

# **BSC4933/5936: Introduction to Bioinformatics**

**Laboratory Section: Tuesdays from 3:45 to 5:45 PM.**

## **Unknown DNA – Rational Primer Design and Analysis – the “Guessmer.”**

**Week 3, Tuesday, September 9, 2003**

**Author and Instructor: Steven M. Thompson**

‘Not your ordinary primer design.’ The “guessmer:” How to design and analyze oligonucleotide primers for discovering genes in organisms where they have not yet been identified, when the gene’s encoded protein sequence is known in related organisms.

Techniques used in today’s computer laboratory tutorial include basic multiple sequence alignment, consensus creation, back translation, and primer discovery and evaluation.

Steve Thompson  
BioInfo 4U  
2538 Winnwood Circle  
Valdosta, GA, USA 31601-7953  
stevet@bio.fsu.edu  
229-249-9751

\*GCG<sup>®</sup> is the Genetics Computer Group, part of Accelrys Inc., a subsidiary of Pharmacia Inc.,  
producer of the Wisconsin Package<sup>®</sup> for sequence analysis.  
□ 2003 BioInfo 4U

## Introduction

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.

## The Polymerase Chain Reaction and Primers

The Polymerase Chain Reaction, PCR, developed at Cetus Corporation by Kary Mullis in the mid '80's (Saiki, et al., 1988), for which he won the Nobel Prize, and patented by Hoffman La Roche and Perkins-Elmer Corporation, has revolutionized modern molecular biology. From Jurassic Park scenarios in popular novels, to everyday research in countless laboratories across the world, to cutting-edge forensic pathology techniques, PCR is being used to analyze tinier concentrations of DNA than ever before imagined possible. PCR allows the investigator to analyze any stretch of DNA in any organism where at least some sequence information is known, either in that organism or in related organisms. It can isolate, and amplify up to around a million-fold, just a few molecules of DNA from complex environmental mixtures, even where the DNA is significantly degraded — the ramifications are incredibly far-reaching. It has been employed to analyze DNA in Egyptian mummies, preserved prehistoric insects in amber, ancient fossilized leaves from Clarkia, Idaho, and both ice-age frozen and tar-pit preserved mastodons and other animals from the 'great age of mammals.' Claims have even been made of dinosaur DNA recovery from specimens recovered in a Utah coal mine, though the results were later proven to be contamination. The practical applications are extensive in medicine, especially in the field of prenatal genetics and, in particular with HIV, immediately postnatal diagnosis. Other pathologies such as Lyme disease are also extremely amenable to PCR diagnosis. Furthermore, molecular evolutionists now have a tremendous tool for inferring phylogenies of any organism, whether they can be cultured or not. Furthermore, forensics has been completely turned about. Now investigators can isolate the DNA from incredibly obscure bits of physical evidence to positively exclude suspects based on distinct patterns, fingerprints, within their DNA. Using it to 'prove' guilt is more difficult because of the population genetics statistics involved, however, even these probabilities can be demonstrated within magnitudes of order. PCR has truly changed the face of molecular biology.

PCR is a modified primer extension reaction using a thermostable DNA polymerase that allows for the heat dissociation of newly formed complementary DNA and subsequent hybridization of oligonucleotide probes to the target regions for subsequent rounds of amplification. The scope and methods of PCR are huge and many varied and beyond the aim of this tutorial, and I will not attempt to teach the procedure. Refer to any good text in molecular biology techniques for details (for good, early, primary reviews of PCR methodology see Mullis [1990], White et al. [1989], and Chorfas [1990]). What I will attempt to teach is a method for inferring appropriate oligonucleotide probes, commonly known as primers, for PCR or hybridization screening analysis. These oligonucleotides are usually about 20 or more bases in length and target the beginning and ending locations of the PCR amplification process.

Coupled with PCR techniques and/or ultra sensitive hybridization screenings, oligonucleotide primers have allowed the 'fishing out' of thousands of genes from complex genomes that would have previously been extremely difficult to ever even find, yet alone sequence. Present-day economic, automated synthesis and the ready availability of nucleotides, have made primers commonplace. (This has also facilitated the development of reliable methods for the introduction of site-specific mutations into known sequences.) Because of the high specificity and adjustable stringency of oligonucleotide hybridization, the sequence knowledge of a relatively short stretch of unique DNA is sufficient to rapidly isolate and/or amplify, clone if desired, and sequence the corresponding gene. However, whatever technique one may use, primers are essential ingredients.

PCR and hybridization screening both require the design of appropriate primers. This can be a 'hit-or-miss' affair or you can use computational methods to greatly assist the efficiency of the process. Several strategies can be imagined for the design of oligonucleotide primers. If an exact nucleotide sequence is known, then a single oligonucleotide probe for hybridization or a pair of primers for PCR of a defined sequence can simply be selected, tested, and synthesized. In the absence of a defined DNA sequence, sometimes a group of similar DNA sequences can be aligned and a consensus sequence created from which primers can be designed. However, this is often not possible because DNA can be very, very difficult to align. In some cases one may even be forced to work off of either a small portion of a protein sequence from an Edman degradation reaction or, as will be illustrated in this tutorial, a consensus pattern from a group of related proteins — the luxury of using DNA directly will not always be available.

### **The Guessmer — from Proteins to Primers.**

When nucleotide data is lacking or problematic, amino acid sequences can be back translated to provide the necessary primers. In the absence of exact protein sequence data, a consensus pattern from a group of related proteins can often be used. Using amino acid sequence information requires one to back translate the sequence. This is not a trivial chore though, because of the degeneracy of the genetic code. There are 64 possible codons for the 20 amino acids. Because of this, many different back translation probe techniques have been employed. Two are, either utilizing large pools of short oligonucleotides whose sequences are highly degenerate, or using small pools, or even just one pair, of longer oligonucleotides of lesser or no

degeneracy. All organisms have preferential biases in codon usage and this information can be used to advantage in deciding which codons to synthesize out of all of the possible choices. This strategy of choosing the longest defined stretches of unambiguous peptide and back translating them to their most probable oligonucleotides, is known as designing “guessmers.”

Guessmers contain the combination of codons most likely to match the authentic gene. Guessmers work because the decrease in hybridization stability caused by mismatched bases is offset by an increase in stability from using longer sequences. In most cases, mismatches will occur in only the third position of incorrect codon choices and, therefore, at least two of the three bases will still be matched. Naturally, the biggest constraint on utilizing this type of strategy is that relatively long stretches of amino acid sequence are required. Because of this, guessmers are particularly appropriate when strong and sufficiently long consensus elements can be discovered in a protein family. They should be at least 30 nucleotides in length, in order to insure sufficient hybridization despite potential mismatches, though PCR primers are seldom designed as long as hybridization probes. It's also not worth the extra effort and expense to synthesize them longer than about 70 bases. For very good descriptions of the factors involved in guessmer design and analysis and references to primary literature see Sambrook et al. (1990) and Wood (1987).

### **Usual First Step in Many Molecular Biology Projects**

Probe genomic digests, shotgun clones, or cDNA libraries with a suitable oligonucleotide hybridization probe, or, much more often these days, PCR techniques designed toward the same end.

But, how do you design the oligo(s)? One way — defined DNA:

Based on known DNA sequences you can define and test probes/primers to any level of specificity using a multiple sequence alignment of those sequences and primer design and analysis software, such as Oligo or GCG's Prime. However, often the DNA is not sufficiently similar to align, therefore...

Another way — the guessmer:

start from known protein sequences and find strong consensus elements within them;  
backtranslate the consensus elements to yield consensus DNA sequences;  
use primer discovery software to locate candidate primers within the conserved DNA regions;  
test candidate primers' suitability with primer analysis software and searching tools.

Today's tutorial will explore this second route, guessmer design. In order to discover possible consensus patterns within a known protein family for the design of a guessmer, the individual members must be maximally aligned and then a consensus must be created. Alignment is usually achieved through an automated progressive, pairwise alignment procedure, here the GCG program PileUp, which inserts gaps to align the full length of its members. Other automated alignment methods are also available such as Thompson and Higgins' ClustalW (1994), Smith and Smith's PIMA (version 1.4, 1995), and Gupta et al.'s

MSA (version 2.0, 1995), as are several different manual alignment editors. Consensus sequences can then be created from the alignment. Many methods merely rely on the positional frequency of individual symbols; however, some utilize much more information. Profile analysis (Gribbskov et al., 1989) is one of these. Profile analysis takes advantage of the BLOSUM62 (Henikoff and Henikoff, 1992) PAM-style scoring matrix (Schwartz and Dayhoff, 1979) that utilizes the relative conservation of various amino acid substitutions within the alignment. Therefore, the resultant consensus residues are the most evolutionarily conserved rather than just statistically the most frequent. This can mean much more to us than an ordinary consensus and is especially appropriate in the design of the type of guessmer that we will be simulating — that is, a situation in which much sequence information for the protein of interest is known in other organisms but not in the one we are studying.

I will illustrate the design of guessmers in this tutorial using the same example as I've used in the last two tutorials — elongation factor 1 $\square$ . As before, you are to use your own 'chosen' molecule off of the project molecular systems list that I have provided earlier, and repeat below.

### **Your Project Molecular System Choices**

I realize that I'm repeating myself, but this is vital — you need to decide on a particular molecule with which to perform this and the remaining seven directed computer exercise tutorials. So that I can provide the necessary data, and to provide a diverse, yet level, playing field, this choice must be made off a list that I provide of four different 'hot' interest molecules. My list will contain molecular systems for which at least one protein structure and the protein's genomic DNA sequence is known. They will all be from organisms possessing exons to make the gene finding exercise fair regardless of choice. My apologies are offered to prokaryote biologists — sorry. You will all gain experience in all aspects of biocomputing covered in the course in a project-oriented fashion using the same natural progression as would be used in an actual experimental setting.

An advantage of this approach, besides its attempt to appeal to a wide cross-section of individuals working in diverse areas, is that the resultant predictive data derived from sequence analysis will no doubt conflict with aspects of the known structural data, but elements of truth will also be found. In this way the strengths and weaknesses of each approach is better understood, and a greater empathy is found for the tremendous problems encountered in the all-too-common case of a newly discovered gene product with no structural homologues. With this approach to computational molecular biology, you will “come full swing,” gaining an appreciation for the full biocomputing spectrum available.

The directed exercise tutorial sequence lasts for the first two thirds of the semester, ten weeks. Scheduled lab sessions are devoted after that to working on and conferring about your semester final research project. Select the molecule that interests you the most, or that most closely fits the general type of work that you plan on doing in your graduate or professional career. That way you should be more interested in using it for

biocomputing practice in the tutorials. Take your pick off of the following project molecular systems list and don't change your choice for the remainder of the directed tutorials:

- 1) Higher plant ribulose (Viridiplantae) biphosphate carboxylase/oxygenase (RuBisCO), the nuclear encoded, small subunit only. This is a very important enzyme in the Calvin cycle of photosynthesis, and, some would claim, the most abundant enzyme on earth.
- 2) Vertebrate (Vertebrata) c-H-Ras, also known as P21 Harvey ras proto-oncogene transforming protein. This incredibly 'hot' molecule is critically important in many cancer ontologies.
- 3) Vertebrate (Vertebrata) basic fibroblast growth factor, also known as heparin-binding growth factor 2 or prostatotropin. This is another popularly investigated cytokine relevant to cancer research.
- 4) Fungal (Fungi) Cu/Zn superoxide dismutase (gene name *sod*). This is a cytoplasmic, oxireducate type, free radical scavenger. Aren't free radicals implicated in both cell aging and cancer?

### **The Scenario**

You are assigned a particular protein to investigate. It is unknown in the particular organism that you're expected to work with; however, you are certain that the same protein has been worked with in other related organisms. There are many ways to approach this design problem; I will present one useful when the protein's sequence is known in several related cases. The first step is to look for it in the protein databases. You used GCG's database browser program LookUp to do this last week.

### **Week Three Tutorial: Rational Design of Guessmer Primers**

Activate and/or log on to the computing workstation you are sitting at. Remember that specialized "X server" graphics communications software is required to use GCG's SeqLab interface. In review, X server emulation software needs to be installed separately on personal style Microsoft Windows/Intel or Macintosh machines but genuine X-Windowing comes standard with most UNIX/Linux operating systems. 'Wintel' machines are often set up with either XWin32 or eXceed to provide this function; Pre OS X Macintoshes are often loaded with either MacX or eXodus software; OS X Macs can have true X windowing installed. I'll also remind you of a few X user tips: X-windows are only active when the mouse cursor is in that window, and always close windows when you are through with them to conserve system memory. Furthermore, rather than holding mouse buttons down, to activate items, just click on them. Also buttons are turned on when they are pushed in and shaded. Finally, do not close windows with the X server software's close icon in the upper right- or left-hand window corner, rather, always use GCG's "Close" or "Cancel" or "OK" button, usually at the bottom of the X window.

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!)
```

Regardless of how you’ve established the X-tunneled ssh connection, after you’ve logged onto Mendel launch SeqLab with the following command (but remember with SSH/XWin32 you need to launch “xclock &” first):

```
> seqlab &
```

Seqlab should open in List Mode displaying your default list file — the top item in the list should be the RSF file created off of the LookUp output list, or it will be the LookUp output itself, from last week. If you never got that far last week, then go back to the Week three tutorial and do the LookUp search for your chosen molecular system in the Swiss-Prot database. Use the “Output Manager” to give the file a more appropriate name that makes sense to you, but save the “.list” extension. Put the output list file in your SeqLab “Main List” and “Save” your list. You need either that LookUp output file or the RSF file that you created from it to begin this week’s tutorial. Select your RSF file if you have one, otherwise select your LookUp list file in SeqLab’s “Main List” window and switch to “Editor” mode.

### Check Out Your LookUp List from the Swiss-Prot Protein Database

You may have already done this last week. Regardless, you need to be very careful that all of the proteins included in your LookUp list are appropriate; in other words, be sure that they all actually belong to the desired molecular system and that they all come from the organisms that you need, Viridiplantae, Vertebrata, or Fungi, as the case may be. To do this very quickly <double-click> each entry’s name (or select the name and press the “INFO” button) and read over the annotation. Be especially careful of proteins that have the words “associated” or “receptor” in their name, as these are not your desired protein, rather they work together with your desired protein. “Close” the “Sequence Information” window after reading it. If you discover an inappropriate protein, you need to remove it from the SeqLab Editor display by selecting the sequence’s name and pressing the “CUT” button. If none of the proteins are what they should be, then you’ll need to repeat your LookUp search with better query terms. After verifying that all of the sequences are correct, go to the “File” menu and select “Save As. . .” Give your sequence dataset a name that makes sense to you, but leave the “.rsf” extension. Also be sure not to change anything before the last diagonal slash (the “path” to your account).

### PileUp the Sequences and Evaluate the Results

Next we need to align all of these proteins to determine the most conserved areas suitable for locating primers. Therefore, select all of your sequence entries in the editor window either by dragging your mouse through all of their names (if they all fit in the window at once), or by using <shift> click on the top and bottom-most entries, or by selecting “Select All” from the “Edit” menu. Now go to the “Functions” “Multiple Comparison” menu and choose “PileUp.”

You may want to see all the options that are available in PileUp, although we shouldn't need to use any this time since your dataset should be fairly similar, all coming from the same major grouping of life, Vertebrata, Viridiplantae, or Fungi, as your choice molecular system dictates. To see the options click on the **"Options"** button and scroll through the window; **"Close"** it when finished. Depending on the level of divergence in a data set, better multiple sequence alignments can oftentimes be generated by using alternate scoring matrices (the `-matrix=` 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, `"b1osum62.cmp,"` is used as the default symbol comparison table. Furthermore, appropriate suggested gap creation and extension penalties are now coded directly into the matrix rather than into the program. This is a greatly improved situation over the normalized Dayhoff PAM 250 table and program encoded penalty values that GCG formerly used. The BLOSUM table is more robust at handling a wider range of sequence divergence than the PAM table. Gap penalties can still be adjusted as desired but the defaults usually work quite well. Furthermore, GCG's `-InSitu` option can be incredibly effective at realigning regions within an alignment, as we'll see later in the semester. However, your sequences should all be similar enough that you can just run PileUp using the GCG defaults, therefore, just press **"Run"** in the "PileUp" window and the program will launch.

PileUp 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. The program 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 BLOSUM62 matrix and gap introduction and extension penalties used by default. Scroll through your alignment to check it out and then **"Close"** the window afterwards. My abridged example elongation factor dataset MSF file follows below:

```
!!AA_MULTIPLE_ALIGNMENT 1.0
PileUp of: @/users/thompson/.seqlab-mendel/pileup_4.list

Symbol comparison table: GenRunData:b1osum62.cmp  CompCheck: 1102

                GapWeight: 8
            GapLengthWeight: 2

pileup_4.msf  MSF: 527  Type: P  January 17, 2003 11:07  Check: 2091 ..

Name: efla_crypv      Len: 527  Check: 5146  Weight: 1.00
Name: efla_plafk     Len: 527  Check: 6400  Weight: 1.00
Name: efla_enth1     Len: 527  Check: 7380  Weight: 1.00
Name: efla_tetpy     Len: 527  Check: 5561  Weight: 1.00
Name: efla_euggr     Len: 527  Check: 8425  Weight: 1.00
Name: efla_trybb     Len: 527  Check: 3654  Weight: 1.00
Name: efla_style     Len: 527  Check: 7315  Weight: 1.00
Name: eflc_porpu     Len: 527  Check: 2148  Weight: 1.00
Name: efla_dicdi     Len: 527  Check: 5915  Weight: 1.00
Name: efl1_eupcr     Len: 527  Check: 7925  Weight: 1.00
Name: efla_blaho     Len: 527  Check: 5879  Weight: 1.00
Name: efla_eimbo     Len: 527  Check: 4498  Weight: 1.00
Name: efla_giala     Len: 527  Check: 9743  Weight: 1.00
Name: efl2_eupcr     Len: 527  Check: 5510  Weight: 1.00
```

Name: efls\_porpu Len: 527 Check: 6592 Weight: 1.00

//

	1				50
efla_crypv	~~MGK.EKTH	INLVVIGHVD	SGKSTTTGHL	IYKLGIDKR	TIEKFEKES
efla_plafk	~~MGK.EKTH	INLVVIGHVD	SGKSTTTGHI	IYKLGIDRR	TIEKFEKESA
efla_enth	~~MPK.EKTH	INIVVIGHVD	SGKSTTTGHL	IYKCGIDQR	TIEKFEKESA
efla_tetpy	~~MARGDKVH	INLVVIGHVD	SGKSTTTGHL	IYKCGIDKR	VIEKFEKESA
efla_euggr	~~MGK.EKVH	ISLVVIGHVD	SGKSTTTGHL	IYKCGIDKR	TIEKFEKEAS
efla_trybb	~~MGK.EKVH	MNLVVVGHVD	AGKSTATGHL	IYKCGIDKR	TIEKFEKEAA
efla_style	~~MPK.EKNH	LNLVVIGHVD	SGKSTSTGHL	IYKCGIDKR	TIEKFEKEAA
eflc_porpu	~~MGK.EKQH	VSIVVIGHVD	SGKSTTTGHL	IYKCGIDKR	AIEKFEKEAA
efla_dicdi	MEFPESEKTH	INIVVIGHVD	AGKSTTTGHL	IYKCGIDKR	VIEKYEKEAS
ef11_eupcr	~~MGK.EKEH	LNLVVIGHVD	SGKSTTTGHL	IYKLGIDAR	TIEKFEKESA
efla_blaho	~~MGK.EKPH	INLVVIGHV	AGKSTTTGHL	IYACGIDKR	TIERFEEGGQ
efla_eimbo	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
efla_giala	~~~~~	~~~~~	~~~STLTGHL	IYKCGIDQR	TIDEYEKRAT
ef12_eupcr	~~MERKEKDH	LNLVVIGHVD	SGKSTTTGHL	IYKLGIDER	TLAKLEEKAL
efls_porpu	~~MGK.EKTH	INLVVIGHVD	AGKSTTTGHL	IYKLGIDAR	TIAKFEADAK

	51				100
efla_crypv	EMGKGSFKYA	WVLDKKAER	ERGITIDIAL	WQFETPKYHY	TVIDAPGHRD
efla_plafk	EMGKGSFKYA	WVLDKKAER	ERGITIDIAL	WKFETPRYFF	TVIDAPGHKD
efla_enth	EMGKGSFKYA	WVLDNLKAER	ERGITIDISL	WKFETSKYFF	TIIDAPGHRD
efla_tetpy	EQGKGSFKYA	WVLDKKAER	ERGITIDISL	WKFETAKYHF	TIIDAPGHRD
efla_euggr	EMGKGSFKYA	WVLDKKAER	ERCITIDIAL	WKFETAKSVF	TIIDAPGHRD
efla_trybb	DIGKASFKYA	WVLDKKAER	ERGITIDIAL	WKFESPKSVF	TIIDAPGHRD
efla_style	EMGKGSFKYA	WVLDKKAER	ERGITIDIAL	WNFETAKSVF	TIIDAPGHRD
eflc_porpu	EMGKGSFKYA	WVLDKKAER	ERGITIDIAL	WKFETDKYNF	TIIDAPGHRD
efla_dicdi	EMGKQSFKYA	WVMDKKAER	ERGITIDIAL	WKFETSKYFF	TIIDAPGHRD
ef11_eupcr	EMGKASFKYA	WVLDKKAER	ERGITIDIAL	WKFETENRHY	TIIDAPGHRD
efla_blaho	RIGKGSFKYA	WVLAKMKAER	ERGITIDISL	WKFETRKDFD	TIIDAPGHRD
efla_eimbo	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
efla_giala	EMGKGSFKYA	WVLDQLKDER	ERGITINIAL	WKFETKKYIV	TIIDAPGHRD
ef12_eupcr	ELNKASFKYA	FVLDNLKAEQ	ERGITINCAL	RQFDTPRSY	TIIDAPGHKD
efls_porpu	EMGKSSFKYA	WVLDKKAER	ERGITIDIAL	WKFSTAKFEY	TVIDAPGHRD

////////////////////////////////////

	201				250
efla_crypv	.....	.....K	IPFVAISGFV	GDNMVERSDK	...MPWYKG
efla_plafk	.....	.....K	VDFIPISGFE	GDNLIEKSDK	...TPWYKG
efla_enth	.....	.....K	IPFVPIISGFQ	GDNMIEPSTN	...MPWYKG
efla_tetpy	.....	.....T	IPFIPISGFN	GDNMLERSTN	...APWYKG
efla_euggr	.....	.....K	VPFIPISGWN	GDNMIEASEN	...MGWYKG
efla_trybb	.....	.....K	VRFPVPIISGWF	GDNMIEKSEK	...MPWYKG
efla_style	.....	.....Q	DPFIPISGWH	GDNMLEKSPN	...MPWFTG
eflc_porpu	.....	.....K	VPKVPTSGWT	GENLFERTGG	DHALGKWKYG
efla_dicdi	.....	.....K	VAFVPIISGWN	GDNMLERSDK	...MEWYKG
ef11_eupcr	.....	.....K	MNFVPIISGFQ	GDNIQENSTN	...MPWYKG
efla_blaho	.....	.....R	IPFIPISGFN	GDNMIEHSAN	...MPWYKG
efla_eimbo	.....	.....K	VFPVPIISGFM	GDNMVERSSN	...MPWYKG
efla_giala	.....	.....E	FDYIPTSGWT	GDNIMEKSDK	...MPWYEG
ef12_eupcr	.....	.....N	VKYIPISGFD	GDNMLEQSEN	...LPWYKG
efls_porpu	GDKKEKKDKK	DKGEKKYVCS	ATFVPIISGWT	GDNMLEKSTN	...MPWYTG

	251				300
efla_crypv	KTLVEALDTM	EPPKRPTDKP	LRLPLQDVYK	IGGVGTVPVG	RVETGIIRPG
efla_plafk	RTLIEALDTM	QPPKRPYDKP	LRIPLQGVYK	IGGIGTVPVG	RVETGILKAG
efla_enth	PTLIGALDSV	TPPERPVDKP	LRLPLQDVYK	ISGIGTVPVG	RVETGILKPG
efla_tetpy	PILVEALDAL	EPPKRPSDKP	LRLPLQDVYK	IGGIGTVPVG	RVETGVIKPG
efla_euggr	LTLIGALDNL	EPPKRPSDKP	LRLPLQDVYK	IGGIGTVPVG	RVETGVLKPG
efla_trybb	PTLLEALDML	EPPVRPSDKP	LRLPLQCTCK	IGGIGTVPVG	RVETGVMKPG
efla_style	STLIDALDAL	DQPKRPDKP	LRLPLQDVYK	IGGIGTVPVG	RVETGLLKPG
eflc_porpu	PCLLEALDAC	DPPKRPSDKP	LRLPLQDVYK	IGGIGTVPVG	RVETGVIKPG
efla_dicdi	PTLLEALDAI	VEPKRPHDKP	LRIPLQDVYK	IGGIGTVPVG	RVETGIIKPG
ef11_eupcr	PTLCAALDSF	KIPKRPIAKP	LRLPLQDVYK	IGGIGTVPVG	RVETGVLKAG

```

ef1a_blaho PTLLEALDNV HPPKRPVDKP LRLPLQDVYK IGGIGTVPVG RVETGVLPKG
ef1a_eimbo KILVEALDNV EPPKRPSDKP LRLPLQDVYK IGGIGTVPVG RVETGILKPG
ef1a_giala PCLIDAIDGL KAPKRPTDKP LRLPIQDVYK ISGVGTVPAG RVETGELAPG
ef12_eupcr PTLTEALDEF KVPKRPIKKP LRVPIQDVYK IAGIGTVPVG RVETGVLPKR
ef1s_porpu PTLFEVLDAM KPPKRPTEDP LRLPLQDVYK IGGIGTVPVG RVETGILKAG

```

////////////////////////////////////

```

                                401                                450
ef1a_crypv ITAKMDKRSK KVLE...ENP KL.....IK SGDAALVVMQ
ef1a_plafk IDSKIDKRSK KVVE...ENP KA.....IK SGDSALVSLE
ef1a_enth LLSKIDRRRTG KSM..EGGEP EY.....IK NGDSALVKIV
ef1a_tetpy IHDKIDRRRTG KSQ..E.ENP KF.....IK NGDAALVTLI
ef1a_euggr IQTKIDRRRSK KEL..E.AEP KF.....IK SGDAAIIVLMK
ef1a_trybb IESKIDRRRSK KEL..E.KAP KS.....IK SGDAAIIVRMV
ef1a_style IESKVDRRSK KVL..E.EEP KF.....IK SGDAALVVMV
ef1c_porpu LLLKMDRRRSK KKL..E.DSP KM.....IK SGDAAMVKMV
ef1a_dicdi IVDKVDRRTG AVVAKEGTAA VV.....LK NGDAAMVELT
ef11_eupcr LLTKADKRSK KMTE...ENP KF.....LK AGDAGLIRLS
ef1a_blaho IMSEMDKRTG KVLRL...ENP DI.....VK NGKSMMAQLV
ef1a_eimbo LEKRLDRRSK KALE...DDP KF.....IK TGDAAIKME
ef1a_giala FLQKLDKRT. .LKPENP PD.....AG RGDCIIVKMV
ef12_eupcr LLSKNEARTG KLIE...EAP KF.....LK NGESGIVELV
ef1s_porpu ILSKDKR.G KQTHDVSDDT EWATKDDAEP RNNRMNIAAK TGESVNVWLQ

```

```

                                451                                500
ef1a_crypv PLKPLCVEAF TDYPPLGRFA VRDMKQTVAV GVIKSVTKKE ..ATSKKK~~
ef1a_plafk PKKPMVVETF TEYPPLGRFA IRDMRQTIIV GIINQLKRKN LGAVTAKAPA
ef1a_enth PTKPLCVEEF AKFPPLGRFA VRDMKQTVAV GVVKAVTP~~ ~~~~~
ef1a_tetpy PTKALCVEVF QEYPPLGRYA VRDMKQTVAV GVIKKVEKKE K~~~~
ef1a_euggr POKPMCIVESF TDYPPLG.VS CGDMRQTVAV GVIKSVNKKE NT.GKVTKAA
ef1a_trybb POKPMCIVEVF NDYAPLGRFA VRDMRQTVAV GIIKAVTKKD GSGGKVTKAA
ef1a_style POKPMCVEAF NQYPPLGRFA VRDMKQTVAV GVIKEVVKKE .QKGMVTKAA
ef1c_porpu ASKPMCVEAF TSYPPPLGRFA VRDMRQTVAV GVIKSVEKKE .VEGKMTKSA
ef1a_dicdi PSRPMCIVESF TEYPPLGRFA VRDMRQTVAV GVIKSTVKA PGKAGDKKGA
ef11_eupcr PSKPLCVETF ATYAPLGRFA VRDMRQTVAV GVIQEIKKKA TEDKKGKKK~
ef1a_blaho PSKPMCIVETF SDYPPLGRFA VRDMRQTVAV GIIKSTVRAK ~~~~~
ef1a_eimbo PSKPMCIVESF IEYPPLGRFA VRDMKQTVAV GVIKGVEKKE .AGGKVTKSA
ef1a_giala POKPLCCETF NDYAPLGPFA VR~~~~~ ~~~~~ ~~~~~
ef12_eupcr PTKPLCVEEF SKYAALGRFV IRDMKRTVAV GVIQEVIIHK ETKKKASKR~
ef1s_porpu PTKAMVVEAY SMYSPLGRFA VRDMKKTAV GVIQCVQPRN MAKGATEELP

```

```

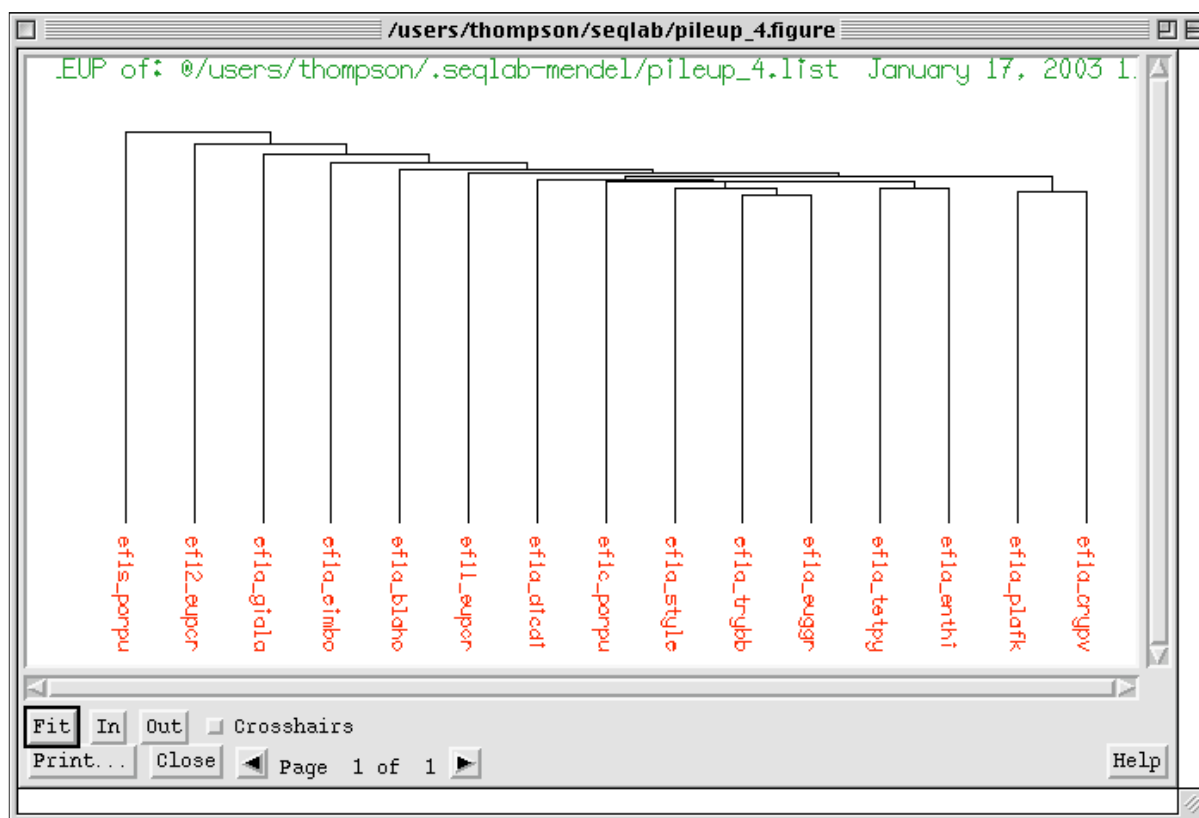
                                501                                527
ef1a_crypv ~~~~~
ef1a_plafk KK~~~~~
ef1a_enth ~~~~~
ef1a_tetpy ~~~~~
ef1a_euggr QKKK~~~~~
ef1a_trybb VKASKK~~~~~
ef1a_style QKKK~~~~~
ef1c_porpu AKK~~~~~
ef1a_dicdi AAPSKKK~~~
ef11_eupcr ~~~~~
ef1a_blaho ~~~~~
ef1a_eimbo QKATGKK~~~
ef1a_giala ~~~~~
ef12_eupcr ~~~~~
ef1s_porpu IRGESDAVSK YIKFRPLPLK AGKKAKK

```

If you get an error message from PileUp (check the “**Windows**” menu “**Job Manager**” for error reports) or an empty output file, or if one or more of the sequences doesn’t seem to fit into the alignment with the rest of the sequences, then most likely you had sequences in your dataset that should not all be aligned together. In other words, you didn’t evaluate your LookUp list carefully enough. Go back to your Editor display and get rid

of the troublemaker sequences and then re-perform the PileUp run. All of your sequences should nicely align. If they don't, you've got some sort of major problem with your dataset.

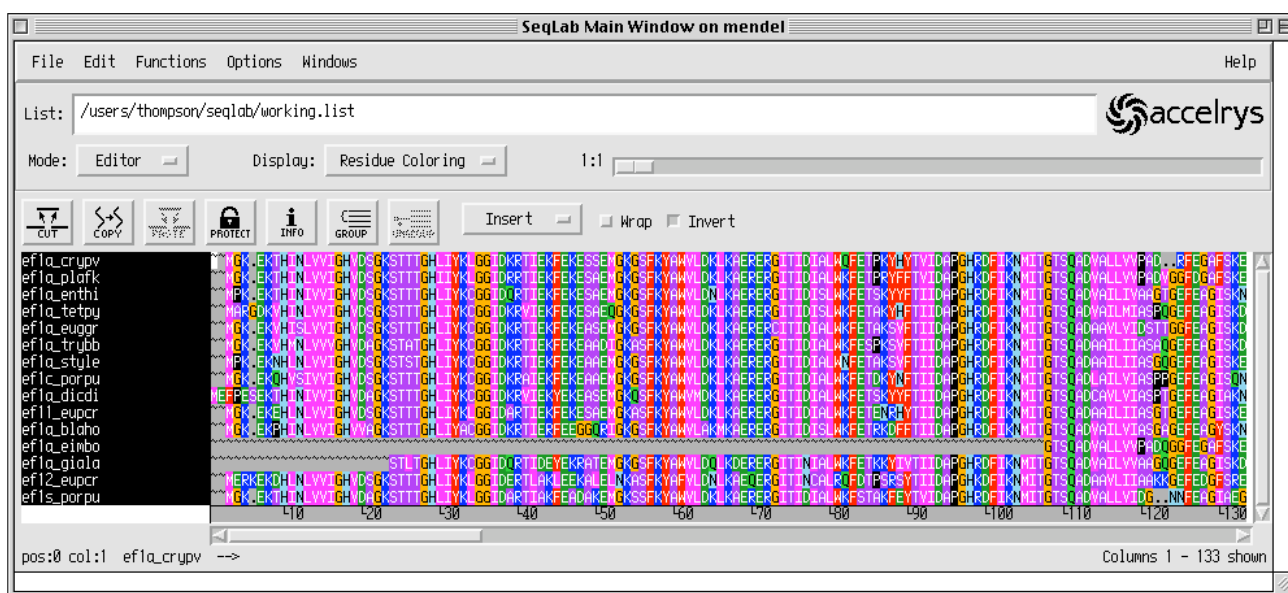
After scrolling through your alignment and then “Close”ing its window, the next window visible will be the “SeqLab Output Manager.” As described last week, this is a very important window and will contain all of the output from your current SeqLab session. Files may be displayed, printed, saved in other locations with other names, and deleted from this window. We need to use an extremely important function at this point. Press the “Add to Editor” button and specify “Overwrite old with new” in the next window when prompted, to take your MSF output and merge it with the RSF (Rich Sequence Format: the alignment as well as all reference annotation) file in the open Editor. This will keep all feature information intact, yet renumber all of its locations based on the alignment. “Close” the “Output Manager” after loading your new alignment into the Editor. The next window will contain PileUp's cluster dendrogram graphic; my example follows:



This similarity dendrogram can be helpful in adjusting the Weight category of sequences in an alignment for the formation of a profile; however, we will not bother with that today. The length of the vertical lines is proportional to the difference between the sequences. Realize, though, that this tree is not an evolutionary tree. No phylogenetic inference algorithms, such as maximum likelihood or parsimony, or correction models, such as Jukes-Cantor or Kimura, are used in its construction. We will learn about these techniques in the next few weeks. PileUp's dendrogram merely indicates the relative similarity of the sequences and, therefore, the clustering order in which the alignment was built.

If desired, you can directly print from SeqLab graphics Figure windows to PostScript files by picking “Print . . .” “[Encapsulated] PostScript File” “Output Device:” Name the output file anything that you want; click “Proceed” to create an EPSF output in your current directory. To actually print this file you may need to ftp it to a local machine attached to a PostScript savvy printer unless you have direct access to the UNIX system printer and it is PostScript compatible. (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.

After loading your new alignment, the SeqLab Editor display should look similar to the following graphic using “Residue Coloring” and a “1:1” zoom ratio, but obviously with your chosen project data, not my example. Notice that your residues now align by color:



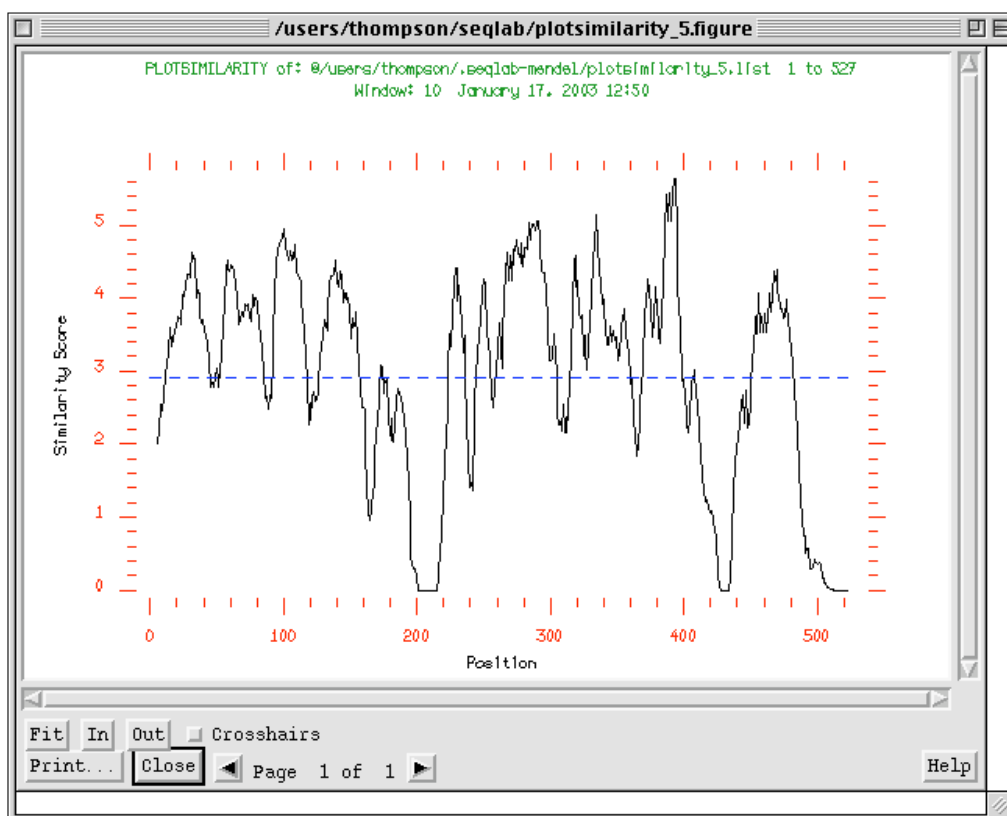
You may want to go to the “File” menu now, and periodically throughout the tutorial, to save your work using the “Save as . . .” function in case of a computer or network problem. Use the same filename as before to overwrite the previous version. There’s no need to save multiple versions.

### Determine Areas of Maximal Conservation

Next, we need to decide what portions of the alignment to find primers in by determining which areas of it are the most highly conserved. To design a hybridization probe, one, most highly conserved section is chosen; to design paired PCR primers, two flanking, highly conserved areas are chosen. We can easily visualize the positional conservation of a multiple sequence alignment with the graphics program PlotSimilarity.

Make sure that all of your entries are still selected and then run **PlotSimilarity** on your protein sequence alignment by going to the SeqLab “**Functions**” menu; select “**Multiple Comparison**” and then “**PlotSimilarity**.” You may get a “Which selection” box if you have previously selected a region of the alignment; if you do, specify “**Selected sequences**” not “Selected region.” This will produce a PlotSimilarity

dialog box. We need to change some of the program defaults there, so choose “**Options . . .**” Check “**Save SeqLab colormask 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 and the second specification launches the program’s **-Expand** option in the command line box. 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 the alignment happens to occupy. “**Close**” the “**PlotSimilarity Options**” window; notice that the “**Command Line:**” text box in the program window now reflects your updated options. Click the “**Run**” button to launch the program. The output will quickly return. “**Close**” the “**plotsimilarity.cmask**” window and the “**Output Manager**” and then take a look at the similarity plot. My elongation factor example follows below:

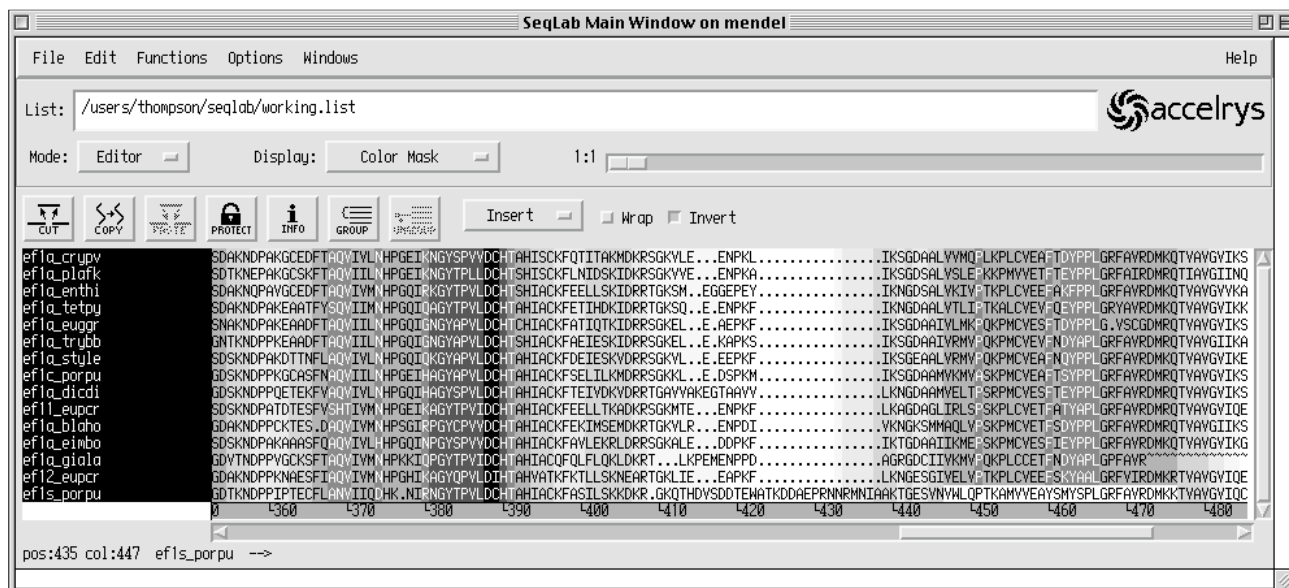


An advantage of running PlotSimilarity on a protein alignment rather than on a DNA alignment is that the peaks on the plot not only represent the most conserved regions of the alignment, but also those areas most resistant to evolutionary change due to the algorithm’s use of the BLOSUM matrix in its calculations.

My example shows a great deal of sequence similarity. Especially strong peaks can be seen centered around positions 100, 290, 335, and 395, though many other areas are nearly as conserved. The ordinate (Y-axis) scale is dependent on the scoring matrix used by the program, by default the BLOSUM62 table in which amino acid identities vary from 4 to 11, or you can specify an alternative. The dashed line across the middle shows the average similarity value for the entire alignment, here a little above 2.9. The point is to identify those regions of the alignment that will be most appropriate for designing primers.

Decide which areas of high conservation you want to use for designing PCR primers from your alignment, and write them down. Try to identify two highly conserved, minimally ambiguous, sequence regions that flank the longest stretch of your alignment possible. Obviously this step is a compromise depending on just how long of a chunk of the gene you are trying to amplify, and there's no way to get the entire thing with a guessmer approach. Make these individual regions as long as possible; try to get two areas that correspond to at least 100 bases each, in other words, around 30 to 40 or more amino acids each. We will isolate the best PCR primers within these stretches, eventually locating an oligonucleotide between 20 and 50 bases in length (corresponding to a peptide of 7 to 17 residues). We need to make the original choices this long to give the primer discovery program a chance to 'explore' within the stretch; otherwise, the program won't have enough of a chance to find good primers. In 'real life,' the more uncertainty you have regarding your template, the closer to 50 bases in length your primers will have to be, to maximize annealing potential. Here we'll just make ones 25 or so bases long. "Close" the PlotSimilarity window after noting the general location of your candidate conserved probe sections of the alignment.

Now go up to the SeqLab "File" menu; select "Open Color Mask Files." Select the file displayed in the dialog box, "plotsimilarity.cmask;" click "Add" and then "Close." Notice that the display is now represented in various gray-tones — the intensity of color is proportional to the level of similarity in the alignment at that point, averaged over the default window of 10 residues. Notice the correspondence between the original plot's peaks and valleys and the color mask's dark and light areas. This representation allows you to see the exact point at which similarities rise and fall in your alignment. Refine your notes from the previous paragraph to indicate the exact starting and stopping residue of your candidate probe regions. Determine these exact positions by placing your cursor in the alignment and noting the "col:" number in the lower left-hand corner of the window. Below, I show my candidate downstream conserved region stretching from column 447 to column 476. Note that I picked an area that was not as highly conserved as several somewhat upstream from there in order to amplify a longer stretch of my gene:



## Use ProfileMake to Create a Consensus

Next we need to generate a consensus of our sequence alignment. We could use the “Consensus” tool under SeqLab’s “Edit” menu; however, the most powerful protein sequence consensus method I am aware of is the Profile algorithm. This algorithm uses all of the data of an alignment, its conservation and its variability, as well as the BLOSUM matrix to create an alignment specific amino acid substitution matrix. Certainly, in this case, because of the high similarity of the sequences in my example, the difference would be trivial, but sometimes it can make a big difference. A profile, and its inherent consensus, is created with the program ProfileMake. Be sure that all of your sequences are still selected and then go to the “**Functions**” “**Multiple Comparison**” menu and launch “**ProfileMake**.” If asked, specify “**Selected sequences**.” Punch the “**Options**” button, select “**Write the consensus into a sequence file**,” and supply an appropriate filename. This will launch the program’s -SeqOut option to generate a normal GCG sequence file of the consensus in addition to the profile. Leave the other options as they are and “**Close**” the “**Options**” window. Press “**Run**” in the “**ProfileMake**” program window and check out the results. Take a look at the consensus sequence, it’ll be the top-most window. The Profile algorithm will decide on the most conserved residue for each position. Also notice that the header contains information relating to the sequence’s creation through ProfileMake; this can be valuable. My abridged example profile consensus sequence follows below:

```
!!AA_SEQUENCE 1.0
(Consensus) (Peptide) PROFILEMAKE v4.50 of: @/users/thompson/.seqlab-mendel/prof
ilemake_6.list Length: 527 Sequences: 15 MaxScore: 1615.55 January 17, 2003
14:34

      Gap: 1.00      Len: 1.00
GapRatio: 0.33 LenRatio: 0.10

      input_6.rsfc{EF1A_CRYPV} From:      1      To:      4
98      Weight: 1.00
//////////
      input_6.rsfc{EF1S_PORPU} From:      1      To:      5
27      Weight: 1.00
Symbol comparison table: GenRunData:blosum62.cmp FileCheck: 982

Relaxed treatment of non-observed characters
Exponential weighting of characters
Length: 527 January 17, 2003 14:34 Type: P Check: 5546 ..

  1  MEMGKSEKTH INLVVIGHVD SGKSTTTGHL IYKCGGIDKR TIEKFEKEAA
 51  EMGKGSFKYA WVLDKLKAER ERGITIDIAL WKFETPKYYF TIIDAPGHRD
101  FIKNMITGTS QADVAILVIA SGQGEFEAGI SKDGQTREHA LLAYTLGVKQ
151  MIVAINKMDD VKDKTVDNYS QERYEEIKKE VSDYLKKGVY NKAPEGDKKK
201  GDKKEKDKK DKGEKKYVCK VPFVPISGWN GDNMIEKSDN DHALMPWYKG
251  PTLIEALDSL EPPKRPTDKP LRLPLQDVYK IGGIGTVPVG RVETGVLKPG
301  MVVTFAPSGK VTTEVKSDEM HHEQLPEAVP GDNVGFNVKN VSVKDIKRGY
351  VAGDAKNDPP KGCESFTAQV IVMNHPGQIK NGYTPVLDCH TAHIAKFFEE
401  ILSKIDRRSG KVLEPEGENP KFATKDDAEP RNNRMNIAIK NGDAALVKMV
```

```
451 PSKPMCSETF TDYPPLGRFA VRDMKQTVAV GVIKSVTKKE NAKGKVTKAA
501 QKKKKKKVSK YIKFRPLPLK AGKKAKK
```

“Close” the consensus window after you’ve looked at it. You may want to look at your resultant “.prf” file. It’ll be just behind the “Output Manager.” It is a huge table of numbers that doesn’t make a whole lot of sense at first glance; however, it can be a tremendously powerful tool in subsequent analysis steps. Other programs can read and interpret all of those numbers to perform very sensitive database searches and alignments by utilizing the information within it that penalizes misalignments in phylogenetically conserved areas more than in variable regions. “Close” the “.prf” window afterwards.

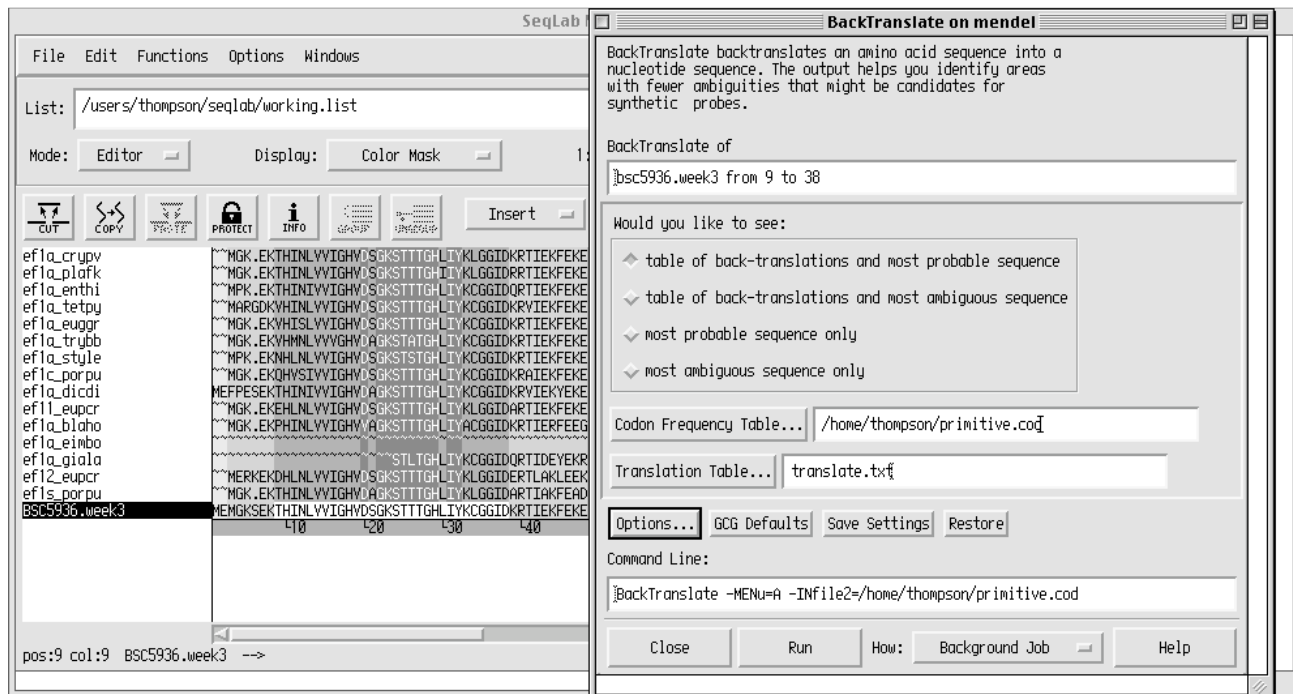
### Use BackTranslate on Your Conserved Candidate Regions of the Consensus Sequence

Begin by loading the new ProfileMake protein consensus sequence into the editor. Do this through the “Windows” menu “Output Manager” window. Select your new consensus sequence file name there and press the “Add to Editor” button. “Close” the “Output Manager” window after loading the consensus sequence. Next, use your notes from PlotSimilarity above to select one of the exact sequence regions on the consensus sequence that corresponds to your chosen region. Begin with the upstream region. I selected residues 9 through 38 for my upstream area. Your peptide candidate sequences may not be quite as long as my examples. I was fortunate to find such strong consensus elements. Regardless of what length sequences you come up with though, they are still peptide sequences and oligonucleotide primers are necessary for PCR methodology. Backtranslation is not trivial because of the degeneracy of the genetic code. GCG has addressed this problem with their program BackTranslate. Alternate codons are indicated in the output along with their order of preference, based on the codon usage table that you specify, for each amino acid of the sequence. You can choose from them; the program generates either the most probable or the most ambiguous sequence.

To use BackTranslate you must decide which codon usage table you want the program to utilize. By default BackTranslate will use a frequency table designed from highly expressed *E. coli* genes. Therefore, if you’re working with an *E. coli* gene, the program’s default is appropriate. However, if your protein comes from anything else, you will want to use an alternate table. GCG provides alternate data files in a public data library with the GCG logical name GenMoreData. The available tables, in addition to the default codon usage table, “ecohigh.cod,” are: “celegans\_high.cod,” “celegans\_low.cod,” “drosophila\_high.cod,” “human\_high.cod,” “maize\_high.cod,” and “yeast\_high.cod.” Even more tables are available at various molecular biology data servers such as IUBIO (<http://iubio.bio.indiana.edu/soft/molbio/codon/>). The TRANSTERM database at the European Bioinformatics Institute (<ftp://ftp.ebi.ac.uk/pub/databases/transterm/>) also contains several and an especially good selection derived from a recent GenBank version comes from the CUTG database (<http://www.kazusa.or.jp/codon/>) available in GCG format through various SRS servers (e.g. see <http://srs.sanger.ac.uk/srs6bin/cgi-bin/wgetz?-page+LibInfo+-lib+CUTG>). Furthermore, if you are not satisfied with any of the available options, GCG has a program, CodonFrequency, that enables you to create your own codon frequency table from known coding sequences or other codon tables. I used

CodonFrequency to create a custom table for my example dataset that combined *Giardia*, *Dictyostelium*, and *Physarum* preexisting tables.

Now go to the “Functions” “Translation” “BackTranslate” menu and specify the “Selected region” in the “Which selection” window. In the BackTranslate program window change the “Would you like to see:” default to “table of back-translations and the most probable sequence.” You also need to change the “Codon Frequency Table . . .” from the default “ecohigh.cod” to something more reasonable, so press the button and choose either the human, maize, or yeast table, whichever is the most appropriate for your project system, from the “Chooser for Codon Frequency Table” window that pops up. Press the “OK” button in the “Chooser” window after selecting the appropriate table. Here’s my run setup for BackTranslate:



Press “Run” in the program window when you’ve gotten your settings correct. The output will quickly be displayed; notice how each amino acid residue’s codon choices are listed as a frequency. My abridged upstream elongation factor backtranslation sequence data file is shown below:

```
!!NA_SEQUENCE 1.0
BACKTRANSLATE of: : input_8.rsfc{BSC5936.week3} check: 5546 from: 9 to: 38

////////////////////////////////////

      Thr           His           Ile           Asn           Leu           Val           Val
ACT 0.30   CAC 0.52   ATT 0.44   AAT 0.50   CTC 0.27   GTT 0.35   GTT 0.35
ACC 0.28   CAT 0.48   ATC 0.41   AAC 0.50   TTA 0.24   GTC 0.31   GTC 0.31
ACA 0.25           ATA 0.15           CTT 0.19   GTG 0.20   GTG 0.20
ACG 0.16           TTT 0.13   GTA 0.14   GTA 0.14
           CTG 0.12
           CTA 0.05
34           31           21           17           15           22           32

16 - 22
```

Ile	Gly	His	Val	Asp	Ser	Gly
ATT 0.44	GGT 0.40	CAC 0.52	GTT 0.35	GAT 0.56	TCA 0.24	GGT 0.40
ATC 0.41	GGC 0.26	CAT 0.48	GTC 0.31	GAC 0.44	TCT 0.23	GGC 0.26
ATA 0.15	GGA 0.20		GTG 0.20		TCC 0.18	GGA 0.20
	GGG 0.13		GTA 0.14		AGC 0.15	GGG 0.13
					AGT 0.12	
					TCG 0.08	
32	41	24	19	31	13	16
23 - 29						
Lys	Ser	Thr	Thr	Thr	Gly	His
AAG 0.57	TCA 0.24	ACT 0.30	ACT 0.30	ACT 0.30	GGT 0.40	CAC 0.52
AAA 0.43	TCT 0.23	ACC 0.28	ACC 0.28	ACC 0.28	GGC 0.26	CAT 0.48
	TCC 0.18	ACA 0.25	ACA 0.25	ACA 0.25	GGA 0.20	
	AGC 0.15	ACG 0.16	ACG 0.16	ACG 0.16	GGG 0.13	
	AGT 0.12					
	TCG 0.08					
12	6	11	19	17	25	33
30 - 36						
Leu	Ile	Tyr	Lys	Cys	Gly	Gly
CTC 0.27	ATT 0.44	TAC 0.53	AAG 0.57	TGC 0.57	GGT 0.40	GGT 0.40
TTA 0.24	ATC 0.41	TAT 0.47	AAA 0.43	TGT 0.43	GGC 0.26	GGC 0.26
CTT 0.19	ATA 0.15				GGA 0.20	GGA 0.20
TTG 0.13					GGG 0.13	GGG 0.13
CTG 0.12						
CTA 0.05						
36	76	69	52	40	39	0
37 - 38						
Ile	Asp					
ATT 0.44	GAT 0.56					
ATC 0.41	GAC 0.44					
ATA 0.15						
0	0					

bsc5936.week3.seq Length: 90 January 18, 2003 15:37 Type: N Check: 2455 ..

```

1  ACTCACATTA ATCTCGTTGT TATTGGTCAC GTTGATTCAG GTAAGTCAAC
51 TACTACTGGT CACCTCATTT ACAAGTGCGG TGGTATTGAT

```

Some positions are very clear-cut with obvious, strong preferences; however, other positions are much more ambiguous with nearly equal codon preferences. In a wet-lab setting you may want to consider synthesizing a population of mixed oligo primers for those positions. Use the “**Output Manager**” “**Save As. . .**” function to give the output sequence file a name that identifies it as the upstream candidate sequence, such as “upstream.consensus.seq,” and then use the “**Add to Editor**” function to load the new sequence into your SeqLab Editor display. This candidate nucleotide sequence is the most likely coding sequence for the upstream peptide that you specified using the codon frequency chart that you chose.

Repeat the procedure by selecting your downstream choice and running BackTranslate again on it. Also run BackTranslate on the entire length of your consensus sequence for use a bit later in the tutorial. Specify “**Selected Sequence**” when prompted by the “**Which Selection**” box rather than “Selected Region” to backtranslate the whole sequence. Be sure to specify your appropriate codon preference table and change

the type of sequence produced from most ambiguous to most probable in each of these additional runs. If it's the same SeqLab session, SeqLab will remember these options from before. Load these other two BackTranslated sequences into the SeqLab Editor display also. These sequences will not load aligned to their respective protein coding regions; in fact, this would be impossible because the protein is not spaced out to leave two gaps between every amino acid. They will load starting at position one in the Editor display. Just realize that they are no longer aligned to the dataset alignment above them.

As mentioned above, a viable alternative, often utilized, is to prepare a mixture of oligo's containing various codons for those positions that are particularly ambiguous. A few more analyses are necessary before running off to synthesize your new primers, however. We need to decide which portions of the consensus elements that we have identified are most appropriate for primers. And of those portions, we need to determine if they have significant internal complementation such that strong 'hairpin' structures would be formed, and we should also check for self- and primer-dimer complementation. The GCG program Prime can be used for all these tests. We also need to run a DNA database search to make sure that only the type of genes that we are interested in are 'found.' GCG's program FindPatterns is probably best for this type of search because it does not allow gapping.

### **GCG's Primer Discovery and Analysis Program — Prime — Finding 'Good' Primers**

Prime can locate acceptable primers within any DNA template sequence. The program is quite powerful and contains many, many options to maximize flexibility. We will use it at this point in a somewhat different manner than would be normally expected, though — that is, to find the best forward and reverse primers separately within defined upstream and downstream sequence regions that we have identified as the best candidate locations based on sequence similarity. That's the whole point of this tutorial.

Specify the upstream and then the downstream BackTranslated sequences in turn for Prime to search. I'll begin with the 5' example. Select the upstream BackTranslated candidate nucleotide sequence's name. Launch Prime by going to the "**Functions**" menu; choose "**Primer Selection**" and then "**Prime**." If you have any region of your dataset selected, you'll get a dialog box asking whether you want to analyze the selected sequences or the selected region; choose "**Selected sequences**." The Prime program box will display; select "**Options**." Scroll down through the extensive options list noting how many are available in the "**Prime Options**" window. The option that we are most concerned with on this first run through the program specifies to search for forward primers only, your 5' candidate. Therefore, minimally take advantage of "**Select: forward primers, only**," but also specify "**Save primers found to a pattern file**" (the -Forward, -NoProducts, and -FoundPrimers options). Give the pattern file a name that makes sense to you, such as "prime.forward.dat." The -FoundPrimers option saves the discovered forward primers into a special GCG data format file as well as into the standard text output. You may also want to specify a slightly longer "**Primer Length**" than the 18 through 22 bases default, though it isn't necessary. I changed the length parameter to a "**Minimum**" of "**25**" and a "**Maximum**" of "**50**" in my example to emulate reality taking into account potential mismatches introduced in the backtranslation step. "**Close**" the options window. Press

“Run” in the program window after making your selections. Prime will now search for the best forward primer within your BackTranslated upstream consensus sequence.

The output will quickly display. The pattern data file lists the primers in a format that can be used in subsequent Prime runs; look it over and then “Close” the window. Use the “Output Manager” to select and “Display” the file that ends with the “.prime” extension. This text file describes the conditions used in the run and lists acceptable primers with their corresponding melting temperatures. Scroll through it and then “Close” its window. A graphics window shows where the primers anneal to your sequence pictorially. The primers are ranked in terms of an annealing score with smaller numbers being better. Read the Prime program “Help” in a subsequent run, if you are interested in how this function is calculated. “Close” the “Output Manager” window and the graphics window to return to SeqLab’s main window.

Your first pass may not find anything. Prime can sometimes be quite frustrating to run, for its parameters can be too stringent to find anything at all. Often you will end up having to change many of the other options in subsequent program runs to make it work. GC content (the -GCMaxPrimer and -Min- parameters, “Primer % G+C”) can be particularly problematic; the default requires between 40 and 50 percent. The GCG program Composition can give you an exact count of nucleotide content if you need it. The -TMMaxPrimer, “Primer Melting Temperature (degrees Celsius)” “Maximum” sometimes causes problems too; the default is 65°. An alternative option is to turn all of the constraints off by selecting the button next to “Ignore most of the constraints set by default . . .” and working your way toward more restrictive conditions rather than the other way around. If no primers were discovered, then either repeat the Prime run with different, more permissive, options (the output tells you exactly what test failed), or choose a different and/or longer section of your dataset to perform the procedure upon. Sometimes it will take many passes through the program adjusting different parameters each time in order to finally get something acceptable. Use the same data file output name in repeated runs so that you end up with only the one successful set of forward primers for your dataset. My example was successful on its first run without having to change any of the other options. That Prime output file follows below in an abridged form; the options I set are in bold:

```
PRIME of: input_12.rsfc{BSC5936.week3.upstream} ck: 2455 from: 1 to: 90 Januar  
y 19, 2003 12:08
```

INPUT SUMMARY  
-----

```
Input sequence: /users/thompson/.seqlab-mendel/input_12.rsfc{BSC5936.week3.upstr  
eam}
```

```
*** PRIME is set to search for primers on forward strand only. ***
```

```
Primer constraints:
```

```
primer size: 25 - 50
```

```
primer 3' clamp: S
```

```
primer sequence ambiguity: NOT ALLOWED
```

```
primer GC content: 40.0 - 55.0%
```

```
primer Tm: 50.0 - 65.0 degrees Celsius
```

```
primer self-annealing. . .
```

```
3' end: < 8
```

```
(weight: 2.0)
```

```
total: < 14
```

```
(weight: 1.0)
```



```
                primer %GC: 40.0
primer Tm (degrees Celsius): 57.0
                annealing score: 18
```

-----

Primer: 4

[DNA] = 50.000 nM [salt] = 50.000 mM

```
                    5'                                3'
forward strand primer (31-mer): 15 CGTTGTTATTGGTCACGTTGATTCAGGTAAG 45
```

```
                primer %GC: 41.9
primer Tm (degrees Celsius): 58.7
                annealing score: 18
```

-----

Primer: 5

[DNA] = 50.000 nM [salt] = 50.000 mM

```
                    5'                                3'
forward strand primer (32-mer): 14 TCGTTGTTATTGGTCACGTTGATTCAGGTAAG 45
```

```
                primer %GC: 40.6
primer Tm (degrees Celsius): 59.5
                annealing score: 18
```

////////////////////////////////////

-----

Primer: 25

[DNA] = 50.000 nM [salt] = 50.000 mM

```
                    5'                                3'
forward strand primer (35-mer): 11 ATCTCGTTGTTATTGGTCACGTTGATTCAGGTAAG 45
```

```
                primer %GC: 40.0
primer Tm (degrees Celsius): 60.6
                annealing score: 22
```

-----

Repeat this procedure with your 3' downstream BackTranslated sequence. The “**Windows**” menu contains a ‘shortcut’ listing of all programs used in the current session; you can launch any of them from there as well as the “**Functions**” menu. Be sure to specify “reverse primers, only” in the “**Prime Options**” dialog box this time. Don’t forget to continue to use the “**Save primers found to a pattern file**” option, but specify a different, distinguishing name than before, e.g. “prime.reverse.dat.” Remember, you may be forced to rerun the program a number of times adjusting options and/or the region searched until you are successful. This can be frustrating — just persevere. Play with the options until you find at least one acceptable primer



Primer: 2

[DNA] = 50.000 nM [salt] = 50.000 mM

reverse strand primer (25-mer): 5' 86 TTCATATCACGAACAGCGAAACGAC 3' 62

primer %GC: 44.0  
primer Tm (degrees Celsius): 56.3

annealing score: 14

-----

Primer: 3

[DNA] = 50.000 nM [salt] = 50.000 mM

reverse strand primer (26-mer): 5' 87 CTTTCATATCACGAACAGCGAAACGAC 3' 62

primer %GC: 46.2  
primer Tm (degrees Celsius): 57.0

annealing score: 14

-----

Primer: 4

[DNA] = 50.000 nM [salt] = 50.000 mM

reverse strand primer (27-mer): 5' 88 GCTTCATATCACGAACAGCGAAACGAC 3' 62

primer %GC: 48.1  
primer Tm (degrees Celsius): 59.1

annealing score: 14

-----

Primer: 5

[DNA] = 50.000 nM [salt] = 50.000 mM

reverse strand primer (25-mer): 5' 85 TCATATCACGAACAGCGAAACGACC 3' 61

primer %GC: 48.0  
primer Tm (degrees Celsius): 57.8

annealing score: 14

////////////////////////////////////

-----

Primer: 25

[DNA] = 50.000 nM [salt] = 50.000 mM

5' 3'

```
reverse strand primer (33-mer):      70 CGAAACGACCGAGTGGTGGGTAATCAGTCAAAG 38
```

```
          primer %GC:  51.5  
primer Tm (degrees Celsius):  63.9  
          annealing score:    15
```

-----

An alternative to individually isolating forward and reverse primers separately is to search through a range delineated by the 5' and 3' most conserved areas on the overall BackTranslated consensus. You can specify primer target regions within this delineation with the -Begin2=, -End2=, and -Include= options to attempt to force primer discovery within the conservation peaks earlier identified. This way you don't need to specify forward only or reverse only primers separately and the pair are automatically tested against one another. You may want to try this procedure on your dataset as well. However, finding the longest product possible with its primers lying in the most conserved regions of the alignment is very tricky using this method; you'll need to 'play with' the product length and the include parameters. It can be just as frustrating as the method that I've shown you, perhaps even more so, since now you must be concerned with all of the product as well as all of the primer parameters.

### **Test Primers Against Each Other Along with Their Template**

If you've designed primers with the forward only and reverse only method that I've described here, you'll want to test the pairs together with each other to be sure that they will not anneal with each other so badly as to interfere with the reaction. The Prime program allows you to do this, and to test any other primers desired, by specifying an input data list of primers at run-time. That way, rather than discovering the best primers within a specified template, the program tests and ranks all the primers fed to it against the template. Another GCG program, PrimePair can test a list of input primers against themselves in the absence of a template. One restriction to both programs is they will not tolerate mismatches or ambiguities in their primers or in those sites where they anneal, so all ambiguities must be taken out of the primers and template annealing regions to be tested. The input data list must be in a specially formatted file containing the input patterns known as a pattern data file structured after GCG's restriction enzyme data files. When we took advantage of the "Save primers found to a pattern file" option, Prime generated this type of file for us. Therefore, you should now have two successful primer pattern data files in your working directory.

Use the "**Output Manager**" to take a look at your pattern data files again. The pattern data file format begins with some helpful documentation at the top of the file. Two periods ".." are essential in all GCG data files for separating header information from the data below; they are very important in all GCG programs! Each entire pattern needs to be on one line apiece without any gaps; it needs to be prefaced with a name, the offset number 1, and followed with an optional overhang number 0. The exact column in which the various fields appear is not important, but the order of the fields is vital. Comments can be embedded anywhere by placing an exclamation point before them. "**Close**" the data files and the "**Output Manager**" when finished looking.

For our final primer evaluation we'll use the Prime program again. However, we'll use it to evaluate these pattern data files of suggested primers against our entire BackTranslated DNA template, rather than using it to locate appropriate primers within the conserved candidate sequences as we did just above. Be sure that your complete BackTranslated full-length sequence is loaded into the SeqLab Editor and select it. (If not, then load it through the "Add Sequences" or "Output Manager" menus and select it. An alternative is to wait until a FindPatterns search finishes in order to use the best matching template sequence available for further primer testing and evaluation.) Launch "**Prime**" through the "**Functions**" or "**Windows**" menu, just like before, but you'll be using a 'magic' combination of options from the previous individual forward and reverse runs (if you had to use more permissive options). This will be a combination of the more relaxed parameters from each, or, if necessary, turn off all constraints with the "Ignore. . ." option. Begin by resetting the program to its defaults by selecting the "**GCG Defaults**" button in the main Prime program window. Next, reset your "**Primer Length**" "**Minimum**" and "**Maximum**" values to match what you used before. Now adjust the "**Maximum**" "**PCR Product Length**" up to the maximum allowed, the full length of your BackTranslated consensus sequence. Next, check "**save results as features in file**" "**prime.rsfs**" to add primer location annotation to our RSF file. You'll see what this means below.

Choose "**Options. . .**" next. We need to specify our pattern data lists of suggested primers in the "**Prime Options**" menu, so click on the check box next to "**Select forward primers from file**" and also on the check box next to "**Select reverse primers from file**" and then click on the "**Forward Primers. . .**" button. This will produce a dialog box, "**Forward\_Primer Chooser for Prime;**" click on its "**Forward\_Primer Data File. . .**" button, and use the "**File Chooser**" to pick your forward primer data file and then check "**OK**". "**Close**" the "**Forward\_Primer Chooser for Prime**" window. Repeat this step with the "**Reverse Primers. . .**" button. The "**Prime Options**" window should now show that you are using your specified files. Since we are now dealing with paired PCR primers, we don't want to check forward or reverse primers only. Therefore, be sure "**Select:**" "**primers on both strands for PCR**" is selected. Again use the "**Save primers found to a pattern file**" option, but specify a new output data file name (e.g. "**prime.final.dat**"). "**Close**" the "**Options**" window and then press "**Run**" in the Prime main window after you get all your settings specified.

Just as before, the results may be frustratingly negative, and you may have to experiment with changing several parameters to discover primer combinations that will work. Now "product GC minimum" and "maximum" (-GCMinProduct and -GCMaxProduct) and "difference in primer TM" (-TMDifference) can be troublesome and may have to be changed. Whether you come up with totally impossible PCR conditions is not indicated by the program, so do not blindly accept the results! As frustrating as Prime can be, it certainly can point out the exact conditions that must be altered from standard PCR reactions in order to have any success in the wet-lab. This may all seem like a genuine pain just to get a couple of primers for PCR; however, realize that successful primers found in this manner will most likely work with all related organisms for this particular gene. You will not have to repeat the experience until you are given a totally different system on which to work. I was again fortunate that default parameters worked fine in my example run with the 'primitive' organisms elongation factor dataset.

When the program successfully finishes and the output is displayed, look through the various files. My complete successful Prime output file is shown below. I've again indicated parameter changes with bold:

PRIME of: input\_14.rsfc{BSC5936.week3.entire} ck: 513 from: 1 to: 1581 January 19, 2003 15:47

INPUT SUMMARY

-----

Input sequence: /users/thompson/.seqlab-mendel/input\_14.rsfc{BSC5936.week3.entire}

**Input forward primer list: /home/thompson/BSC5936.week3.prime.forward.dat with 25 primers.**

**Input reverse primer list: /home/thompson/BSC5936.week3.prime.reverse.dat with 25 primers.**

Primer constraints:

**primer size: 25 - 50**  
 primer 3' clamp: S  
 primer sequence ambiguity: NOT ALLOWED  
 primer GC content: 40.0 - 55.0%  
 primer Tm: 50.0 - 65.0 degrees Celsius  
 primer self-annealing. . .  
     3' end: < 8 (weight: 2.0)  
     total: < 14 (weight: 1.0)  
 unique primer binding sites: required  
 primer-template and primer-repeat annealing. . .  
     3' end: ignored  
     total: ignored  
 repeated sequences screened: none specified

Product constraints:

product length: 100 - 1581  
 product GC content: 40.0 - 55.0%  
 product Tm: 70.0 - 95.0 degrees Celsius  
 duplicate primer endpoints: NOT ALLOWED  
 difference in primer Tm: < 2.0 degrees Celsius  
 primer-primer annealing. . .  
     3' end: < 8 (weight: 2.0)  
     total: < 14 (weight: 1.0)

PRIMER SUMMARY

-----

	forward	reverse
Number of primers considered:	25	25
Number of primers rejected for . . .		
primer 3' clamp:	0	0
primer sequence ambiguity:	0	0
primer GC content:	0	0
primer Tm:	0	0
non-unique binding sites:	0	0
primer self-annealing:	0	0
primer-template annealing:	0	0
primer-repeat annealing:	0	0
Number of primers accepted:	25	25

PRODUCT SUMMARY

-----

Number of products considered:	625
Number of products rejected for. . .	
product length:	0

```

        product GC content:           0
          product Tm:                 0
            product position:         0
duplicate primer endpoints:         56
  difference in primer Tm:           243
    primer-primer annealing:        321

Number of products accepted:         5
  Number of products saved:          5
    Maximum overlap between products: 1581 bp

```

THE FOLLOWING PRODUCTS ARE SORTED BY THEIR ANNEALING SCORE

-----

Product: 1

[DNA] = 50.000 nM [salt] = 50.000 mM

PRIMERS

-----

forward primer: forward13  
reverse primer: reverse13

```

                                5'                               3'
forward primer (27-mer):      64 GGTAAGTCAACTACTACTGGTCACCTC 90
reverse primer (25-mer):    1427 TGCTTCATATCACGAACAGCGAAAC 1403

```

	forward	reverse
primer %GC:	48.1	44.0
primer Tm (degrees Celsius):	56.2	56.5

PRODUCT

-----

```

product length: 1364
product %GC: 41.2
product Tm: 76.3 degrees Celsius
difference in primer Tm: 0.4 degrees Celsius
annealing score: 53

```

optimal annealing temperature: 55.4 degrees Celsius

-----

Product: 2

[DNA] = 50.000 nM [salt] = 50.000 mM

PRIMERS

-----

forward primer: forward12  
reverse primer: reverse2

```

                                5'                               3'
forward primer (26-mer):      65 GTAAGTCAACTACTACTGGTCACCTC 90
reverse primer (25-mer):    1424 TTCATATCACGAACAGCGAAACGAC 1400

```

	forward	reverse
primer %GC:	46.2	44.0
primer Tm (degrees Celsius):	54.4	56.3

PRODUCT

-----

product length: 1360  
product %GC: 41.1  
product Tm: 76.3 degrees Celsius  
difference in primer Tm: 1.9 degrees Celsius  
annealing score: 58

optimal annealing temperature: 54.8 degrees Celsius

-----  
Product: 3

[DNA] = 50.000 nM [salt] = 50.000 mM

PRIMERS

-----

forward primer: forward2  
reverse primer: reverse13

forward primer (27-mer): 43 5' GTTATTGGTCACGTTGATTCAGGTAAG 3' 69  
reverse primer (25-mer): 1427 TGCTTCATATCACGAACAGCGAAAC 1403

	forward	reverse
primer %GC:	40.7	44.0
primer Tm (degrees Celsius):	54.7	56.5

PRODUCT

-----

product length: 1385  
product %GC: 41.2  
product Tm: 76.3 degrees Celsius  
difference in primer Tm: 1.8 degrees Celsius  
annealing score: 59

optimal annealing temperature: 54.9 degrees Celsius

-----  
Product: 4

[DNA] = 50.000 nM [salt] = 50.000 mM

PRIMERS

-----

forward primer: forward15  
reverse primer: reverse1

forward primer (29-mer): 62 5' CAGGTAAGTCAACTACTACTGGTCACCTC 3' 90  
reverse primer (25-mer): 1420 TATCACGAACAGCGAAACGACCGAG 1396

	forward	reverse
primer %GC:	48.3	52.0
primer Tm (degrees Celsius):	57.9	59.4

PRODUCT

-----

product length: 1359  
product %GC: 41.2  
product Tm: 76.3 degrees Celsius

difference in primer Tm: 1.5 degrees Celsius  
annealing score: 60

optimal annealing temperature: 55.9 degrees Celsius

Product: 5

[DNA] = 50.000 nM [salt] = 50.000 mM

PRIMERS

forward primer: forward7  
reverse primer: reverse13

forward primer (26-mer): 5' 40 GTTGTTATTGGTCACGTTGATTCAGG 65 3'  
reverse primer (25-mer): 1427 TGCTTCATATCACGAACAGCGAAAC 1403

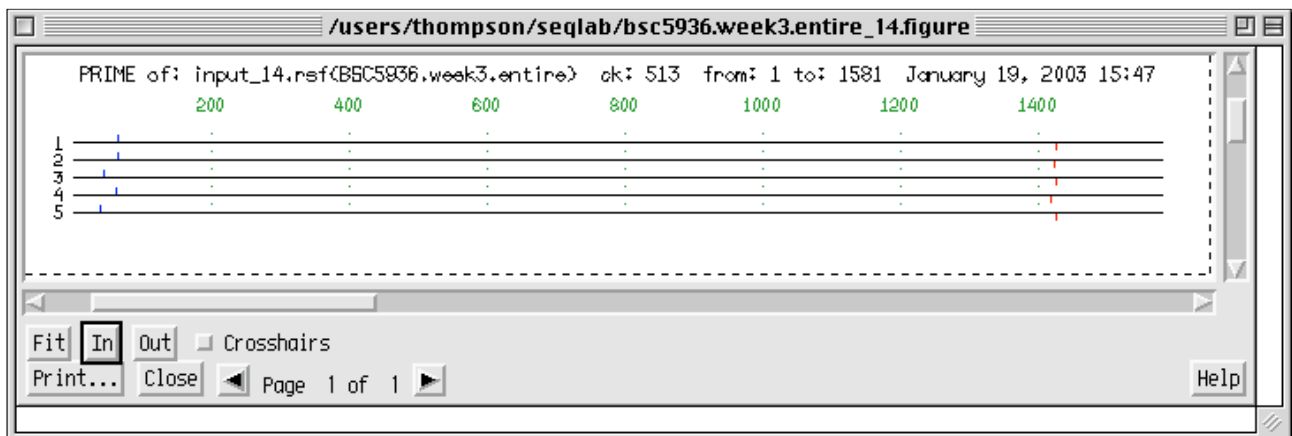
	forward	reverse
primer %GC:	42.3	44.0
primer Tm (degrees Celsius):	55.2	56.5

PRODUCT

product length: 1388  
product %GC: 41.1  
product Tm: 76.3 degrees Celsius  
difference in primer Tm: 1.3 degrees Celsius  
annealing score: 62

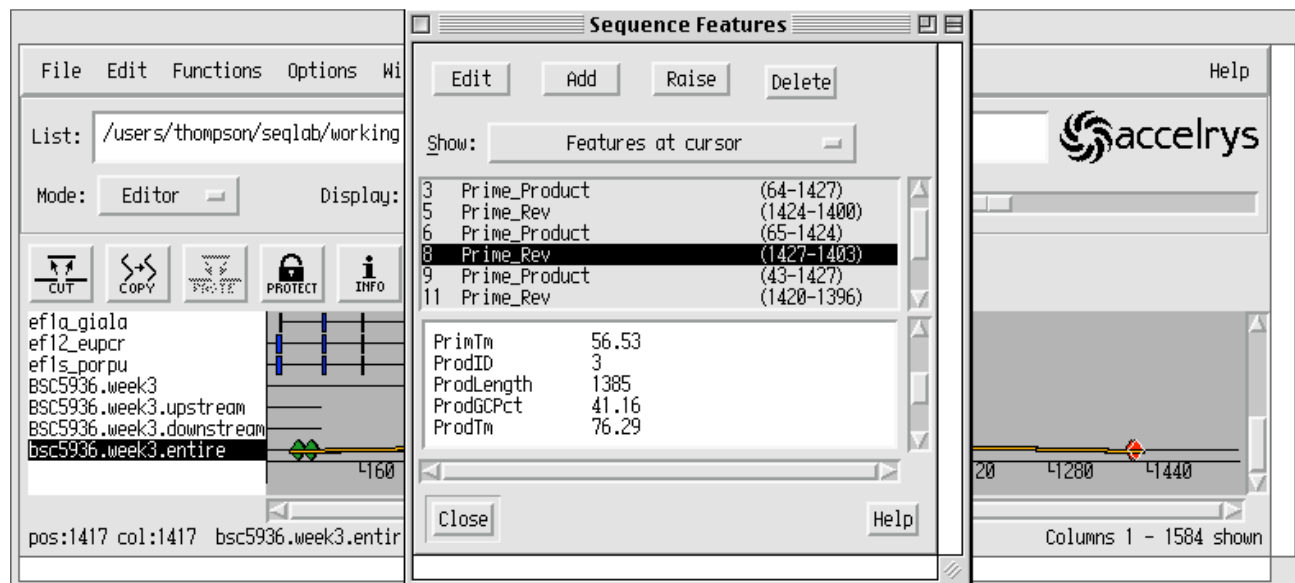
optimal annealing temperature: 55.1 degrees Celsius

The graphic from my successful run is shown below:



Be sure to "Add to Editor" the "prime.rs" file displayed in the "Output Manager." Choose "Overwrite old with new" in the "Reloading Same Sequence" window that pops up. This will merge the new feature information that locates the successful primers and products into your RSF file. Take a look at this new feature information by changing your "Display:" to the "Graphic Features" cartoon representation. Double click on a new feature and then select it in the "Feature" window to see how it is described. My "Sequence

**Features**” window for one of my downstream primers is illustrated below against a backdrop of the Editor window using a “16:1” zoom ratio so that the entire sequence will fit in the window at once:



### Test for Specificity – Will Your Primers Only ‘Find’ the Correct Genes? GCG’s FindPatterns

Another valuable test of any candidate primers should be done before running off to or sending off your request to the oligo’ synthesis lab. You should check for your primers’ specificity to insure that your primers will not hybridize to completely the wrong type of sequence by checking them against the DNA database. This step can also point out, and allow you to correct if necessary, errors in your primer sequence created in the backtranslation step, if enough DNA sequences are available in the database to allow a comparison. The GCG program FindPatterns is probably best for this. It can be used to screen your candidate primers against the entire DNA database, any of the subset databases, or any other sequences desired. There are several advantages to using FindPatterns over standard similarity searching software: 1) you can test more than one primer at a time against as many sequences as you want; 2) the algorithm will not allow any gapping of your primers to the template, which would represent loop structures in the hybrid and should not be allowed; 3) similarities don’t count — identities are required but mismatches are allowed by option; again, just what you want in primer analysis; and 4) word size parameters are not relevant since the algorithm doesn’t use them, therefore, they don’t have to be messed with (which you would need to do if you were using heuristic style similarity searches such as BLAST since they are not designed to find short regions of DNA similarity). For these reasons, I do not recommend using BLAST or FastA style searches for testing primer specificity. The easiest way to run FindPatterns is to provide it with your primers as an input file rather than typing them in interactively. To do this FindPatterns needs its input list of patterns to be in exactly the same pattern data format as described above for Prime. This makes it relatively easy to test them against the database.

However, running a full-blown FindPatterns GenBank search would require too much time for the time constraints of our lab session. If you would like to run this type of analysis for your own research, it is very

important to use appropriate parameters! You need to specify the correct pattern data file and a realistic mismatch level. Give a mismatch level of slightly less than 20% the length of your shortest sequence. The less than 20% mismatch cut-off level is a 'rule-of-thumb' because that is the number of expected mismatches if all codon choices were made on a completely random basis. In the example that I am providing I used a mismatch level of four, but you should probably use less in your own run. FindPatterns is one of the rare GCG programs that's easier to run from the command line than from SeqLab. Not that you can't run it from SeqLab, it's just more bother. If running FindPatterns from the command line, the program will ask you which sequences you want to find your pattern in. These are not your primer sequences; these are the sequences you want to search your primer patterns against. Therefore, answer with either all of GenBank or the appropriate subdivision of GenBank. Since I am trying to find elongation factor 1 $\square$  in lower, primitive eukaryotes, the invertebrate section of GenBank is most relevant (invertebrate:\* which means all of the sequences in the invertebrate subdivision of GenBank). (See the GenHelp User's Guide chapter Using Sequences, topic Using Database Sequences, subtopic Nucleic Acid Database tables, if this still confuses you.) An example FindPatterns command line run is shown below. Notice the GCG -Check command line 'super-option' that lists all of the available options within a program and gives you a chance to use any of them. Remember, do not run FindPatterns on GenBank here today:

```
> findpatterns -check
```

```
FindPatterns identifies sequences that contain short patterns like
GAATTC or YRYRYR. You can define the patterns ambiguously and allow
mismatches. You can provide the patterns in a file or simply type them
in from the terminal.
```

```
Minimal Syntax: % findpatterns [-INfile=]Genbank:Humig* -Default
```

```
Prompted Parameters:
```

```
-PATterns=GAATTC,GGAY          patterns to be found
[-OUTfile=]findpatterns.find    the output file name
```

```
Local Data Files:
```

```
-DATA=pattern.dat              a file with a set of patterns
```

```
Optional Parameters:
```

```
-MISmatch=1                    allows mismatches in the search for your subsequence
-NAMEs                          makes an output file in "file of filenames" format
-ONEstrand                      searches only the top strand of nucleotide sequences
-SIXbase                        searches only for patterns with six or more symbols
-CIRCular                       searches all sequences as if they were circular
Press q to quit or <Return> for more: <rtn>
-ALL                            does an "overlapping-set" search in nucleotide sequences
-PERfect                       looks only for perfect matches
-APPend                         appends the pattern data file to the output file
-SHOW                           shows every file searched even if there are no finds
-TERminal                      writes output to the terminal screen instead of a file
-NOMONitor                     suppresses the screen trace showing each file
-ONCe                          limits finds to patterns found a maximum of 1 time
-MINCuts=1                     limits finds to patterns found a minimum of 1 time
-MAXCuts=3                     limits finds to patterns found a maximum of 3 times
-EXCLude=n1,n2                 excludes patterns found between positions n1 and n2
-SINce=6.90                   limits search to sequences dated on or after June 1990
-BATch                         Submits the program to run in the batch queue
```

```

Add what to the command line ? -data=prime.final.dat -mismatch=4 -batch
FINDPATTERNS in what sequence(s) ? invertebrate:*

What should I call the output file (* findpatterns.find *) ? prime.invert.finds

** findpatterns will run as a batch or at job.

** findpatterns was submitted using the command:
   " at "

```

Because the DNA database is so big and getting bigger all the time, if you ever do this type of search from the command line, be sure to do it in batch mode. GCG has made this an easy chore by providing a -Batch option to many of their CPU intensive programs. Searching GenBank or even any of its subdivisions with FindPatterns takes quite a while to run because FindPatterns doesn't use any heuristics, so I am providing you with that output file. Do not run FindPatterns today against GenBank or any of its subdivisions — just look through and note the types of sequences that were found by my example run displayed below and over the next few pages.

```

! FINDPATTERNS on Invertebrate:* allowing 4 mismatches

! Using patterns from: /users/thompson/BSC5936.week3.prim.final.dat  January 19
, 2003 19:36 ..

      AB002753  ck: 8398  len: 527  ! AB002753 Entamoeba histolytica mRNA
for elongation factor 1 alpha, partial

forward2          GTTATTGGTCACGTTGATTTCAGGTAAG
      25: TTGTC  gttattggtcacgctcgattctggtaaa AGTAC mis=3

forward7          GTTGTTATTGGTCACGTTGATTTCAGG
      22: ATATT  gtcgttattggtcacgctcgattctgg TAAAA mis=3

      AB029058  ck: 4048  len: 1,754 ! AB029058 Anthocidaris crassispina A
cEFP mRNA for elongation factor-1a-rela

reverse13 /Rev    GTTTCGCTGTTTCGTGATATGAAGCA
      1,384: TGGAC  gtttcgctgtccgtgacatgaggca GACCG mis=4

      AB070232  ck: 639   len: 5,045 ! AB070232 Asterias amurensis EF-1a g
ene for EF-1a partial cds. 1/2002

forward13         GGTAAGTCAACTACTACTGGTCACCTC
      7: ACTCC  ggcaagtcaaccaccactggtcatctc ATCTA mis=4

forward12         GTAAGTCAACTACTACTGGTCACCTC
      8: CTCCG  gcaagtcaaccaccactggtcatctc ATCTA mis=4

////////////////////////////////////

AF058282  ck: 7174  len: 1,301 ! AF058282 Trichomonas vaginalis elon
gation factor 1 alpha (tef1) mRNA, part

reverse13 /Rev    GTTTCGCTGTTTCGTGATATGAAGCA
      1,187: CGGCC  gtttcgcatccgtgatatgaagca GACAG mis=3

reverse2 /Rev     GTCGTTTCGCTGTTTCGTGATATGAA
      1,184: ACTCG  gccgtttcgcatccgtgatatgaa GCAGA mis=4

reverse1 /Rev     CTCGGTTCGTTTCGCTGTTTCGTGATA

```

1,180: CACCA ctcggccgtttcgccatccgtgata TGAAG mis=4

AF124488 ck: 6789 len: 1,082 ! AF124488 Chromolepida pruinosa elongation factor 1-alpha gene, partial cds

reverse1 /Rev CTCGGTCGTTTCGCTGTTTCGTGATA  
1,057: CTCCA ttgggtcgccttcgctgtacgtgata T mis=4

AF150983 ck: 6126 len: 1,084 ! AF150983 Bonjeania clamosis elongation factor 1-alpha gene, partial cds. 6

reverse2 /Rev GTCGTTTCGCTGTTTCGTGATATGAA  
1,061: CTTGG gtcgcttcgctgtacgtgatatga mis=3

reverse1 /Rev CTCGGTCGTTTCGCTGTTTCGTGATA  
1,057: CACCC ttgggtcgccttcgctgtacgtgata TGA mis=4

AF172083 ck: 3759 len: 1,314 ! AF172083 Paramecium tetraurelia translation elongation factor 1-alpha (EF-

forward2 GTTATTGGTCACGTTGATTCAGGTAAG  
34: TCGTC gttattggacacgctcgattcaggaaaa TCAAC mis=4

forward7 GTTGTATTGGTCACGTTGATTCAGG  
31: ATCTC gtcggttattggacacgctcgattcagg AAAAT mis=3

AF190771 ck: 8152 len: 1,204 ! AF190771 Acrasis rosea elongation factor 1 alpha (tef1) gene, partial cds.

reverse2 /Rev GTCGTTTCGCTGTTTCGTGATATGAA  
1,179: ACTCG gacgtttcgcccgttcgtgacatgag A mis=4

reverse1 /Rev CTCGGTCGTTTCGCTGTTTCGTGATA  
1,175: CACCA ctcggacgtttcgcccgttcgtgaca TGAGA mis=3

AF198110 ck: 1492 len: 3,300 ! AF198110 Oxytricha trifallax eukaryotic release factor 3 GTPase subunit (e

forward2 GTTATTGGTCACGTTGATTCAGGTAAG  
1,413: TAGTA tttattggtcacgctcgatgcaggtaaa TCAAC mis=4

forward7 GTTGTATTGGTCACGTTGATTCAGG  
1,410: GTTTA gtatttattggtcacgctcgatgcagg TAAAT mis=4

AF278663 ck: 4707 len: 354 ! AF278663 Coloceras savoi elongation factor 1 alpha gene, partial cds. 10/2

reverse1 /Rev CTCGGTCGTTTCGCTGTTTCGTGATA  
332: CTCCT ctcggacgtttcgctgtgcgtga mis=4

AF278664 ck: 5540 len: 354 ! AF278664 Coloceras sp. KPJ-2000 elongation factor 1 alpha gene, partial cd

reverse1 /Rev CTCGGTCGTTTCGCTGTTTCGTGATA  
332: CTCCT ctcggacgtttcgctgtgcgtga mis=4

AF278665 ck: 5576 len: 354 ! AF278665 Coloceras chinense elongation factor 1 alpha gene, partial cds. 1

reverse1 /Rev CTCGGTCGTTTCGCTGTTTCGTGATA  
332: CTCCT ctcggacgtttcgctgtgcgtga mis=4

AF278666 ck: 5768 len: 354 ! AF278666 Physconelloides ceratoceps haplotype 1 elongation factor 1 alpha

reverse1 /Rev CTCGGTCGTTTCGCTGTTTCGTGATA  
332: CTCCT ctcggacgtttcgctgtacgtga mis=4

AF278667 ck: 5384 len: 354 ! AF278667 *Physconelloides cubanus* elongation factor 1 alpha gene, partial c

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
332: CTCCT ctcgacggttcgctgtacgtga mis=4

AF278668 ck: 4941 len: 354 ! AF278668 *Physconelloides ceratoceps* haplotype 2 elongation factor 1 alpha

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
332: CTCCT ctcgacggttcgctgtacgtga mis=4

AF278669 ck: 5037 len: 354 ! AF278669 *Physconelloides ceratoceps* haplotype 3 elongation factor 1 alpha

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
332: CTCCT ctcgacggttcgctgtacgtga mis=4

AF278670 ck: 6484 len: 354 ! AF278670 *Auricotes rotundus* elongation factor 1 alpha gene, partial cds. 1

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
332: CTCCT ctcgacggttcgctgtacgtga mis=4

AF278671 ck: 5753 len: 354 ! AF278671 *Campanulotes compar* elongation factor 1 alpha gene, partial cds.

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
332: CTCCT ctcgacggttcgctgtacgtga mis=4

AF278672 ck: 5431 len: 354 ! AF278672 *Physconelloides spenceri* elongation factor 1 alpha gene, partial

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
332: CTCCT ctcgacggttcgctgtacgtga mis=4

AF278673 ck: 4230 len: 354 ! AF278673 *Physconelloides anolaimae* elongation factor 1 alpha gene, partial

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
332: CTCCT ctcgacggttcgctgtgacgtga mis=4

AF302930 ck: 8150 len: 409 ! AF302930 *Corcyra cephalonica* elongation factor-1 alpha (EF-1a) mRNA, parti

reverse13 /Rev GTTTCGCTGTTCGTGATATGAAGCA  
223: CGGTC gtttcgcccgtgcgtgacatgaggca AACAG mis=4

reverse2 /Rev GTCGTTTCGCTGTTCGTGATATGAA  
220: CCTCG gtcgtttccgctgcgtgacatgag GCAA mis=4

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
216: CACCC ctcggtcgtttccgctgcgtgaca TGAGG mis=3

AF423812 ck: 2826 len: 1,089 ! AF423812 *Hemileuca* sp. L6 elongation factor 1-alpha (Ef-1a) gene, partial

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
1,066: CTCCC ctcgacggttcgctgtacgtgac mis=4

AF423823 ck: 5842 len: 1,089 ! AF423823 *Hylobittacus apicalis* elongation factor 1-alpha (Ef-1a) gene, par

reverse1 /Rev CTGGTCGTTTCGCTGTTCGTGATA  
1,066: CTCCA cttggacggttcgctgttccgtgac mis=4

//

Databases searched:  
GenBank, Release 131.0, Released on 15Aug2002, Formatted on 13Sep2002

Total finds:	121
Total length:	586,568,878
Total sequences:	134,508
CPU time:	17:59.01

Mainly elongation factor 1 $\alpha$  sequences were found by my new primers — excellent.

### Homework assignment

Sometime during the week log onto Mendel. You won't need SeqLab so you can do it with any ssh connection, with or without X. Find your final primer pattern data file and submit the type of FindPatterns search described above. However, use a mismatch level of two, not four. Be sure to specify an appropriate subdivision of GenBank for your project molecular system, not Invertebrate as I used. If submitted in batch mode, as I show above, you'll be able to log out and let Mendel do its search. Log on again after an hour or so and the result will be in your account. Use the Pine mailer as I described last week to send me the file at [stevet@bio.fsu.edu](mailto:stevet@bio.fsu.edu).

### Conclusion

This is the conclusion of today's computer laboratory. Please exit SeqLab and log off Mendel and then quit any ssh and/or X windowing software that you used. Log off the teaching lab computer but do not shut it down. I hope that you all will have come to realize the tremendous help that computational technology can be in this area by going through today's tutorial. Obviously the same general ideas taught here can be tailored to any particular system for the design of primers to any level of specificity. Contact me for further assistance.

### Supplement

Back in the wet lab you would have synthesized or ordered oligo's (and labeled them, if doing hybridization), performed the PCR reaction or hybridization screen, and isolated the products with plaque/colony purification or direct PCR purification, as appropriate.

After you found a candidate sequence, what next? Often it's restriction mapping. The unknown stretch of DNA is restriction digested with various enzymes and agarose gel electrophoresed. The resultant fragment sizes are extrapolated from migration distances. From this information a tentative restriction map can be hypothesized. This type of restriction mapping, i.e. reconstructing a physical map based on overlaps without having an actual sequence, is computationally very difficult. Few automated solutions exist. Alternative strategies include subcloning the pieces into a manageable vector and then sequencing those fragments or direct PCR product sequencing.

After generating sequence data, the other type of restriction mapping, where you know the sequence and merely want to know where all the various restriction enzymes may cut, can be very helpful. The GCG programs, Map, MapPlot, MapSort and PlasmidMap can all assist in guiding and illustrating this process.

Once all cut sites have been mapped, SeqLab, or the stand-alone sequence editor SeqEd, can be used to actually perform the subcloning operation on the computer before doing it in the wet lab.

## References

- Cherfas, J. (1990). Genes unlimited. *New Scientist* **14**, 29–33.
- Genetics Computer Group (GCG<sup>®</sup>), (Copyright 1982-2002) *Program Manual for the Wisconsin Package<sup>®</sup>*, version 10.3, [http://www.accelrys.com/products/gcg\\_wisconsin\\_package/index.html](http://www.accelrys.com/products/gcg_wisconsin_package/index.html) Accelrys, a wholly owned subsidiary of Pharmacia Inc., San Diego, California, U.S.A.
- Gribskov, M., Luethy, R., and Eisenberg, D. (1989). Profile analysis. *Methods in Enzymology*, **183**, 146–159, Academic Press, San Diego, California, U.S.A.
- Gupta, S. K., Kececioğlu, J., and Schaffer, A.A. (1995) Making the shortest-paths approach to sum-of-pairs multiple sequence alignment more space efficient in practice, *Proc. 6th Annual Combinatorial Pattern Matching conference (CPM '95)*.
- Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences U.S.A.* **89**, 10915–10919.
- Mullis, K.B. (1990). The unusual origin of the Polymerase Chain Reaction. *Scientific American* **April**, 56–65.
- Saiki, R.K., Gelfand, D.H., Stoffel, S., Scharf, S.J., Higuchi, R., Horn, G.T., Mullis, K.B., and Erlich, H.A. (1988). Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science* **239**, 487–491.
- Sambrook, J., Fritsch, E.F., and Maniatis, T. (1989). Synthetic oligonucleotide probes. In *Molecular Cloning A Laboratory Manual, 2nd ed.* (pp 11.2-11.53), Cold Spring Harbor Laboratory Press, New York, New York, USA.
- Schwartz, R.M. and Dayhoff, M.O. (1979). Matrices for detecting distant relationships. In *Atlas of Protein Sequences and Structure, 5, Suppl. 3*, (pp; 353-358), National Biomedical Research Foundation, Washington, D.C., U.S.A.
- Smith, R.F. and Smith, T.F. (1992). Pattern-Induced Multi-sequence Alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for comparative protein modelling. *Protein Engineering* **5**, 35–41.
- Smith, S.W., Overbeek, R., Woese, C.R., Gilbert, W., and Gillevet, P.M. (1994) The Genetic Data Environment, an expandable GUI for multiple sequence analysis. *Computer Applications in the Biosciences* **10**, 671–675.
- Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research* **22**, 4673–4680.
- White, T.J., Arnheim, N., and Erlich, H.A. (1989). The Polymerase Chain Reaction. *Trends in Genetics* **5**, 185–189.
- Wood, W.I. (1987). Gene cloning based on long oligonucleotide probes. *Methods in Enzymology* **152**, 443–447, Academic Press, San Diego, California, USA.