Advanced multiple sequence alignment

Week Seven, Tuesday, October 7, 2003

Author and Instructor: Steven M. Thompson

Expectation maximization, alignment refinement, profile analysis, and Markov models

Lab covers: 1) using PROSITE and MEME to discover known and hidden motifs; 2) running the progressive, pairwise alignment program PileUp with the SeqLab Editor to develop and refine a multiple sequence alignment; 3) creating traditional Gribskov and statistically based HMM profiles for remote similarity searching and further alignment; 4) exploring visualization and annotation techniques for multiple sequence alignments; 5) dealing with amino acid and DNA sequences concurrently.
GCG® is the Genetics Computer Group, part of Accelrys Inc., a subsidiary of Pharmacopeia Inc., producer of the Wisconsin Package® for sequence analysis.

© 2003 BioInfo 4U
Beyond just aligning sequences — How good can you make it and so what?

Standard disclaimer

I write these tutorials from a ‘lowest-common-denominator’ biologist’s perspective. That is, I only assume that you have fundamental molecular biology knowledge, but are relatively inexperienced regarding computers. As a consequence of this they are written quite explicitly. Therefore, if you do exactly what is written, it will work. However, this requires two things: 1) you must read very carefully and not skim over vital steps, and 2) you mustn’t take offense if you already know what I’m discussing. I’m not insulting your intelligence. This also makes the tutorials longer than otherwise necessary. Sorry.

I use three writing conventions in the tutorials, besides my casual style. I use bold type for those commands and keystrokes that you are to type in at your keyboard or for buttons or menus that you are to click in a GUI. I also use bold type for section headings. Screen traces are shown in a ‘typewriter’ style Courier font and “/ / / / / / / / /” indicates abridged data. The arrow symbol (>) indicates the system prompt and should not be typed as a part of commands. Really important statements may be underlined.

As you’ve learned, specialized X-server graphics communications software is required to use GCG’s SeqLab. I’ll remind you of a few user hints while using X: X Windows are only active when the mouse cursor is in that window, and always close X Windows when you are through with them to conserve system memory. Furthermore, to activate X items, just <click> on them, rather than holding your mouse button down. Also, X buttons are turned on when they are pushed in and shaded. Finally, don’t close X Windows with the X-server software’s close icon in the upper right- or left-hand window corner, rather, always, if available, use the window’s own “File” menu “Exit” choice, or “Close,” or “Cancel,” or “OK” button.

Introduction

The power and sensitivity of sequence based computational methods dramatically increases with the addition of more data. More data yields stronger analyses — if done carefully! Otherwise, it can confound the issue. The patterns of conservation become clearer by comparing the conserved portions of sequences amongst a larger and larger dataset. Those areas most resistant to change are functionally the most important to the molecule. The basic assumption is that those portions of sequence of crucial functional value are most constrained against evolutionary change. They will not tolerate many mutations. Not that mutations do not occur in these portions, just that most mutations in the region are lethal so we never see them. Other areas of sequence are able to drift more readily being less subject to evolutionary pressure. Therefore, sequences end up a mosaic of quickly and slowly changing regions over evolutionary time. However, in order to learn anything by comparing sequences, we need to know how to compare them. We can use those constrained portions as ‘anchors’ to create a sequence alignment allowing comparison, but this brings up the alignment problem and ‘similarity.’ It is easy to see that sequences are aligned when they have identical symbols at identical positions, but what happens when symbols are not identical or the sequences are not the same
length. How can we know that the most similar portions of our sequences are aligned, when is an alignment optimal, and does optimal mean biologically correct? How can anybody figure any of this out?

**Multiple sequence dynamic programming**

As seen in pairwise dynamic programming, looking at every possible position by sliding one sequence along every other sequence, just will not work for alignment. Therefore, dynamic programming reduces the problem back down to \( N^2 \). But how do you work with more than just two sequences at a time? It becomes a much harder problem. You could painstakingly manually align all your sequences using some type of editor, and many people do just that, but some type of an automated solution is desirable, at least as a starting point to manual alignment. However, solving the dynamic programming algorithm for more than just two sequences rapidly becomes intractable. Dynamic programming’s complexity, and hence its computational requirements, increases exponentially with the number of sequences in the dataset being compared (complexity = \([\text{sequence length}]^\text{number of sequences}\)). Mathematically this is an \( N \)-dimensional matrix, quite complex indeed. As we’ve seen earlier, pairwise dynamic programming solves a two-dimensional matrix, and the complexity of the solution is equal to the length of the longest sequence squared. Well, a three member standard dynamic programming sequence comparison would be a matrix with three axes, the length of the longest sequence cubed, and so forth. You can at least draw a three-dimensional matrix, but more than that becomes impossible to even visualize. It quickly boggles the mind!

Several different heuristics have been employed over the years to simplify the complexity of the problem. One program, MSA (Gupta et al. [version 2.0, 1995] and version 2.1), does attempt to globally solve the \( N \)-dimensional matrix equation using a bounding box trick. However, the algorithm’s complexity precludes its use in most situations, except with very small datasets. One way to still globally solve the algorithm and yet reduce its complexity is to restrict the search space to only the most conserved ‘local’ portions of all the sequences involved. This approach is used by the program PIMA (Smith and Smith, version 1.4, 1995). MSA and PIMA are both available through the Internet at several bioinformatics servers (in particular the Baylor College of Medicine’s Search Launcher at [http://searchlauncher.bcm.tmc.edu/](http://searchlauncher.bcm.tmc.edu/)) where you can investigate these resources on your own time.

**How the algorithm works**

The most common implementations of automated multiple alignment modify dynamic programming by establishing a pairwise order in which to build the alignment. This modification is known as pairwise, progressive dynamic programming. Originally attributed to Feng and Doolittle (1987), this variation of the dynamic programming algorithm generates a global alignment, but restricts its search space at any one time to a local neighborhood of the full length of only two sequences. Consider a group of sequences. First all are compared to each other, pairwise, using normal dynamic programming. This establishes an order for the set, most to least similar. Subgroups are clustered together similarly. Then take the top two, most similar sequences, and align them using normal dynamic programming. Now create a consensus of the two and
align that consensus to the third sequence using standard dynamic programming. Now create a consensus of the first three sequences and align that to the forth most similar. This process continues until it has worked its way through all sequences and/or sets of clusters. The pairwise, progressive solution is implemented in several programs. Perhaps the most popular is Higgins’ and Thompson’s ClustalW (1994) and its multi-platform, graphical user interface ClustalX (Thompson, et al., 1997). These can be found at biocomputing sites around the globe and installed on your own machine or run through the World Wide Web (WWW). ClustalX has versions available for most windowing computing Operating Systems — most UNIX flavors, Microsoft Windows, and Macintosh. The ClustalX homsite guarantees the latest version: ftp://ftpigbmc.u-strasbg.fr/pub/ClustalX/. Complete documentation comes with the program and is accessed through a “Help” menu. The GCG program PileUp implements a very similar method.

As seen with pairwise alignments and sequence database similarity searching, all of this is much easier with protein sequences versus nucleotide sequences. Twenty symbols are just much easier to align then only four; the signal to noise ratio is again so much better. And, as in database searching, the concept of similarity applies to amino acids but generally not to nucleotides. Therefore, just like in database searching, multiple sequence alignment should always be done on a protein level if at all possible, unless the DNA sequences are so similar as to not cause any problem. Therefore, translate nucleotide sequences to their protein counterparts if you are dealing with coding sequences before performing multiple sequence alignment. The process is much more difficult if you are forced to align nucleotides because the region does not code for a protein. Automated methods may be able to help as a starting point, but they are certainly not guaranteed to come up with a biologically correct alignment. The resulting alignment will probably have to be extensively edited, if it works at all. Success will largely depend on the similarity of the nucleotide dataset.

One liability of global progressive, pairwise methods is they are entirely dependent on the order in which the sequences are aligned. Fortunately ordering them from most similar to least similar usually makes biological sense and works very well. However, the techniques are very sensitive to the substitution matrix and gap penalties specified. Programs such as ClustalW and PileUp that allow ‘fine-tuning’ areas of an alignment by re-alignment with different scoring matrices and/or gap penalties can be extremely helpful because of this. However, any automated multiple sequence alignment program should be thought of as only a tool to offer a starting alignment that can be improved upon, not the ‘end-all-to-meet-all’ solution, guaranteed to provide the ‘one-true’ answer.

Reliability?

To help assure the reliability of sequence alignments always use comparative approaches. A multiple sequence alignment is a hypothesis of evolutionary history. Insure that you have prepared a good one — be sure that it makes sense. Think about it — a sequence alignment is a statement of positional homology. It establishes the explicit homologous correspondence of each individual sequence position, each column in the alignment. Therefore, devote considerable time and energy toward developing the most satisfying multiple sequence alignment possible. Editing alignments is allowed and to be encouraged. Specialized sequence
editing software such as GCG’s SeqLab Editor help achieve this but any editor will do as long as the sequences end up properly formatted afterwards. After some automated solution has offered its best guess, go into the alignment and use your own brain to improve it. Use all available information and understanding to insure that all columns are truly homologous. Look for conserved functional sites to help guide your judgement. Assure that known enzymatic, regulatory, and structural elements all align, for the results of subsequent analyses are absolutely dependent upon the alignment.

Researchers have successfully used the conservation of co-varying sites in ribosomal and other structural RNA alignments to assist in alignment refinement. That is, as one base in a stem structure changes the corresponding Watson-Crick paired base will change in a corresponding manner. This process has been used extensively by the Ribosomal Database Project at the Center for Microbial Ecology at Michigan State University to help guide the construction of their rRNA alignments and structures. The WWW Uniform Resource Locator (URL) is http://rdp.cme.msu.edu/html/.

Be sure an alignment makes biological sense — align things that make sense to align! Beware of comparing ‘apples and oranges.’ If creating alignments for phylogenetic inference, either make paralogous comparisons (i.e. evolution via gene duplication) to ascertain gene phylogenies within one organism, or orthologous (within one ancestral loci) comparisons to ascertain gene phylogenies between organisms which should imply organismal phylogenies. Try not to mix them up without complete data representation. Lots of confusion can arise, especially if you do not have all the data and/or if the nomenclature is contradictory; extremely misleading interpretations can result. Be wary of trying to align genomic sequences with cDNA when working with DNA; the introns will cause all sorts of headaches. Similarly, do not align mature and precursor proteins from the same organism and loci. It does not make evolutionary sense, as one is not evolved from the other, rather one is the other. These are all easy mistakes to make; try your best to avoid them.

Remember the old adage “garbage in — garbage out!” Some general guidelines to remember include the following:

- If the homology of a region is in doubt, then throw it out.
- Avoid the most diverged parts of molecules; they are the greatest source of systematic error.
- Do not include sequences that are more diverged than necessary for the analysis at hand.

**Applicability?**

So what’s so great about multiple sequence alignments; why would anyone want to bother? They are:

- very useful in the development of PCR primers and hybridization probes;
- great for producing annotated, publication quality, graphics and illustrations;
- invaluable in structure/function studies through homology inference;
- essential for building “Profiles” for remote homology similarity searching; and
required for molecular evolutionary phylogenetic inference programs such as those from PAUP* (Phylogenetic Analysis Using Parsimony [and other methods]) and PHYLIP (PHYLogeny Inference Package).

A multiple sequence alignment is useful for probe and primer design by allowing you to visualize the most conserved regions of an alignment. This technique is invaluable for designing phylogenetic specific probes as it clearly localizes areas of high conservation and high variability in an alignment. Depending on the dataset that you analyze, any level of phylogenetic specificity can be achieved. Pick areas of high variability in the overall dataset that correspond to areas of high conversation in phylogenetic category subset datasets to differentiate between universal and specific potential probe sequences. After localizing general target areas on the sequence, you can then use any of a number of primer discovery programs to find the best primers within those regions and to test those potential probes for common PCR conditions and problems.

Graphics prepared from multiple sequence alignments can dramatically illustrate functional and structural conservation. These can take many forms of all or portions of an alignment — shaded or colored boxes or letters for each residue, cartoon representations of features, running line graphs of overall similarity, overlays of attributes, various consensus representations, etc. — all can be printed with high-resolution equipment, usually in color or gray tones. These can make a big difference in a poster or manuscript presentation.

Conserved regions of an alignment are functionally important. In addition to the conservation of primary sequence and function, structure is also conserved in these crucial regions. In fact, recognizable structural conservation between true homologues extends way beyond statistically significant sequence similarity. An oft-cited example is in the serine protease superfamily. *S. griseus* protease A demonstrates remarkably little similarity when compared to the rest of the superfamily (Expectation values \(E() \gg 10^{-18}\) in a typical search) yet its three-dimensional structure clearly shows its allegiance to the serine proteases (Pearson, W.R., personal communication). These principles are the premise of ‘homology modeling’ and it works remarkably well.

Profiles are a position specific weight matrix description of an alignment or a portion of an alignment. Gap insertion is penalized more heavily in conserved areas of the alignment than it is in variable regions, and the more highly conserved a residue is, the more important it becomes. Originally described by Gribskov (1987), later refinements have added more statistical rigor (see e.g. Eddy’s Hidden Markov Model Profiles [1996 and 1998]). Several profile style programs will be described in detail later in the tutorial. Generally, a profile is created from an alignment of related sequences and then used to search databases for remote sequence similarities. Profile searching is tremendously powerful and can provide the most sensitive, albeit extremely computationally intensive, database similarity searches possible.

Finally, we can use multiple sequence alignments to infer phylogeny. Based on the assertion of homologous positions in an alignment, several algorithms can estimate the most reasonable evolutionary tree for that alignment. This is a huge, complicated, and highly contentious field, hopefully to be delved into later in your lifelong learning experience. (See the Woods Hole Marine Biological Laboratory’s excellent summer course,
the Workshop on Molecular Evolution, at http://workshop.molecularevolution.org/.) However, always remember that regardless of algorithm used, parsimony, any distance method, maximum likelihood, or even Bayesian Inference, all molecular sequence phylogenetic inference programs make the absolute validity of your input alignment their first and most critical assumption.

I reiterate, the most important factor in inferring reliable phylogenies is the accuracy of the multiple sequence alignment. The interpretation of your results is utterly dependent on the quality of your input. In fact, many experts advice against using any parts of the sequence data that are at all questionable. Only analyze those portions that assuredly do align. If any portions of the alignment are in doubt, throw them out. This usually means trimming down or masking out the alignment’s terminal ends and may require internal trimming or masking as well. Biocomputing is always a delicate balance — signal against noise — and sometimes it can be quite the balancing act!

**Complications**

One of the biggest problems in computational biology is that of molecular sequence data format. Each suite of programs to come along seems to require its own different sequence format. The major databases all have their own; Clustal has its own; even the database similarity searching program FastA has a sequence format associated with it. GCG Wisconsin Package sequence format exists both as single and Multiple Sequence Format (MSF) and GCG's SeqLab has its own format called Rich Sequence Format (RSF) that contains both sequence data and reference and feature annotation. PAUP* has a required format called the NEXUS file and PHYLIP has its own unique input data format requirements. The PAUP* interfaces in the GCG Wisconsin Package, PAUPSearch and PAUPDisplay, automatically generate their required NEXUS format directly from the GCG formatted files. Most systems are not nearly so helpful. Several different programs are available to convert formats back and forth between the required standards, but it all can get quite confusing. One program available, ReadSeq by Don Gilbert at Indiana University (1990), allows for the back and forth conversion between several different formats. I would heartily recommend installing it on all of your computers. It comes as an old ‘tried-and-trued’ C version or a new JAVA version with a graphical interface. I don’t have much experience with the JAVA version but have relied on the C version for many years. Alignment gaps are another problem. Different program suites may use different symbols to represent them. Most programs use hyphens, “-“, the Wisconsin Package uses periods, “.”. Furthermore, not all gaps in sequences should be interpreted as deletions. Interior gaps are probably okay to represent this way, as regardless of whether a deletion, insertion or a duplication event created the gap, logically they will be treated the same by the algorithms. These are indels. However, end gaps should not be represented as indels because a lack of information beyond the length of a given sequence may not be due to a deletion or insertion event. It may have nothing to do with the particular stretch being analyzed at all. It may just not have been sequenced! These gaps are just place holders for the sequence. Therefore, it is safest to manually edit an alignment to change leading and trailing gap symbols to “x”s which mean “unknown amino acid,” or “n”s
which mean “unknown base,” or “?”’s which is supported by many programs, but not all, and means “unknown residue or indel.” This will assure that the programs don’t make incorrect assumptions about your sequences.

Searching PROSITE

Before aligning a bunch of sequences, it’s a good idea to scan those sequences for features that will help you recognize a good alignment, and the programs that do this type of scan work best before any gaps have been introduced by the alignment process. Many, many features have been described and catalogued in biological sequences over the years. Most of these have recognizable consensus patterns that allow you to screen an unknown sequence for their occurrence. Check out the following example to understand how these patterns are developed. A very simplistic approach is to look at an alignment, see that certain regions are conserved, and create a consensus of that region. A multiple sequence alignment of Elongation Factor Tu/1 from many different organisms illustrates the conservation of the first of several GTP-binding domains in these proteins, that area around position twenty in the alignment below:

Based on experimental evidence, we know that the indicated region bounded by the Glycine and Serine above is essential. So we merely count up the various residues in those locations and assign the most common one to the consensus. Simple. But what about the fact that the middle Histidine isn’t always a Histidine; in this data set, just as often it’s a Serine and sometimes it’s an Alanine. Other positions are also seen not be invariant. And there’s lots of other members of this gene family not being represented here at all. A consensus isn’t necessarily the biologically “correct” combination. How do we include this other information? A simple consensus throws much of it away. Therefore, we need to adopt some sort of standardized ambiguity notation. The trick is to define a motif such that it minimizes false positives and maximizes true positives; i.e. it needs to be just discriminatory enough. The development of the exact motif is largely empirical; a pattern is made, tested against the database, then refined, over and over, although when
experimental evidence is available, it is always incorporated. This approach is known as motif definition and fortunately a scientist in Switzerland, Amos Bairoch, has done it for tons of sequences!

His database of catalogued structural, regulatory, and enzymatic consensus patterns or ‘signatures’ originally named the PROSITE Dictionary of Protein Sites and Patterns (1992) and now called the PROSITE Database of protein families and domains, contains 1170 documentation entries that describe 1604 different patterns, rules, and profiles/matrices (Release 17.37, February 13, 2003). Pattern descriptions for these characteristic local sequence areas are variously and confusingly known as motifs, templates, signatures, patterns, and even fingerprints; don’t let the terminology bewilder you. Those that GCG’s Motifs program can access are one-dimensional, ‘regular-expression’ descriptions, that encode ambiguity, of some sort of functional or otherwise constrained consensus region of a sequence alignment (e.g. glycosylation and phosphorylation sites, SH3-binding sites, nuclear localization sequence, and enzymatic active sites). Common motifs may or may not represent sequence homology and may or may not encompass an entire structural domain — they do not all signify known function or common origin. Regardless, PROSITE is one of the quickest and easiest databases to search with a peptide sequence. The GCG program Motifs performs this search. The program can tolerate mismatches with a -MisMatch option and it displays an abstract with selected references for each motif signature found. In many cases this can be a tremendous aid in ascertaining possible function of an unknown peptide sequence. It can often lead to immediate answers and routes of investigation. It should always be utilized — it’s just too fast and simple to ignore.

Extensive abstract and reference lists follow the identified sequence locations for each site. This information can save anybody a tremendous amount of work! The sites themselves are shown with their sequence locations below each consensus pattern. The consensus pattern described above and characteristic of most nucleophosphate binding proteins is called the P-Loop and is defined as (A,G)x4GK(S,T), i.e. either an Alanine or a Glycine, followed by four of anything, followed by an invariant Glycine-Lysine pair, followed by either a Serine or a Threonine. Exceptions are noted in the documentation. This particular site has been very well researched and many three-dimensional structures are available for it. It always has a beta/alpha/beta secondary structure conformation and is sometimes known as the “Rossman Fold.” The site is shown below in the Guanine Nucleotide-Binding Protein G(I), Alpha-1 Subunit (Adenylate Cyclase-Inhibiting) from Rattus norvegicus (the common rat), Swiss-Prot GBI1_Rat, courtesy ExPASy’s Swiss-3DImage collection (ftp://ca.expasy.org/databases/swiss-3dimage/IMAGES/JPEG/S3D00521.jpg):
Post-translational modification sites commonly found in many proteins, such as glycosylation, phosphorylation, amidation, and myristylation, will only be listed if you specify the -Frequent option. However, realize that sites may be false positives, especially if you use the -Frequent option. This is always a danger with simple consensus style searches. The GCG programs ProfileScan and HmmerPfam use a much more sensitive profile matrix approach to search your sequence with profiles including most of PROSITE and will be discussed further later on. Notice in the example above that Motifs discovered the truly positive GTP-binding elongation factor signature and the ATP/GTP-binding P-loop site, yet it also found two probable false positives, the Prokaryotic membrane lipoprotein lipid attachment site and the FGGY family of carbohydrate kinases signature.

**Expectation maximization**

Another powerful motif discovery algorithm can be run before actually performing multiple sequence alignment on a dataset. This algorithm is called Expectation Maximization; it uses Bayesian probabilities and unsupervised learning to find, *de novo*, unknown conserved motifs among a group of unaligned, un-gapped sequences (Bailey and Elkan, 1994). The motifs do not have to be in congruent order among the different sequences; i.e. it has the power to discover ‘unalignable’ motifs between sequences. This characteristic differentiates MEME from the other profile building techniques described below. It is implemented in the Wisconsin Package as the MEME program and it produces output containing a multiple profile file as well as a readable report file. Its profile output serves as input to MotifSearch (Bailey and Gribskov, 1998). I would strongly suggest reading the MEME and MotifSearch chapters in the GCG Program Manual (genmanual at
the command line or the Help buttons in the program in SeqLab) — they explain the details of the algorithms way better than I can.

Profile analysis — Weighted Position Specific Site Matrices (PSSM) of multiple sequence alignments.

OK, so one-dimensional motifs are one way to ‘capture’ the information of an important portion of an alignment. However, motifs can’t convey any degree of residue ‘importance.’ For instance, in the GTP-binding P-Loop described above, is it better to have an Alanine or a Glycine in that first position or doesn’t it matter? This lack of sense of importance causes a loss of sensitivity. More ‘robust’ methods can convey the importance of each residue in the region.

Given a multiple sequence alignment, how can we use the extra information contained in it to find ever more remotely similar sequences? How do we search and explore into and past Russell Doolittle’s “Twilight Zone,” i.e. those similarities below ~25% identity, those Z scores below ~4, those BLAST/Fast E values above ~10^{-3} or so? Just because a similarity score between two sequences is quite poor, we do not automatically know that the two structures do not fold in a similar manner or perform a similar function, we have no idea of homology at all!

Obviously much of the information in a multiple sequence alignment is ‘noise’ at this similarity level. We gain little by searching with the full-length of any of its members. Too much evolution has happened over its full length — the ‘history’ of most of it has been lost. However, certain regions of the alignment have been constrained throughout evolutionary history. They are somehow very ‘important’ to the sequence — functionally, structurally, or whatever — we can use them to find other sequences with similarly constrained regions, if we can somehow ‘capture’ that information with some method more sensitive than simple motifs.

Enter two-dimensional consensus techniques. The basic idea is to tabulate how often every possible residue occurs at each position. You saw this last week with DNA matrix consensus descriptions of promoter, splice site, and terminator regions. With proteins the information is stored in a matrix twenty residues wide by the length of your pattern. Does this remind you of anything besides last week’s DNA matrices? We’re talking about the same concept as a amino acid substitution table or scoring matrix, in other words a very special log-odds PAM style table — a matrix custom built based on a specific pattern in a collection of related sequences.

This powerful approach is called Profile analysis (Gribskov, et al., 1987 and 1989). It, and later refinements thereof (e.g. Eddy, 1996 and 1998) is great for discovering distantly related proteins and structural motifs. John Devereux, past president of GCG, wrote an excellent overview essay of the method in the GCG Program Manual; there’s also a great review of HMM profiles in there. It’s worth the time to read these sections at some point (“genmanual” from the command line or the “Help” buttons in SeqLab). The strategy is used after you’ve prepared and refined as much as possible (and saved!) your multiple sequence alignment of significantly similar sequences or regions within sequences. A good plan is to find similar sequences to a newly sequenced section of DNA using traditional database searching techniques and then align all of the
significantly similar translated sequences or domains. Next, run the aligned sequences through the Profile package to generate a profile of the family — a very sensitive and tremendously powerful probe for further searching analyses.

Profile methods enable the researcher to recognize features that may otherwise be invisible to individual sequence members. Profile analysis uses the full information content of an alignment. The greatly enhanced information content, over that of individual sequences, has the potential to find similar motifs in sequences that are only distantly related, more so than any other class of search algorithm. All other methods of describing an alignment such as consensus or pattern description either through away too much information or become too ambiguous. Profiles achieve additional sensitivity with a two-dimensional weight matrix approach versus a simple one-dimensional string technique. Furthermore, profiles are a special type of two-dimensional weight matrix in which conserved areas of the alignment receive the most importance and variable regions hardly matter! Even the popular program PSI-BLAST (Altschul et al. 1997) uses profiles.

A distinct advantage is further manipulations and database searches consider evolutionary issues by virtue of the Profile algorithms. The creation of gaps is highly discouraged in conserved areas and occurs easily in variable regions in subsequent profile alignments and searches. This occurs because gaps are penalized more heavily in conserved areas than they are in variable regions. Furthermore, the more highly conserved a residue is, the greater its position-specific matrix score is. These two factors are what give profiles so much power. The matrix and its associated consensus sequence are not based merely on the positional frequency of particular residues, but rather utilize the evolutionary conservation of amino acid substitutions within the alignment based on the scoring matrix specified, by default the BLOSUM62 table (Henikoff and Henikoff, 1992) (other substitution matrices can also be specified). Therefore, the resultant consensus residues are the most evolutionarily conserved, rather than just statistically the most frequent.

‘Traditional’ profiles

The Gribskov et al. (1987) method is implemented in the Wisconsin Package with a series of five programs:

- ProfileMake — creates the profile from a multiple sequence alignment.
- ProfileSearch — searches other sequences (or the database) with a profile.
- ProfileSegments — aligns the output list of a ProfileSearch.
- ProfileGap — aligns individual sequences to a profile.
- ProfileScan — searches sequences against a validated profile library built by Gribskov and based on PROSITE.

A profile, and its inherent consensus, is created with the GCG program ProfileMake. When you create a profile all of its members should be appropriately weighted to even out each contribution. Each sequence, by default, contributes an equal importance, i.e. weight, to the profile. This may or may not be appropriate for your situation. Consider a multiple sequence alignment with several very similar sequences and a few more
dissimilar divergent ones. In this case, the contribution of the more divergent sequences would be ‘lost’ among the overpowering signal of all the similar ones. It may be appropriate to increase the weight of the more divergent sequences to even out the sequences’ contribution. This is often done in an ‘ad-hoc’ manner, although a similarity dendrogram, can aid the decision. Those clusters with less than their ‘fair share’ of contribution, have their weights increased. To figure out the appropriate weighting factors, choose the largest cluster, assign each member a weight of one and then propagate that up throughout the clusters. (If you’re interested, I can explain further personally.) The process of weighting your sequences appropriately and repeatedly searching the database with your profile and then adjusting the weights and including or excluding subsequent members of the profile is known as “validating” your profile. If using Traditional Profile analysis in your own research, following the validation procedures outlined in the GCG Program Manual in the ProfileScan description is very prudent. A ‘motif’ style profile library based on the PROSITE Dictionary of Protein Sites and Patterns has been prepared by Gribskov and made available within the GCG system. The program ProfileScan searches your query protein sequence against this library. The present version of GCG has 629 validated profiles in its ProfileScan library.

I created a small profile of just the P-Loop region to show you how to interpret a profile matrix. The greatest amount of conservation of the P-Loop region is centered about residue position twenty or so. What happens if I prepare a profile around just this region? What does it look like? It’s a big table of numbers that doesn’t make a whole lot of sense at first inspection, but it is a tremendously powerful tool in subsequent analyses. As described above, other programs can read and interpret this alignment customized scoring matrix to perform very sensitive database searches and further alignments by utilizing the information within the matrix that penalizes misalignments in phylogenetically conserved areas more than in variable regions.

Let’s check it out next:
On closer inspection, the matrix begins to make some sense. Across the top are all possible residues. The first column is that residue that received the highest score in the program — the consensus. But notice the interior of the matrix. Numbers bounce all over the place, from 150 to -87. What’s that all about? Well, without going into all the mathematics involved, based on the alignment we fed it and the initial scoring matrix used (by default the BLOSUM62 matrix but you can specify others) the program has scaled those positions which are most important up and those positions least important down. For instance the Threonine at position 27 in our alignment is the only residue absolutely conserved throughout — it gets the highest score! The Aspartate at position 22 substituted with a Tryptophan would never happen, hence the -87 score. Tryptophan is the most conserved residue in the BLOSUM matrix series and the Aspartate is conserved at all positions in our alignment that have residues at that position — the negative matrix score of any substitution to Tryptophan times the high conservation at that position for Aspartate equals the most negative score in the profile. How about those positions where the conservation is not as striking? Position 16 is a good one to pick on. Valine is the assigned consensus residue because it has the highest score, 37, but Glycine also occurs several times, a score of 20. However, other residues are ranked in the substitution matrices as being quite similar to Valine; therefore Isoleucine and Leucine also get similar scores, 24 and 14, and Alanine occurs some of the time in the alignment so it gets a comparable score, 15. But realize that all of these numbers are way less than the highest numbers in the matrix — because the position is not well conserved all the values are fairly mediocre at that position.

OK, but what about the last two columns in the matrix, and the last row? The last row is the composition of the whole profile. Our alignment has twenty Alanines overall and no Cysteines — big deal. However, the last two columns are very important! They relate to gap penalties in any subsequent analysis with this particular profile. I stated that gaps are more easily introduced into variable regions than conserved regions in profile analysis. Well, this is where that comes from. The first column is the gap opening penalty and the second is the gap extension penalty for that particular spot in any subsequent analysis (both as a percentage). Unlike other implementations of dynamic programming, the penalties are not constant throughout the length of the profile. Those regions where conservation is highest, receive 100% of the assigned gap penalty. Those regions with less conservation, receive less gap penalty. Here, everywhere else only gets 12% of the assigned gap penalty!

**HMMER — Hidden Markov Modeling and profiles**

As powerful as Gribskov style profiles are, they require a lot of time and skill to prepare and validate, and they are heuristics based. An excess of subjectivity and a lack of formal statistical rigor also contribute as drawbacks. In collaboration with the author, Sean Eddy (1996 and 1998), GCG has incorporated the HMMER (pronounced “hammer”) package into the Wisconsin Package. HMMER uses Hidden Markov modeling, with a formal probabilistic basis and consistent gap insertion theory, to build and manipulate HMMER profiles and profile databases, to search sequences against HMMER profile databases and visa versa, and to easily create multiple sequence alignments using HMMER profiles as a ‘seed.’ Again, GCG has taken the time to
write an excellent essay in the Program Manual on HMMER, what Hidden Markov Models are, and how the algorithms work. I urge you to read it, as well as each individual HMMER program description, at some point. The ‘take-home’ message is HMMER profiles are much easier to build than traditional profiles and they do not need to have nearly as many sequences in their alignments in order to be effective. Furthermore, they offer a statistical rigor not available in Gribskov profiles, and they have all the sensitivity of any profile technique.

Coding DNA issues

When dealing with very similar sequences, it is usually best to align DNA sequences along with their corresponding proteins (the “Group” function is very helpful for this). Phylogenetic analyses is then performed on the DNA rather than on the proteins. This is especially important when dealing with datasets that are quite similar since the proteins may not reflect many differences hidden in the DNA. Furthermore, many people prefer to run phylogenetic analyses on DNA rather than protein regardless of how similar they are — the multiple substitution models are much more robust for DNA. In fact, many phylogenetic inference algorithms do not even take advantage of amino acid similarity when dealing with protein sequences; they only count identities! However, the more diverged a dataset becomes, the more random third and eventually first codon positions become, which introduces noise (error) into the analysis. Therefore, often third positions and sometimes first positions are masked out of datasets. Just like in most of computational molecular biology, one is always balancing signal against noise. Too much noise or too little signal both degrade the analysis to the point of nonsense.

The logic to this paired protein and DNA alignment approach is as follows:

1) The easy case where you can align the DNA directly. If the DNA sequences are directly alignable because they are quite similar, then merely create your DNA alignment. Next use the “Edit” menu “Translate” function and the “align translations” option to create aligned corresponding protein sequences. Select the region to translate based on the CDS reference in each DNA sequence’s annotation. Be careful of CDS entries that do not begin at position 1 — the GenBank CDS feature annotation “/codon_start=” identifies which position the translation begins within the first codon listed. You may also have to trim sequences down to just the relevant gene, especially if they’re genomic. You’ll have to change their protections with the padlock icon if this is the case. Group each protein to its corresponding DNA sequence so that subsequent manipulations will keep them together.

2) The way more difficult case where you need to use the protein sequences to create the alignment because the DNA is not directly alignable. In this case you need to load the protein sequences first, create their alignment, and then load their corresponding DNA sequences. You can find the DNA sequence accession codes in the annotation of the protein sequence entries. Next translate the unaligned DNA sequences into new protein sequences with the Edit-Translate function using the “align translations” option and Group these to their corresponding DNA sequences, just as above. However, this time the DNA along with their translated sequences are not aligned as a set, just the other protein set
is aligned. Also, Group all of the aligned protein dataset together, separately from the DNA/aligned translation set. Now comes the manual part; painstakingly rearrange your display to place the DNA, its aligned translation, and the original aligned protein sequence side-by-side and then manually slide one set to match the other. Use the “CUT” and “PASTE” buttons to move the sequences around. When pasting realize that the “Sequence clipboard” contains complete sequence entries, whereas the “Text clipboard” only contains sequence data, amino acid residues or DNA bases as the case may be. The translated sequence entries can be “CUT” away after they’re aligned to the rest of the set. Merge the newly aligned sequences into the existing alignment Group as you go and then start on the next one. It sounds difficult, but since you’re matching up two identical protein sequences, the DNA translation and the original aligned protein, it’s really not too bad. The Group function keeps everything together the way it should be so that you don’t lose your original alignment as you space residues apart to match them up to their respective codons. Some codons may become spaced apart in this process and will have to be adjusted afterwards. As usual, save your work often.

**Week 6 Tutorial: A ‘Real-Life’ Project Oriented Approach. Advanced Multiple Sequence Analysis.**

Activate and/or log on to the computing workstation you are sitting at and then log onto Mendel with an X-tunneled ssh session. Remember that we do this on the Conradi PC’s with the combination SSH and XWin32. Review the Biology Computing Facility Help pages if you’ve forgotten how. If using an xterm window on Mac OSX or UNIX/Linux then issue the following command (the X has to be capitalized and replace “user” with your account name):

```
> ssh -X user@mendel.csit.fsu.edu  (Do not issue this command on MS Windows SSH/XWin32!)
```

**Preliminary preparations**

Change your directory (cd) from ‘home’ to last week’s subdirectory. List that directory (ls) and check out the files left over from last week’s tutorial. Look through them (more) and remove (rm) any that you don’t want to save. Next, change directory back to your home directory, create a subdirectory (mkdir) for this week’s tutorial data, and then change directory into it.

After you’ve taken care of these file maintenance chores launch SeqLab with the following command (but remember with SSH/XWin32 you need to launch “xclock &” first):

```
> seqlab &
```

Next, it will again be helpful to change your SeqLab working directory to your present location so that everything you do today will automatically be saved in your new directory rather than last week’s directory. Do this with SeqLab’s “Options” “Preferences…” “Working Dir…” button.
Now verify that you are in SeqLab’s “Main List” “Mode:" and start a new list to contain this week’s data. Therefore, select “New List..” from the “File” menu and give your new list an appropriate name. It’s not essential to use the file name extension “.list” but it’s a good idea. Check “OK.”

You should now be in List Mode with an empty window. Go to the “File” menu and select “Add Sequences From” “Sequence Files..” Use the “Directories” column to move from your present directory over to Week Five’s subdirectory and then replace the text in the “Filter” text box with the name or a wildcard specification that will identify your FastX output file from the Swiss-Prot search done in that tutorial. Press the “Filter” button and then select the correct entry. Press the “Add” button to add it into your new empty list file and then “Close” the “Add Sequences” window. Select the FastX search output list file and switch “Mode:” to “Editor.” The next prompt requires some thought when loading the results of a similarity search into the SeqLab Editor. You’ll be asked whether to “Modify the sequences” or “Ignore all attributes” in a “List file attributes set” window. The answer will depend on the type of alignment you are creating and the biological questions that you asking. In many cases, especially if you are asking phylogenetic questions, then you will not want to modify the sequences. In today’s case you will not want to trim them down, so select “Ignore all attributes” to load their full length and maximize available signal. However, if dealing with extremely diverse sequences and/or just the domains of sequences, then trimming the sequences down to those most conserved portions identified by the similarity search can be very helpful.

Similarity search results are loaded into the Editor in order of similarity to your query, from most similar to least similar. Our FastX Expectation cutoff was 0.10; therefore, the bottom of the list contains sequences quite dissimilar to your query. Furthermore, we want to deal with a dataset that we can easily align and analyze in one lab session. Therefore, we need to get rid of most of the entries that just loaded. Temporarily switch to your xterm window behind your SeqLab session. Change directory into your week five subdirectory and take a look at your FastX output file. Try to find an area in it where the Expectation scores markedly increase, such as where they jump from $10^7$ to $10^3$ in my elongation factor example below:

```
SW:EF1A_PYRAE Begin: 58 End: 418
! O93729 pyrobaculum aerophilum. elon... 265 1222 269 268.7 4.2e-08
SW:HBS1_YEAST Begin: 206 End: 359
! P32769 saccharomyces cerevisiae (ba... 244 520 246 244.1 9.8e-07 Here!
SW:EFTU_AQUPY Begin: 49 End: 323
! O50293 aquifex pyrophilus. elongati... 171 459 185 186.0 0.0017
SW:EFTU_AQUAE Begin: 49 End: 323
! O66429 aquifex aeolicus. elongation... 171 459 185 186.0 0.0017
SW:EFTU_CORGL Begin: 52 End: 294
! P42439 corynebacterium glutamicum (... 172 390 183 184.2 0.0021
```

This will often identify the jump from one group of homologues to another. In my example it looks like this is a tentative break between orthologues and paralogues. If you can’t identify such an area in your FastX output, contact the lab instructor for some assistance. Note the most similar entry that belongs to this group of less similar sequences, in my example “SW:EFTU_AQUPY.” Also scan over your FastX file for entries that are flagged with the warning comment “! vs rev query.” This means that the similarity was found on the
reverse-complement strand of your DNA query, not the forward strand. These will cause endless headaches and should certainly be excluded from the analysis! They are not homologues of your system.

Return to your SeqLab Editor display and select that top-most less similar sequence entry, as described above. You can use the “Edit” menu “Select by Name . . .” function to find the entry if it’s not obvious. “Close” the “Select by Name” window after pressing the “Select” button. Next, scroll to the bottom of the alignment, press the <shift> key and <click> on the bottom-most entry. This should select the entire range of sequences from that top-most less similar entry identified above to the very bottom of the FastX output list. Press the “CUT” icon button to get rid of them from the analysis. No go back and “CUT” any entries where the similarity was found on the reverse-complement strand. Finally, see how many entries you have left — we want a final dataset with between 20 and 50 entries. If you still have way more that that, indiscriminately “CUT” more entries from the bottom of the dataset until you have a manageable number. Quickly double click on some of the entries’ names left to see the database reference descriptions for them (or click on the “INFO” button). The following graphic shows the Editor display after loading my elongation factor example FastX file:

Now would be a good time to go to the “File” menu and save the RSF file. Give it a name that makes sense to you, rather than the default name based on the bottom-most sequence, but do retain the “.rsf” extension.

A ‘quick and dirty’ PROSITE scan — GCG’s Motifs search

Before aligning these sequences let’s look over them for consensus patterns from PROSITE. This does not work very well with sequences that have gaps in them, as if the gaps occur in a motif it will not be recognized, so it needs to be run before alignment. Start the Motifs program by selecting all of the protein entries’ names in SeqLab, using the “Edit” menu “Select All” function and then going to the “Functions” “Protein Analysis” menu and picking “Motifs . . .” The "Motifs" program window will be displayed. Check the “Save results as features in file motifs.rsf” button in the “Motifs” program window. This file will contain annotation discovered
by the program and we'll use it below. None of the other options are required for this run, so press the “Run” button. After a few moments you should get output. The first file displayed, “motifs.rsf,” isn't very interesting to read (it's SeqLab's “Rich Text Format”) so “Close” it and use the “Output Manager” to display the file with the “.motifs” extension instead. Carefully look over the text file that is displayed. Notice the sites in your Motifs output file that have been characterized and the extensive bibliography associated with them.

“Close” the “Motifs” output window when you've looked it over and then load the “motifs.rsf” file into SeqLab. This will add the feature annotation created with the -RSF option. The location of the PROSITE signatures will now be included in the Editor sequence display. Use the “SeqLab Output Manager” to do this. Select the file “motifs.rsf,” then press the “Add to Editor” button and specify “Overwrite old with new” to take the new RSF feature file and merge it with the old RSF file in the open Editor. “Delete from disk” the “motifs.rsf” file after adding it to the Editor and then Close” the “Output Manager.” Look at your display using “Features Coloring” or “Graphic Features” to display the new annotation and see if you can recognize the differences. Quickly <double-click> on one of the new areas or use the “Windows” menu “Features” button to read about the added annotation. My dataset example is illustrated below using “Features Coloring” now annotated with its original database features as well as the new Motifs patterns:

![Motifs Output Window](image)

MEME

As mentioned in the Introduction, MEME is used to discover unrecognized patterns in an unaligned dataset. To run MEME be sure all of the sequences are still selected in the Editor window and then launch “MEME...” off of the “Functions” “Multiple Comparisons” menu. A "Which selection" window may pop up asking if you want to use the "Selected sequences" or "Selected region;" choose "Selected sequences" to run the program on the full length of all the sequences. In most cases the default parameters will work fine but the algorithm can be sped up at the cost of sensitivity by decreasing the number of motifs to be found, by
restricting the number of motifs found to exactly one in each sequence, and/or by decreasing the allowable motif window size. Again, I suggest reading the relevant GCG Program Manual chapters by using SeqLab’s “Help” buttons. Press the “Run” button to execute the program. MEME will take quite a while to process your sequences. Do not wait for it to finish. Go on with the rest of the tutorial and then return to this section when it finally does finish, so that you can do the next two paragraphs work.

MEME output consists of two files; a .meme readable text file and a .prf multiple profile text file. MotifSearch will scan any original ‘training’ dataset that you created the profiles with. This will annotate those regions that MEME discovered in your SeqLab Editor RSF file. After alignment the MEME motifs that are alignable will all line up. Go to the “Database Sequence Searching” menu and select “MotifSearch. . .” Specify your “query profile(s),” the one you just made, and change the “Search set” to “Remove from Search Set” “PIR:*” and “Add Main List Selection. . .” the RSF dataset that you now have loaded in the Editor. “Close” the “Build MotifSearch’s Search Set” window. Be sure to activate “Save motif features to the RSF file.” Press “Run.” The output will quickly return with the .rsf file on top. Don’t bother trying to read it; just “Close” it. It contains the feature data discovered by MEME in your dataset. The .ms file contains the readable results of the search in list file format with Expectation value statistics and the number of motif hits for each fit. After the list file portion a “Position diagram” schematically describes the hits in each sequence. Take a moment to look it over by pressing the “Display” button in the Output Manager and then “Close” it.

Use the Output Manager to merge the new “motifsearch.rsf” feature file with the existing data already in the open SeqLab Editor. This will add the newly discovered MEME feature annotation created when you activated the MotifSearch -RSF option. The location of each motif will now be included in the Editor sequence display. To do this, again use the extremely important “Add to Editor” Output Manager function. As above, specify “Overwrite old with new” in the next window when prompted. “Delete from disk” the “motifsearch.rsf” file and then “Close” the “Output Manager” after loading your new RSF file. One thing that you may immediately notice is that sequences from your dataset that don’t have all of the motifs discovered by MotifSearch are now placed at the bottom of the dataset. There’s a good chance that they do not belong in the dataset and should be removed. The Swiss-Prot database uses very good naming conventions, you can read the sequences’ “INFO,” and you should evaluate the domain structure to decide whether they should be “CUT” out of your dataset or not. It’s OK to leave a few outliers, but they need to be true homologues. Confer with your lab instructor if you’re confused. Change “Display;” to “Graphic Features” and check out the additional annotation. The following figure illustrates my example using “Graphic Features” display at a “4:1” zoom ratio:
Performing the alignment — the PileUp program

Next, we need to align your protein sequences. Select all of the entries in the Editor window and then go to the “Functions” menu and select “Multiple comparison.” Click on “PileUp . . .” to align the entries. A “Which selection” window may pop up asking if you want to use the “Selected sequences” or “Selected region;” choose “Selected sequences” to run the program on the full length of all the sequences. A new window will appear with the parameters for running PileUp. Often you’ll accept all of the program defaults on a first run by pressing the “Run” button; however, here I am going to change the scoring matrix for the alignment from the default BLOSUM62 to the alternate BLOSUM30 matrix.

Depending on the level of divergence in a data set, better multiple sequence alignments can often be generated with alternate scoring matrices (the -Matrix command line option, specifying the desired matrix from the GCG logical directory GenMoreData) and/or different gap penalties. Beginning with GCG version 9.0, the BLOSUM62 (Henikoff and Henikoff, 1992) matrix file, “blosum62.cmp,” is used as the default symbol comparison table in most programs. Furthermore, appropriate gap creation and extension penalties are now coded directly into the matrix, though they can still be adjusted within the program if desired. This is a greatly improved situation over the normalized Dayhoff PAM 250 table (Schwartz and Dayhoff, 1979) and the program encoded penalty values that GCG formerly used. The BLOSUM series are more robust at handling a wider range of sequence divergence than the PAM tables ever were — the BLOSUM30 table being most appropriate for the most divergent datasets, ranging to the BLOSUM100 table for the most conserved datasets. Since these sequences are from quite a wide spectrum of organisms, we’ll use the BLOSUM30 matrix. Your particular dataset may not require the BLOSUM30 matrix, but I want you to see how it’s done.

Therefore, click on the “Options” button. To specify the BLOSUM30 matrix select the check button next to and click on the “Scoring Matrix . . .” box in the “Pileup Options” window. This will launch a “Chooser for Scoring Matrix” window from which you can select the BLOSUM30 matrix file, “blosum30.cmp.” Double-
click the matrix’s name to see what it looks like; click “OK” to close both windows. Scroll through the rest of “PileUp Options” window to see all those available. “Close” it when finished. Be sure that the “How:” box says “Background Job” and press then “Run” in the “PileUp” window to launch the program.

The program will first compare every sequence with every other one. This is the pairwise nature of the program, and then it will progressively merge them into an alignment in the order of determined similarity, from most to least similar (Feng and Doolittle, 1987). The window will go away and then, after a few moments, depending on the complexity of the alignment and the load on the server, new output windows will automatically display. The top window will be the Multiple Sequence Format (MSF) output from your PileUp run. Notice the BLOSUM30 matrix specification and the default gap introduction and extension penalties associated with that matrix, 15 and 5 respectively. As mentioned above, in most cases the default gap penalties will work fine with their respective matrixes, though they can be changed if desired. In fact, see below on improving regions within alignments, where it is absolutely required.

Scroll through your alignment to check it out and then “Close” the window afterwards. My much abridged output file example follows below. Notice the interleaved character of the sequences, yet they all have unique identities, addressable through their MSF filename together with their own name in braces, {name}:

```
PileUp of: @/users/thompson/.seqlab-mendel/pileup_5.list
Symbol comparison table: /usr/local/gcg/gcgcore/data/moredata/blosum30.cmp  Com
pCheck: 8599
GapWeight: 15
GapLengthWeight: 5
pileup_5.msf  MSF: 690  Type: P  February 15, 2003 10:27  Check: 5328 ..

Name: ef12_horvu   Len:  690  Check:  6172  Weight:  1.00
Name: ef11_horvu   Len:  690  Check:  7747  Weight:  1.00
Name: ef1a_wheat   Len:  690  Check:  5515  Weight:  1.00
Name: ef1a_vicfa   Len:  690  Check:  3560  Weight:  1.00
Name: ef1a_orysa   Len:  690  Check:  6926  Weight:  1.00
Name: ef1a_maize   Len:  690  Check:  7241  Weight:  1.00
Name: ef1a_tobac   Len:  690  Check:  4161  Weight:  1.00
Name: ef1a_haln1   Len:  690  Check:  7917  Weight:  1.00
Name: ef1a_halma   Len:  690  Check:  9036  Weight:  1.00
Name: ef1a_metja   Len:  690  Check:  9736  Weight:  1.00
Name: ef1a_metva   Len:  690  Check:  6161  Weight:  1.00
Name: ef1a_metth   Len:  690  Check:  539  Weight:  1.00
Name: ef1s_porpu   Len:  690  Check:  5488  Weight:  1.00
Name: gsp1_human   Len:  690  Check:  3629  Weight:  1.00
Name: hbs1_yeast   Len:  690  Check:  8166  Weight:  1.00

.Mouse
1
50
ef12_horvu   --------- --------- --------- --------- ---------
ef11_horvu   --------- --------- --------- --------- ---------
ef1a_wheat   --------- --------- --------- --------- ---------
ef1a_vicfa   --------- --------- --------- --------- ---------

.Mouse
1
50
ef1a_arcfu
ef1a_metth
ef1s_porpu
--------- --------- --------- --------- ---------
--------- --------- --------- --------- ---------
--------- --------- --------- --------- ---------
Return to the listing of sequence names near the top of the file. This listing contains an important number called the checksum. All GCG sequence programs use this number as a unique sequence identifier to identify corrupted sequences. There is a checksum line for the whole alignment as well as individual checksum lines for each member of the alignment. If any two of the checksum numbers are the same, then those sequences are identical. If they are, an editor can be used to place an exclamation point, "!", at the start of the checksum line in which the duplicate sequence occurs. Exclamation points are interpreted by GCG as remark delineators; therefore, the duplicate sequence will be ignored in subsequent programs. 

Another important number on the individual checksum lines is the "Weight" designation. It determines how much importance each sequence contributes to a profile made of the alignment. Sometimes it is worthwhile to adjust these values so that the contribution of a collection of very similar sequences does not overwhelm the signal from a few more divergent sequences. In the SeqLab interface the "Sequence Info . . . " window can be used to accomplish this, or you can use a simple text editor. However, we will not be bothering with it here.
Scroll through the alignment and then “Close” its window. Again use the “Output Manager” to “Add to Editor” and “Overwrite old with new,” to take your new MSF output and merge it with the old RSF file in the open Editor. This will keep the feature annotation intact, yet renumber all of its reference locations based on the inclusion of gaps in the alignment. “Close” the “Output Manager” after loading your new alignment. The next window will contain PileUp’s cluster dendrogram, in my EF-1 example, the following graphic:

PileUp automatically creates this dendrogram of the similarity clustering relationships between the sequences. It can be very helpful for adjusting sequence Weight values, which even out each sequences’ contribution to a profile. The lengths of the vertical lines are proportional to the differences in similarity between the sequences. However, realize that this tree is not an evolutionary tree, and it should never be presented as one. No phylogenetic inference optimality criteria algorithm, such as maximum likelihood, least-squares fit, or parsimony, nor any molecular substitution, multiple-hit correction models, such as Jukes-Cantor, Kimura, or any other subset of the GTR (General Time Reversible) model, nor any site rate heterogeneity models such as a Gamma correction, are used in its construction. (It is roughly an uncorrected UPGMA tree, prone to all the same errors seen in UPGMA. Therefore, if the rates of evolution for each lineage were exactly the same and there was no saturation of residue positions, then it could represent a ‘true’ phylogenetic tree, but this is seldom the case in nature.) PileUp’s dendrogram merely indicates the relative similarity of the sequences based on the scoring matrix used, by default the BLOSUM62 but the BLOSUM30 in our runs, and, therefore, the clustering order used to create the alignment.

If desired, you can directly print from any SeqLab graphics Figure windows to PostScript files by picking “Print...” “[Encapsulated] PostScript File” “Output Device:” Name the output file anything you want; click “Proceed” to create an EPSF output in your current directory. To actually print this file you may need to
transfer it to a local machine attached to a PostScript compatible printer, unless you have access to Mendel's system printer. (All Macintosh compatible laser printers run PostScript by default. Carefully check any laser printer connected to a ‘Wintel’ system to be sure that it is PostScript compatible.) “Close” the dendrogram window.

Now notice that your residues align by color. My Editor display looks like the following after loading the MSF file using “Residue Coloring” and a “1:1” zoom ratio:

Notice the nice columns of color representing columns of aligned residues. Change the “Display:” box from “Residue Coloring” to “Graphic Features.” Now the display shows a schematic of the feature information from each entry, as well as all of the motifs discovered by the programs Motifs and MotifSearch, and will look like the following, at a “4:1” zoom:
Remember, quickly <double-clicking> on any of the color coded feature regions in the Editor display will produce a “Features” window where more information is available about that particular feature by selecting the Feature entry in the new window. Clicking once in the colored region and then using the “Features” option from the “Windows” menu will also produce the “Features” window. Now would also be another good time to save your work as an updated RSF file! Use the same name as before and “Overwrite” the previous file; there’s no need to save multiple versions of the RSF file.

Visualizing conservation in multiple sequence Alignments

The most conserved portions of an alignment are those most resistant to evolutionary change, often due to some type of structural constraint. To easily visualize the positional conservation of a multiple sequence alignment use the graphics program PlotSimilarity. The program draws a graph of the running average similarity along a group of aligned sequences (or of a profile with the -Profile command line option). The PlotSimilarity peaks of a protein alignment represent the most conserved areas of the alignment, but even more so, those areas most resistant to evolutionary change due to the algorithm’s use of the BLOSUM matrix in its calculations. PlotSimilarity is also a nice way to see those areas of an alignment that may need improving by pointing out the most variable regions. Furthermore, PlotSimilarity can be helpful for ascertaining alignment quality by noting changes in the overall average alignment similarity and in those regions of conservation within the alignment, as it is adjusted and refined.

Select all of your sequence entry names and then go to the “Functions” menu and under the “Multiple comparison” section choose “PlotSimilarity...” I recommend changing some of the program defaults so choose “Options” in the program window. Check “Save SeqLab colormap to” and “Scale the plot between:” the “minimum and maximum values calculated from the alignment.” The first option’s output file will be used in the next step. The second specification launches the program’s command line -Expand option. This blows up the plot, scaling it between the maximum and minimum similarity values observed, so that the entire graph is used, rather than just the portion of the Y-axis that your alignment happens to occupy. The Y-axis of the resulting plot uses the similarity values from whichever scoring matrix you used to create your alignment unless you specify an alternative. The default matrix, BLOSUM62, begins its identity value at 4 and ranges up to 11; mismatches go as low as -4. “Close” the “Options” window; notice that the “Command Line:” box now reflects your updated options. Click the “Run” box to launch the program. The output will quickly return. “Close” the plotsimilarity.cmask display and the “Output Manager” and then take a look at the similarity plot. The graphic from my elongation factor example follows next:
My example shows a good deal of sequence similarity. Exceptionally strong peaks are seen centered around positions 260 and 630, among many others nearly as strong. The ordinate scale is dependent on the scoring matrix used by the program that created the alignment, here the BLOSUM30 table, which ranges in score from -7 to +20. The dashed line across the middle shows the average similarity value for the entire alignment, here about 2.25. Make a PostScript file of this plot too, if desired. As before, to print a SeqLab graphics Figure to a PostScript file: select “Print . . .” off the Figure window, choose “Output Device:” “[Encapsulated] PostScript File,” and click “Proceed,” to create EPSF output. Regardless of whether you print this plot or not, take notes of where the similarity significantly falls off within and at the beginning and end of the alignment. In my example above, this is the first 160 residues or so, a few regions around 350, 400, 470, and 600, and about the last 30 residues. “Close” the PlotSimilarity window after noting where these deepest valleys, the least similar regions of the alignment, lay.

Now go to the “File” menu and click on “Open Color Mask Files.” This will produce another window from which you should select your new “plotsimilarity.cmask” file; click on “Add” and “Close” the window. This will produce a gray scale overlay on your sequences that describes their regional similarity where darker gray corresponds to higher similarity values. My example alignment, using a zoom factor of 4 to 1, looks like the following. Notice the strong conservation peak centered around residue 250 in the alignment, one of EF-1γ’s GTP binding regions:
Improving alignments within SeqLab

The beauty of this representation is you can now easily select those regions of low similarity to try to improve their alignment automatically. This is possible because of PileUp's incredibly effective -InSitu command line option that can realign regions within an alignment. Be sure that all of your sequences are selected and then zoom back in on your alignment to “1:1” so that you can see individual residues and then scroll to the carboxy end. It’s best to start at the carboxy termini in this process so that the positions of the low similarity regions do not become skewed as you proceed through the procedure. Now select a region of low similarity across the complete sequence set, that is, the low similarity region of all of the entries. This can be done using the mouse if it’s all on the screen in front of you, which is not the case in our examples. Therefore, use the “Edit” “Select Range” function (determine the positions by placing your cursor at the beginning and end of the range to be selected and noting the column number in the lower left-hand of the Editor display). Once all of your sequences and the region that you wish to improve are selected, go to the “Functions” menu and again select “Multiple comparison.” Click on “PileUp . . .” to realign all of the sequences within that region. (The “Windows” menu also contains a ‘shortcut’ listing of all of the programs that you have used in the current session; you can launch any of them from there as well.) You will be asked whether you want to use the “Selected sequences” or “Selected region,” it is very important to specify “Selected region.” This will produce a new window with the parameters for running PileUp. Next, be sure to click on “Options . . .” to change the way that PileUp will perform the alignment. In the “Options” window check the gap creation and extension boxes and change their respective values to much less than the default. Changing them to about a third the default value works pretty well for a start, so for the BLOSUM30 matrix change the values to “5” and “2” respectively. Most importantly, check “Realign a portion of an existing alignment;” this calls up the command line -InSitu option. Otherwise only that portion of your alignment selected will be retained in the output. Furthermore, we really don’t need another similarity dendrogram, so uncheck the “Plot dendrogram” box. “Close” the window and notice the new options in the PileUp “Command Line”: “Run” the program to
improve your alignment. The window will go away and your results will return very quickly since you are only realigning a portion of the alignment; new output windows will automatically display. The top window will be the MSF output from your PileUp run. Notice the BLOSUM30 matrix specified (others available through the options menu) and the lowered gap introduction and extension penalties of 5 and 2. Scroll through your alignment to check it out and then “Close” the window. The next window will be the “Output Manager.” Just like before, click on “Add to Editor,” and then specify “Overwrite old with new” in the new “Reloading Same Sequences” window to merge the new alignment with the old one and retain all feature annotation. This feature information may help guide your alignment efforts in subsequent steps. “Close” the “Output Manager” window after loading your new alignment.

Your alignment should now be a bit better within the specified region. Repeat this process in all areas of low similarity, again, working from the carboxy termini toward the amino end. Notice that all of the options that you last specified are retained by the program, so you don’t need to re-specify them. You can also save these run parameters so that they will come up in subsequent sessions by clicking on the “Save Settings” box in any of the program run windows. You may want to go to the “File” menu periodically to save your work using the “Save as . . .” function in case of a computer or network problem. It’s also probably a good idea to re-perform the PlotSimilarity and color mask procedure after going through the entire alignment to see how things have improved after you’ve finished the various -InSitu PileUps. If you discover an area that you can’t improve through this automated procedure, then it is time to either manually ‘correct’ it or ‘throw it away.’ Again, note those ‘problem’ areas and then switch back to “Residue Coloring.” This will ease manual alignment by allowing your eyes to work with columns of color.

Other things that can help manual alignment are “GROUP”ing and “ protections.” The “GROUP” function allows you to manipulate ‘families’ of sequences as a whole — any change in one will be propagated throughout them all. To “GROUP” sequences, select those that you want to behave collectively and then click on the “GROUP” icon right above your alignment. You can have as many groups as you want. The space bar will introduce a gap into the sequence and the delete key will take a gap away. However, you can not delete a sequence residue without changing that sequence’s (or the entire alignment’s) “ protections.” Click on the padlock icon to produce a “ Protections” window. Notice that the default protection allows you to modify “Gap Characters” and “Reversals” only. Check “All other characters” to allow you to “CUT” regions out of your alignment and/or delete individual residues and then click “OK” to close the window. A very powerful manual alignment function can be thought of as the ‘abacus’ function. To take advantage of this function select the region that you want to slide and then press the shift key as you move the region with the right or left arrow key. You can slide residues greater distances by prefacing the command keystrokes with the number of spaces that you want them to slide.

Make subjective decisions regarding your alignment. Is it good enough; do things line up the way that they should? If, after all else, you decide that you just can’t align some region, or even an entire sequence, then perhaps get rid of it with the “CUT” function. Another alternative is the mask function that I will describe next.
week. Cutting out an entire sequence may leave some columns of gaps in your alignment. I decided to get rid of the two outliers in my example. If this happens to you, then reselect all of your sequences and go to the “Edit” menu and select “Remove Gaps . . .” “Columns of gaps.” Notice the extreme amino and carboxy ends of the alignment. Amino and carboxy termini seldom align properly and are often jagged and uncertain. This is fairly common in multiple sequence alignments and subsequent analyses should probably not include these regions. If loading sequences from a database search, allowing SeqLab to trim the ends automatically based on beginning and ending constraints considerably improves this situation. Overall, things to look for include columns of strongly conserved residues such as tryptophans, cysteines, and histidines, important structural amino acids such as prolines, tyrosines and phenylalanines, and conserved isoleucine, leucine, valine substitutions; make sure they all align. After you have finished tweaking, evaluating, and readjusting your alignment to make it as ‘satisfying’ as possible, change back to “Feature Coloring” “Display.” Those features that are annotated should now align perfectly. This is another way to assure that your alignment is as biologically ‘correct’ as possible. Everything you do from this point on, and especially later if you use alignments to ascertain molecular phylogenies, is absolutely dependent on the quality of the alignment! You need a very clean, unambiguous alignment that you can have a very high confidence in — truly a biologically meaningful alignment. Each column of symbols must actually contain homologous characters.

Many other alignment editors are available for cleaning up multiple sequence alignments. However, I think that you will find SeqLab most satisfying, and only using a GCG compatible editor assures that the format will not be corrupted. If you do make any changes to a GCG sequence data file with a non-GCG compatible editor, you must reformat the alignment afterwards. However, reformatting GCG MSF or RSF files requires a couple of tricks. If this step is not done exactly correct, you will get very weird results. If you do need to do this for any reason, you must use the appropriate Reformat option (either -MSF or -RSF respectively) and you must specify all the sequences within the file using the brace specifier, i.e. “{“,” for example:

```plaintext
> reformat -msf your_favorite.msf(*)
```

You should never need to do this, unless for some reason you decide to edit an alignment with a non-GCG compliant editor; however, it may prove necessary in some situations. After reformatting, the new MSF or RSF file will follow GCG convention, with updated format, numbering, and checksums.

**SeqLab Editor on-screen annotation**

Something that you may want to do to your alignment after you’ve gotten it all cleaned up is to add text annotation to the display. Changing the entries’ names for presentation purpose might also be helpful. Both are easy to do in the SeqLab Editor. Double-click on an entry’s name to get its “Sequence Information” window and directly edit the name there. Selecting the entry name and then pressing the “INFO” icon does the same thing. To put text lines directly into your display go to the SeqLab “File” menu “New sequence . . .” entry and select the “Text” button to the “What type of sequence?” question. This will put a “NewText” line at the bottom of the Editor display that you can directly type annotation into. You can also add customized
“Graphic Features” and “Features Coloring” annotation with the “Windows” “Features” window. Select a desired region across an alignment and launch the “Features” window. Press “Add” to get a “Feature Editor” window where you can designate the feature’s “Shape:” “Color:” and “Fill:” as well as give the region a “Keyword:” and “Comments:.” Warning: You can add feature annotation to a region across an entire alignment, but you can’t delete or edit the annotation from the whole region collectively afterwards. You can only edit or delete feature annotation from an RSF file with the SeqLab Editor one sequence feature at a time!

Profile techniques

Traditional profiles ala Gribskov are created with ProfileMake. To run ProfileMake be sure that all of your aligned sequences are selected and then, based on your previous observations and your experimental objectives, select the longest, most conserved, overall sequence length available. Restrict the length of your profile so that jagged ends in the alignment are excluded. In SeqLab do this through the “Edit” “Select Range. . .” menu. “Select” and then “Close” the box. Another effective strategy is to develop multiple shorter profiles just centered about the similarity peaks of your alignment. These most likely will correspond to functional or structural domains in your protein. After your range is selected, use the “Functions” “Multiple Comparison” “ProfileMake. . .” menu and reply “Selected region” in the “Which selection” dialog box. You can also use the “Options. . .” menu from the “ProfileMake” dialog box to specify the -SeqOut command option by checking “Write the consensus into a sequence file” and giving it an appropriate name. This will generate a normal sequence file of the consensus in addition to the profile file. Play with any of the other options that you would like, such as the scoring matrix, and then “Close” the “Options” box and “Run” ProfileMake. After running ProfileMake, the top window returned will display your profile consensus sequence. The header contains information relating to the sequence’s creation through ProfileMake. “Close” the consensus window. The “Output Manager” will also list a “.prf” file. This is the profile itself. “Save As. . .” the profile in your “Output Manager” giving it an appropriate name that you can recognize; retain the “.prf” extension. “Close” the “Output Manager.”

ProfileSearch is launched through SeqLab with the “Functions” menu; select “Database Sequence Searching” “ProfileSearch. . .” Specify the “Query profile. . .” in the “File Chooser” and click “OK.” Search whichever protein database you prefer, though I suggest you just use NRL_3D for now, so the results will be quickly available. I like to run ProfileSegments separately after my ProfileSearch is done. Therefore, uncheck “ProfileSegments. . .” to prevent ProfileSearch’s output from automatically being passed to ProfileSegments. This way I can edit the ProfileSearch output file so that ProfileSegments only makes pairwise or multiple alignments of the sequences that I am interested in to my profile. Also, under “Options. . .” I like to use the -MinList option by changing “Lowest Z score to report in output list” from 2.5 to 3.5 or higher. MinList sets a list Z score cut-off value — a handy way to limit your output list size. “Close” the “Options” window and be sure that “How:” “Background Job” is selected and then click “Run.”

As in BLAST and FastA searches, ProfileSearch estimates a realistic significance parameter. In the case of profile searching it is a Z score based on the distance, in the number of standard deviations, from the rest of
the ‘insignificant’ database matches. ProfileSearch Z scores are normalized and reflect the significance of the results. Here rather than randomizing sequences to evaluate a Z score, as is done in Monte Carlo approaches, it is calculated based on all of the non-similar sequences from the database search, similar to the way that FastA calculates its Expectation values. Pay particular attention to the reported Z scores in the output. As with Monte Carlo approaches, Z scores below 3 are probably not worth considering, from around 4 to 7 may be interesting, and above 7 are most probably significant and should definitely be checked out further. You can find remote similarities that all other methods will miss using Profile analysis properly — it is extremely powerful. Your ProfileSearch will take a while to run; go on with the rest of the tutorial for now and then return to this portion after the ProfileSearch has finished.

Interpreting profile analysis — Why even bother; what can it show us?

Even though ProfileSearches require some work to setup and run — a meaningful multiple sequence alignment must be assembled and refined, ProfileMake needs to be run, and the search job itself takes quite a long time to run — it is well worth the bother. ProfileSearches are incredibly CPU intensive, together with HmmerSearch some of the most so in the GCG package, so be sure to submit them as early as possible (if launched from the command line, use the -$Batch option). When you return to a completed ProfileSearch take a careful look at the output. There is a good chance that other search algorithms will have missed some of the sequences listed as significant matches. If launched from SeqLab, the output will be located in your working directory and it will have a cryptic name of the form profilesearch_some-number.pfs. ProfileSearch finds all of the Elongation Factors in NRL_3D, all with Z scores >8. An abridged screen trace of my Elongation Factor 1[] example ProfileSearch output follows below:

!!SEQUENCE_LIST 1.0
(Peptide) PROFILESEARCH of: /home/thompson/seqlab/EF1a.prf Length: 554 to: NRL_3D:* 

Scores are not corrected for composition effects

    Gap Weight: 45.57
    Gap Length Weight: 0.51
    Sequences Examined: 20019
    CPU time (seconds): 147

Profile information:
(Peptide) PROFILEMAKE v4.50 of:
@/users/thompson/.seqlab-mendel/profilemake_8.list Length: 554
    Sequences: 97  MaxScore: 1562.93  February 16, 2003 16:02
    Gap: 1.00  Len: 1.00
    GapRatio: 0.33  LenRatio: 0.10
    input_8.rsf{EF12_HORVU}  From: 1  To: 554  Weight: 1.00
    input_8.rsf{EF11_HORVU}  From: 1  To: 554  Weight: 1.00 ...

Normalization:

Curve fit using 18 length pools
0 of 18 pools were rejected

Normalization equation:

Calc_Score = 298.80 * ( 1.0 - exp(-0.0024*SeqLen - 0.2521) )

Correlation for curve fit: 0.930
Z score calculation:
Average and standard deviation calculated using 19995 scores
24 of 20019 scores were rejected

\[ Z_{\text{score}} = (\text{Score}/\text{Calc}\_\text{Score} - 0.991) / 0.077 \]

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Strd</th>
<th>ZScore</th>
<th>Orig Length</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRL_3D:1TU1B</td>
<td>+ 20.21</td>
<td>534.57</td>
<td>397</td>
<td>translation elongation factor EF-Tu, chain B - Thermus aquaticus</td>
</tr>
<tr>
<td>NRL_3D:1EFT</td>
<td>+ 19.97</td>
<td>535.01</td>
<td>405</td>
<td>translation elongation factor EF-Tu (with guanosine-5’-(beta,gamma-imido) triphosph</td>
</tr>
<tr>
<td>NRL_3D:1D2EA</td>
<td>+ 18.45</td>
<td>506.09</td>
<td>397</td>
<td>elongation factor tu (ef-tu), chain A - bovine</td>
</tr>
<tr>
<td>NRL_3D:1EFUA2</td>
<td>+ 14.18</td>
<td>404.68</td>
<td>330</td>
<td>translation elongation factor EF-Tu, chain A, fragment 2 - Escherichia coli</td>
</tr>
<tr>
<td>NRL_3D:1EFUC2</td>
<td>+ 14.18</td>
<td>404.68</td>
<td>330</td>
<td>translation elongation factor EF-Tu, chain C, fragment 2 - Escherichia coli</td>
</tr>
<tr>
<td>NRL_3D:1EFM2</td>
<td>+ 9.32</td>
<td>220.84</td>
<td>130</td>
<td>elongation factor Tu (trypsin-modified with GDP), fragment 2 - Escherichia coli</td>
</tr>
<tr>
<td>NRL_3D:1ETU2</td>
<td>+ 8.78</td>
<td>222.94</td>
<td>141</td>
<td>translation elongation factor EF-Tu, domain I (with GDP), fragment 2 - Escherichia coli</td>
</tr>
<tr>
<td>NRL_3D:1A062</td>
<td>+ 4.85</td>
<td>159.04</td>
<td>100</td>
<td>calcium/calmodulin-dependent protein kinase (EC 2.7.1.123), fragment 2 - rat</td>
</tr>
<tr>
<td>NRL_3D:1UCH2</td>
<td>+ 3.72</td>
<td>127.69</td>
<td>64</td>
<td>ubiquitin c-terminal hydrolase uch-l3 (EC 3.1.2.15), fragment 2 - human</td>
</tr>
<tr>
<td>NRL_3D:1QASA3</td>
<td>+ 3.69</td>
<td>151.98</td>
<td>106</td>
<td>phospholipase c delta-1 (EC 3.1.4.11) phosphoinositide-specific phospholipase c delt</td>
</tr>
</tbody>
</table>

Notice the very clean demarcation in Z scores between the EF-1 homologues, with Z scores above around 8, and what is most likely just noise, with Z scores of around 5 and less.

The program ProfileSegments makes BestFit style alignments of the results of a ProfileSearch. A great option in ProfileSegments, -MSF, allows you to prepare a multiple sequence alignment of the ProfileSearch segments. This can be a very helpful strategy for merging ever-increasingly distant sequences into an existing alignment. The full information content of the profile including the importance of the conserved portions of your alignment is used in this alignment procedure. When checking out a ProfileSearch output, something I’ll often do is edit it to exclude (or comment out by placing an exclamation point at the beginning of the entry’s line) the insignificant entries at the bottom of the list and many of the sequences that I expected to be found by the search, except some positive controls; i.e. in my example most of the EF-1 homologues. If you ever do this, be sure not mess with the header portion of the file, it specifies your profile’s directory location! Alignments are made from the modified ProfileSearch output file with ProfileSegments. When running ProfileSegments be sure to set your list size big enough to include all of the relevant sequences remaining in the ProfileSearch output. Another handy option is -Global versus the -Local default; this will force full-length alignments, which might be what you would want, especially if you are trying to build up a multiple sequence alignment. A screen snapshot centered about the t-RNA binding region of a ProfileSegments -MSF -Global alignment made from many of the entries from the above ProfileSearch example aligned against my example EF-1 profile follows below:
Notice the difference between this alignment and examples seen with other algorithms. Profile alignments are often much more ‘gappy’ than other alignments, more so than just that caused by the divergence of this particular example. The conserved portions of the profile do not allow the corresponding portion of alignment to gap; yet gaps are easily put in the less conserved regions of the alignment. ‘Clustering’ occurs much more often with profile analyses than other methods. This is because of profile analysis’ variable gap penalties where conserved areas are not allowed to gap and variable regions are.

Hidden Markov models and profiles

As with Gribskov style profiles, HMM profiles are built from a set of pre-aligned sequences. It’s just not as important that the alignment be as comprehensive and perfect. To build a HMM profile of an alignment in SeqLab, select all of the relevant sequences, and perhaps a region within them to exclude jagged, un-alignable ends. Go to the “Functions” “HMMER” menu and pick “HmmerBuild. . .”. Specify “Selected region” rather than “Selected sequences” if restricting your profile’s length. Accept the default “create a new HMM” and specify some “Internal name for profile HMM.” Also specify the “Type of HMM to be Built” — “multiple global” is the default; “single global” may be more appropriate. This is a big difference between HmmerBuild and other profile building programs; when the profile is built you need to specify the type of eventual alignment it will be used with, rather than when the alignment is run. The HMMER profile will either be used for global or local alignment, and it will occur multiply or singly on a given sequence. Weighting is also handled differently in HMMER than it is with Gribskov profiles. To use a custom weighting scheme, e.g. if you’ve modified your RSF file weight values for ProfileBuild, you need to tell HmmerBuild not to use one of its built-in weighting schemes with the -Weighting=N option. Otherwise HmmerBuild’s internal weighing algorithm will calculate the best weights for you automatically based on the sequences’ similarities using a cluster analysis approach. It again becomes important to understand the types of biological questions that you are asking to rationally set many of the program parameters.

Notice HmmerCalibrate is checked by default. The completion of HmmerBuild automatically launches a calibration procedure that increases the speed and accuracy of subsequent analyses with the resultant profile.
HMM. The other HmmerBuild options can be explored, but read the Program Manual first. For now accept
the default HmmerBuild optional parameters and press “Run.” It’ll take a couple of minutes to build a
HMMER profile. The output is an ASCII text profile representation of a statistical model, a Hidden Markov
Model, of the consensus of a sequence family, deduced from a multiple sequence alignment. A utility
program, HmmerConvert, can change HMMER style profiles into Gribskov profiles, however information is
lost in the process. You can use your new HMMER profile as either a search probe for extremely sensitive
database searching or as a template upon which to build ever-larger multiple sequence alignments.

To use a HMMER profile as a search probe go to the “Functions” menu and pick “HMMER” “HmmerSearch.
. .” Specify the new HMMER profile by clicking “Profile HMM to use as query. . .” and using the “File
Chooser” window to select the correct HMMER profile. Change the “Sequence search set. . .” to NRL_3D
again so that the search won’t take too long. HmmerSearch has similar cutoff parameters as other GCG
database searches, that is, you can restrict the size of the output based on significance scores and you can
limit the number of pairwise alignments displayed. HmmerSearch is very slow because it is a true dynamic
programming implementation, a HMMER profile matrix against a whole database. So definitely run it in the
background when using SeqLab or, if at a terminal session, use the -Batch command line option. If your
server has multiple processors, HmmerSearch supports the multithreading -Processors=x option to speed
things up. “Run” the program when you’ve got the options set the way you want them. Do not wait for the
program to finish, go on with the rest of the tutorial, and then return to this point when it does finish. The
output is huge but very informative. Everything is based on significance Expectation value scores. The top
portion is a modified GCG list file of the most similar sequences found up to your specified Expectation cutoff
based on all domains. You should see the same entries as from ProfileSearch. The second section shows
all the pairwise alignments and finally a score distribution is plotted. Since it is a GCG list file, it can be read
by other GCG programs, in particular HmmerAlign.

HmmerAlign can be an incredible help to people working with very large multiple alignments and for adding
newly found sequences to an existing alignment regardless of size. Somewhat similar in concept to the -MSF
option of ProfileSegments, it takes a specified profile, in this case a HMMER profile, and aligns a specified set
of sequences to it, to produce a multiple sequence alignment based on that profile. Unlike ProfileSegments,
HmmerAlign takes any GCG sequence specification as input, not just the output from its own database
searching program. It is much faster to create very large multiple alignments this way, versus using PileUp,
on an entire large dataset. The rationale being — take the time to make a good small alignment and HMMER
profile, then use that to build up the original larger and larger. The alignment procedure used by HmmerAlign
is a full-blown, recursive, dynamic programming implementation, the profile’s matrix against every sequence
individually, until an entire alignment is built.

HmmerAlign can also use its profile to align one multiple alignment to another and produce a merged result of
the two. Using the original alignment that you made the profile with, against another sequence set is very
fast; it is the -MapAlignment=some.rsf(+) command line option and provides an exact, non-heuristic alignment.
A heuristic (optimality is not guaranteed) solution is provided if you use “another alignment” (the command line -Heuristic=some.msf\*{} option). Launch HmmerAlign off the “Functions” “HMMER” menu by picking “HammerAlign...” Specify the correct HMMER profile with the “profile HMM to use...” button and pick the sequences that you want to align to the profile with the “Sequences to align...” button. Press the “Options” button next and choose “Combine output alignment and...” “Original HMM alignment” and then press the “select alignment...” button. Use the next window to “Add Main List Selection...” specifying the RSF file you are currently working on. Close the “Build HmmerAlign’s Search Set” window and the “HmmerAlign Options” window and then press “Run” in the main program window.

1EFT is one of the most similar Elongation Factor 1 homologues to my example EF-1 profile that has a solved structure. Therefore, an alignment of its primary sequence with structural annotation against my sample dataset should allow a decent inference of secondary structure across the entire alignment. This is the basis of homology modeling. Here I’ve loaded the results of a HmmerAlign run on NRL_3D:1EFT, the EF-Tu structure from Thermus aquaticus, against my example EF-1 HMMER profile and its associated alignment. My inferred secondary structure is illustrated in the following “Features Coloring” graphic by highlighting the alpha helices in red:

**HmmerPfam**

As with Motifs and MotifSearch, HmmerPfam can help build up the annotation of an RSF file. This program scans sequences against a library of HMMER profiles, by default the Pfam library (A database of protein domain family alignments and HMMs 1996-2000 The Pfam Consortium). Select all of your protein sequences (do not select annotation or mask lines) and launch the program through the “Functions” “HMMER” “HmmerPfam...” menu. “Save the best scoring profile HMMs as an RSF file” and give an appropriate name. You can check out the options if desired; you may want to reduce the Expectation cutoff values. “Run” the program. When its finished (It can take quite a while to run — don’t wait for it to finish.) add it’s RSF output file to the Editor display as before with the “Output Manager”’s “Add to Editor” and
“Overwrite old with new” functions. The output .hmmerpfam file lists Pfam domain matches ranked by Expectation values and with the -RSF option writes the domain identification and Expectation value as a feature in an RSF file. The screen snapshot below shows my sample alignment over the same span as above but now including additional HmmerPfam annotation using “Graphic Features” “Display,” mode at a 4:1 zoom ratio. Inferred alpha helices are now seen as transparent red coils. I’m in the process of changing the new HmmerPfam annotation from “Solid” to “Empty” “Fill:” with the “Features Editor” so that you can see through the “Graphic Features” HmmerPfam annotation to the features behind:

Consensus issues

Consensus methods are another powerful way to visualize similarity within an alignment as well as GCG’s PlotSimilarity program. The SeqLab “Edit” menu allows you to easily create several types of consensus representations. To create a protein sequence consensus select all your sequences and use the “Edit” “Consensus . . .” menu and specify “Consensus type:” “Protein Sequence.”. When making a consensus of a protein alignment you can generate figures with black highly similar residues, gray intermediate similarities, and white non-similar amino acids. This is a nice way to prepare alignment figures for publication. The default mode is to create an identity consensus at the 2/3’rds plurality level (“Percent required for majority”) with a threshold of 5 (“Minimum score that represents a match”). This is a very strict consensus. Try lower pluralities, threshold values, and different scoring matrices to see the difference that it makes in your alignment’s appearance. Be sure that “Shade based on similarity to consensus” is checked to generate a color mask overlay on the display to help in the visualization process. The following screen illustrates a region near the carboxy termini of my example using the BLOSUM30 matrix, a “Percent required for majority” (plurality) of 33%, and a “Minimum score that represents a match” (threshold) cutoff value of 4:
A print option is available in SeqLab’s main window after you’ve found a plurality, threshold, and scoring matrix combination that you like. Go to the “File” “Print...” command and change the “Output Format:” to “PostScript” in order to prepare a PostScript file of your SeqLab display alignment. The PostScript file will capture the color scheme being displayed by the Editor at the time. Play around with the parameters — notice that as you change the font size the number of pages to be printed varies. In the “Print Alignment” menu specify “Destination... File” and give it an appropriate filename and then click “OK.” This will produce a PostScript language graphics file in the directory that you launched SeqLab from and is a great way to prepare presentations of your research. This PostScript file can be sent to a color PostScript printer, or a black and white laser printer that will simulate the colors with gray tones. Unfortunately, if it’s longer than one page, the ‘raw’ PostScript format is so different from standard single page Encapsulated PostScript format that you may have to use a different UNIX print queue. Discuss these matters with your system administrator. It may require some variation of the following type of command:

```
> lpr -PPostScript_queue seqlab_alignment.ps
```

Another option is to transfer the file to another computer where it can be imported into a PostScript savvy graphics program for further manipulation, and then it can be printed from there.

Exit SeqLab with the “File” menu “Exit” choice and save your RSF file and any changes in your list with appropriate responses. Accept the suggested changes and designate names that make sense to you; SeqLab will close. Log out of your current UNIX session on Mendel and exit the X software on the workstation that you are sitting at.

**Homework assignment**

Did you have a difficult time deciding what should or should not be aligned in your FastX file? Describe the problems that you had. What PROSITE motifs were discovered in your dataset? Do you think that they were all relevant with no false positives? If there were any false positives, what were they? What was the most
similar sequence in NRL_3D to your project molecule’s profile as reported by ProfileSearch and by HmmerSearch? How about from the BLAST run of NRL_3D two weeks ago? What were the Z score and Expectation scores, respectively, of those matches? Can you discuss the differences? What did HammerPfam find in your sequences?

**Conclusion**

The comparative method is a cornerstone of the biological sciences. Multiple sequence alignment is the comparative method on a molecular scale, and is a vital prerequisite to some of the most powerful biocomputing techniques available. Understanding something about the algorithms and the program parameters of each is the only way to rationally know what is appropriate. Knowing and staying well within the limitations of any particular method will avert much frustration. Oftentimes you’ll need to deal with very large datasets, or you may need to manually adjust alignments. A comprehensive multiple sequence editor such as the GCG Wisconsin Package SeqLab graphical user interface can be a lifesaver in these situations.

Furthermore, keep in mind that we’ve used very similar, quite easily aligned datasets for these tutorials. This was done so that we could proceed in ‘real time’ and finish the tutorial during our assigned lab period. However, most datasets that you will encounter, especially the ‘very-interesting!’ ones, will not have a bunch of obvious homologues, or you’ll be trying to align distantly related domains, or you’ll be working on a paralogous system, the list goes on. These are the situations that will present vexing alignment problems and difficult editing decisions. These are the times that you’ll really have to think.

**References**


