClustalW. The advantage of the Biology Workbench is that all analysis tools are interconnected with each other eliminating the tedious file conversion process, which often needs to be done when accessing tools from distinct locations.

**CASP5** (`http://predictioncenter.llnl.gov/casp5/Casp5.html`)
Every two years a community-wide protein structure prediction contest takes place,where groups complete for prediction of unpublished protein structures. One can check out how well has our Resource done in the last year contest. Just search for Zan Schulten Group results on this site.

# References

[1] Michael S. Waterman. *Introduction to Computational Biology: Maps, Sequences, and Genomes.* CRC Press, 1995.

[2] M. Leavitt S.E. Benner, P. Koehl. The astral compendium for sequence and structure analysis. *Nucleic Acid Research*, pages 254–256, 2000.

[3] Z. Feg G. Gilliland T.N. Bhat H. Weissig I.N. Shindyalov P.E. Bourne H.M. Berman, J. Westbrook. The protein data bank. *Nucleic Acid Research*, (28).

[4] T. Hubbard C. Chothia A.G. Murzin, S.E. Brenner. Scop: a structural classification of protein databases for the investigation of sequences and structures.

[5] P. O'Donoghue and Z. Luthey-Schulten. On the evolution of structure in aminoacyl-trna synthetases. *Microbiol Mol Biol Rev.*, 67(4), 2003.

[6] A.R. Leach. *Molecular Modelling: Principles and Applications (2nd edition).* Prentice Hall, Upper Saddle River, New Jersey, 2001.

[7] B Boeckmann, A Bairoch, R Apweiler, M-C Blatter, A Estreicher, E Gasteiger, M J Martin, K Michoud, C O'Donovan, I Phan, and *et al.* The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. 31:365–370, 2003.

[8] A Sethi, P O'Donoghue, and Z Luthey-Schulten. Evolutionary profiles from the qr factorization of multiple sequence alignments. *PNAS*, 2005. In Press.

[9] Patrick O'Donoghue and Zaida Luthey-Schulten. Evolution of Structure in Aminoacyl-tRNA Synthetases. *Microbiol Mol Biol Rev*, 67:550–573, 2003.

[10] P O'Donoghue and Z Luthey-Schulten. Evolutionary profiles derived from the QR factorization of multiple structural alignments gives an economy of information. 2004. In press.