

BSC4933/5936: Introduction to Bioinformatics

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

DNA Sequencing, Contig Assembly, and Restriction Enzyme Mapping

Week Four, Tuesday, September 16, 2003

Author and Instructor: Steven M. Thompson

The GCG Fragment Assembly Package (FAS), plus —

How to get sequencing fragment data from an automated sequencer into the computer and assembled into one continuous sequence, and then how to perform restriction enzyme mapping and compositional analysis on that contig for subcloning and other purposes.

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

*GCG[®] is the Genetics Computer Group, part of Accelrys Inc., a subsidiary of Pharmacia Inc.,
producer of the Wisconsin Package[®] 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.

Fragment assembly systems: the best thing going for coping with DNA sequencing data!

DNA sequencing data can be voluminous and perplexing; its management is a formidable task. Many packages exist for assembling and managing this sort of data. They all build up complete DNA sequences from individual sequencing fragments and manage the myriad of data obtained in DNA sequencing experiments. They turn a sometimes dreaded and often tedious job, that of recreating an entire sequenced stretch, into a manageable proposition. One of the best known packages is from the University of Washington's Genome Center (<http://www.genome.washington.edu/>). It has three components Phred, Phrap, and Consed (<http://www.phrap.org/>). The Institute for Genomic Research's (<http://www.tigr.org/>) Lucy and Assembler programs are also very popular. Sequencher (<http://www.genecodes.com/>) is a very popular desktop computer based approach. And Celera's (<http://www.celera.com/>) Whole-Genome Assembly (WGA) system proved the world wrong when they independently assembled a 'shotgun' strategy mishmash of the entire human genome (Venter, et al., 2001, <http://www.sciencemag.org/content/vol291/issue5507/>).

The Wisconsin Package's version is called the Fragment Assembly System (FAS). They also sell a much more powerful system that fully integrates into SeqLab (FAS doesn't) called SeqMerge, but it is quite expensive. FSU has not purchased SeqMerge and has no plans to do so. FAS, however, is powerful enough for most research projects at FSU. It can build a continuous DNA sequence from up to 1,650 individual fragment sequences up to a total contig length of 200,000 bases (380,000 bases per overall project). Each individual fragment has a maximum length restriction of 2,500 bases. FAS is based on an "electronic notebook" concept similar to Roger Staden's (1980); its editor, GelAssemble, was developed from Dr. William Gilbert's MSE program at the Massachusetts Institute of Technology. The Fragment Assembly System has five objectives:

- 1) to provide a manageable method for storing fragment sequence data in a project database that remains invisible to the user so that the intricacies of data and file manipulations are not necessary for the user to tackle.
- 2) to recognize the overlaps between separate fragments and perform alignments of those fragments; yet allow for . . .
- 3) user manipulation and editing of these alignments so that one is not 'trapped' into accepting anything that the system suggests and any alterations can be performed in an interactive 'on-screen' manner.
- 4) to display the alignment and allow export of any part or all of it to standard GCG file format in either a base-by-base sequence file or in a pictorial "Big Picture" representation.
- 5) Finally, the system generates a consensus, at any point desired within the process, based on your accepted alignment using standard ambiguity codes and it can easily export that consensus into your directory structure.

The Fragment Assembly System has been extensively modified to become more powerful and intuitive since it was first introduced with GCG version 7. GelMerge both discovers and assembles overlaps. Contigs are automatically assembled, thus avoiding a user required assembly step. The GelAssemble program is used as an editor for checking and manipulating alignments. GelMerge can optionally automatically excise designated vector sequences. GelMerge has a whole slew of options and the GelAssemble editor has several powerful features too; therefore, be sure to carefully review the documentation before attempting to take advantage of the fragment assembly package. Furthermore, GCG provides a very good overview essay of the Fragment Assembly System in their Program Manual — please take the time to read that essay (<http://www.csit.fsu.edu/gcg/gelinintroduction.html>).

Fragment assembly packages can provide an incredible relief from the reams of paperwork necessary in 'old-fashioned,' traditional sequencing data management. It can free up massive amounts of time to allow the investigator to concentrate more on the research and less on the tedium of any given project. Few people still manually run and read gels because it is so very time consuming; likewise, there is absolutely no sense in not utilizing the computer to manage the generated data — directly input the fragments to the computer and let it do the work.

Your Project Molecular system choices

The following molecules are again listed for your reference. Please maintain using the same one as in the previous tutorial. This really is important. Make special note of its number in this list:

- 1) higher plant ribulose biphosphate carboxylase/oxygenase, small subunit only
- 2) vertebrate P21 ras proto-oncogene transforming protein
- 3) vertebrate basic fibroblast growth factor
- 4) fungal Cu/Zn superoxide dismutase

Week 4 Tutorial: A 'Real-Life' Project Oriented Approach. Fragment Assembly

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-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!)
```

Preliminary preparations

List a directory for your account (**ls**) and notice all of the files left over from last week's tutorial. They really tend to accumulate quickly, especially while using SeqLab. I would suggest looking through them (**more**) and remove (**rm**) any that you don't want to save. Be sure to save your RSF file from last week and the results from your FindPatterns search; we'll be using them later. Furthermore, we should probably start using subdirectories to store each week's tutorial data, so create a new subdirectory (**mkdir**) for last week's data and move (**mv**) any files that you want to keep from last week into it. Next, make a subdirectory for this week's data and then change directory (**cd**) into it.

Regardless of how you've established the X-tunneled ssh connection, after you've logged onto Mendel and taken care of these file maintenance chores, launch SeqLab with the following command (but remember with SSH/XWin32 you need to launch "xclock &" first):

```
> seqlab &
```

Next, it would probably be helpful to change your SeqLab working directory to your present location so that everything that you do today will automatically be saved in your new directory rather than your main 'home' directory. Do this with SeqLab's "Options" "Preferences. . ." "Working Dir. . ." button.

FAS sessions using fragments from data sets provided an entire sequence

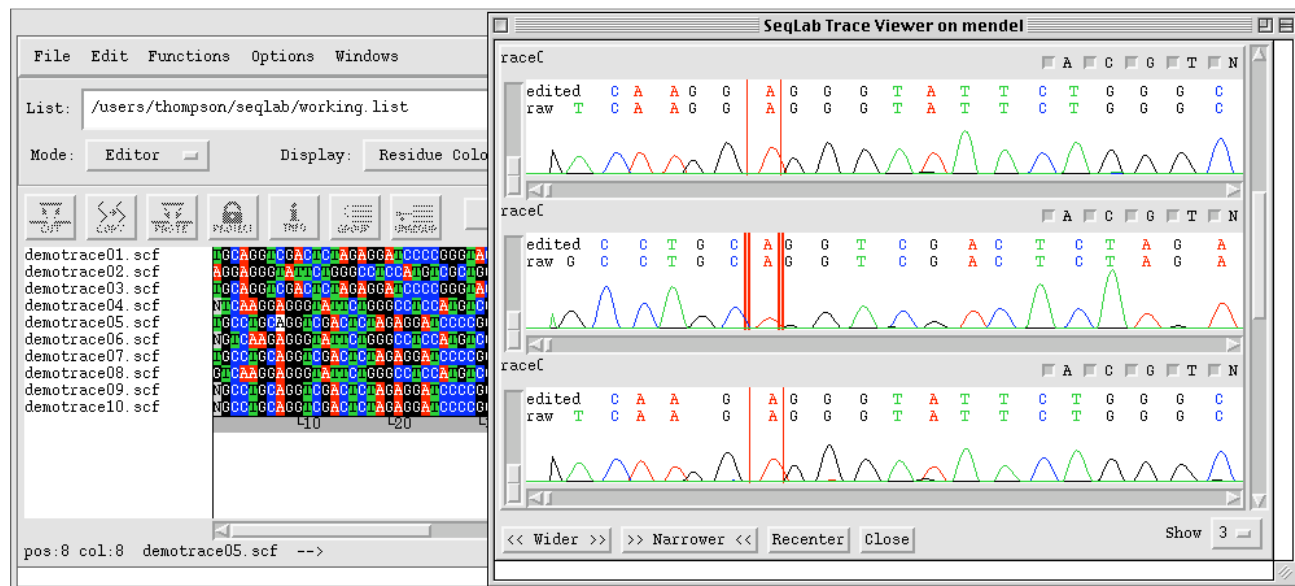
In an actual laboratory situation I would suggest directly entering fragment sequences as the data comes off the sequencer so that your chromatograph trace information is not lost. SeqLab has this ability in its Edit mode under the "File" menu "Import" function. However, the sequences that I have created for you to

assemble did not come off of an automated sequencer; they were modified from existing GenBank files. I purposely placed mistakes in the overlaps of these fragments to force some interaction with the fragment assembly package; otherwise the system would automatically assemble the entire sequence with no user intervention — not the objective of a learning experience.

Make sure that you are in the directory created above — stay in it for the duration of this tutorial except where otherwise noted. Also, please read the introduction to the FAS package within the GCG Program Manual (<http://www.csit.fsu.edu/gcg/gelintroduction.html>) before beginning the tutorial — most of the other FAS documentation is helpful reference but does not require comprehensive reading; however, this section is essential! Therefore, launch a Web browser, read the intro' and keep it open throughout today's session.

Practice session with GCG supplied data

To assist in learning the system GCG has installed several sets of practice sequencing fragments in their GenTrainData data libraries. Before using your Project Molecule data, run through the fragment assembly system with one of the example sets provided by GCG. One of these practice sets has chromatograph trace data as well as sequence data in it. To use these fragments in a practice session so that you can see how trace data can be visualized with SeqLab switch to **“Editor”** mode without having anything selected in your working list. This should give you an empty SeqLab Editor window. Now go to the **“File”** menu and select **“Import. . .”** Type **“gentraindata:*.scf”** in the **“Import sequences”** file chooser **“Filter”** text box and then press the **“Filter”** button. You'll see a list of ABI style binary trace data files in the **“Files”** window. **<Double-click>** each file individually to load it into the SeqLab Editor and then **“Cancel”** the “Import sequences” window. Now go to the **“Windows”** menu and select **“Traces.”** You'll see the trace data in its own window:



Selecting a base call in the Trace Viewer selects the corresponding base in the Editor and editing a sequence in the Editor updates the sequence in the Trace Viewer. **Close** the Trace Viewer after checking it out.

This dataset contains M13mp18 vector contamination. GelMerge can excise these sequences before creating the alignment, if you specify the sequence name in response to the vector prompt in GelStart (separate more than one vector sequence with commas) along with specifying the option -Excise in GelMerge. GelMerge -NoMerge -Report can write a file with information about what sequences will be excised if -Excise is added to the command line subsequently (for the faint of heart). Let's see how this works in SeqLab. The initialization program GelStart must first be run to use FAS.

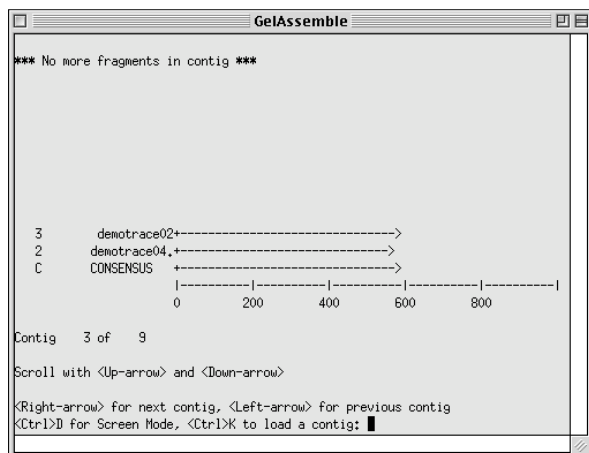
Go to the **Functions** **Fragment Assembly** menu and choose **GelStart. . .** to begin. Specify an appropriate **Project Name,** I used "tracedata," and check **Begin a new project.** Next select **Options. . .** and type **gb:m13mp18** into the **Recognize these vector sequences in GELENER and GELMERGE** text box. This identifies M13mp18 as the contaminant in this project and activates the -Vector option.

Now select all of the sequences in the Editor. Go to the **Functions** **Fragment Assembly** menu and choose **GelEnter. . .** Be sure that **Enter the selected sequences from the Main Window** is selected and press **Run.** GelEnter is either a sequence editor for raw data entry, or it can transfer preexisting files into the FAS database, as we are doing here. Neither GelStart nor GelEnter will give you much indication that they have run, but they only take a few seconds each, so just run one and then the other.

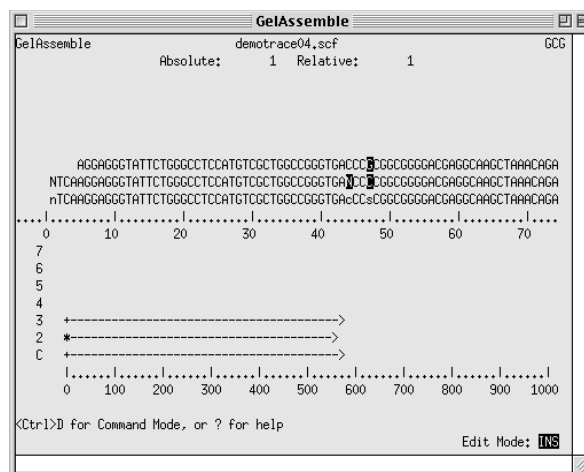
Now that all of the fragments have been loaded into the system, let the overlap program GelMerge discover how, and if, they all fit together. Go to the **Functions** **Fragment Assembly** menu and choose **GelMerge. . .** Leave the defaults as they are in the main program window but punch the **Options. . .** button to read and choose from the lengthy options list available. Notice the extensive option list. This makes the program quite powerful, especially at recognizing and excising vector sequences. The two command line options that we wish to use at this point are -ReportFile and -Excise; these will produce a report file of the found vector sequences, and automatically cut them out of the fragments, if they are located near either end of a fragment, respectively. Therefore, under the section entitled **Vector Recognition and Excision:** check the boxes next to **Remove vector sequences from single-fragment contigs** and **Output file of alignments between fragment and vector sequences,** and give the report file a name that makes sense to you. Accept the rest of the default option parameters. **Close** the "Options" window and press **Run** in the main GelMerge program window. The program reads the fragments, searches for the vector sequences, cuts them out, finds overlaps between the fragments, and aligns them if possible, all in one session. The report file will display. Look it over and then **Close** its window and the "Output Manager"'s window. Notice the comment "Excised . . ." that documents which sections of sequence were cut out by GelMerge's -Excise option. Take a look at the report file that was written, in my case, "tracedata.report:"

```
GELMERGE vector report of Project: tracedata
VECTORS:  m13mp18
```


The window with the contig where two sequences are aligned should look like the panel on the left below:



You'll see the two trace fragments with a consensus of the alignment below them. After selecting the contig that you would like to edit, in this case the number 3 of 9 since the others are either empty or just contain a single fragment, press **<Ctrl-k>**. The screen will change into "Screen Mode" as seen to the right:



Notice that areas of discrepancy are highlighted with inverse video and that the consensus reflects these discrepancies. Press **<Ctrl-d>** to get the command prompt. Explore some of the commands listed in the manual while the contig alignment is in the editor in front of you. Return to the initial Contig List by giving the "contig" command. The "zero-length-contig" can be erased from the database by loading it into the editor and issuing the "erase" command. Exit the GelAssemble editor with the "exit" command.

Run GelView again and notice that the "zero-length-contig" is no longer listed:

```
GELVIEW Fragment Assembly contig display of Project: tracedata
January 26, 2003 12:42

Contig: demotrace01.scf

 2      demotrace01.+----->
 C      CONSENSUS  +----->
          |-----|-----|-----|-----|
          0          20          40          60          80

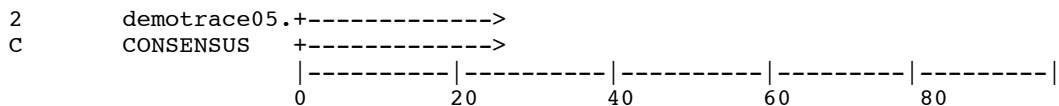
Contig: demotrace03.scf

 2      demotrace03.+----->
 C      CONSENSUS  +----->
          |-----|-----|-----|-----|
          0          20          40          60          80

Contig: demotrace04.scf

 3      demotrace02.+----->
 2      demotrace04.+----->
 C      CONSENSUS  +----->
          |-----|-----|-----|-----|
          0          200         400         600         800
```

Contig: demotrace05.scf



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

9 Fragments in 8 Contigs

Your Project Molecule data and the Fragment Assembly System

Now that you've seen how the system works with a practice data set, begin a new FAS session with your Project Molecule data. Switch from "Editor" to "Main List" "Mode:" to get rid of the trace data presently loaded into the SeqLab Editor. There's no need to save the trace data RSF file unless you really want to. We'll run this session from "Main List" "Mode:" since there's no advantage to using the SeqLab Editor with FAS and there's no trace data in the Project data that I'm supplying. Let's start a brand new list to contain your Project fragment data. Therefore, select "New List. . ." from the "File" menu and give your new list an appropriate name. It's not essential to use the file name extension ".list" but it's a good idea. Check "OK."

You should now be in List Mode with an empty window. Go to the "File" menu and select "Add Sequences From" "Sequence Files. . ." Replace the text in the "Filter" text box with "gentraindata:##.seq" where # is replaced with the number of your Project Molecule from the list (1 for RuBisCO, 2 for P21 Ras, 3 for basic FGF, and 4 for Cu/Zn SOD). Be sure to specify your sequences' Project Molecule number, otherwise you'll get a ton of other stuff. Press the "Filter" button and then select all of the entries that display in the "Files" window by selecting the top-most entry and <shift><clicking> the bottom-most entry. Press the "Add" button to add them all into your new empty list file and then "Close" the "Add Sequences" window.

GelStart must be run every time you start a new FAS session. The first time a project is started you assign it a project name and thereafter always refer to it by that name. GCG's FAS can find your project in any directory in future sessions in this manner. As you saw above, GelStart has options to provide for the recognition of vector and restriction site sequences. This can be very handy for identifying overcloned vector contaminants; however, we will not use it with the Project Molecule data as I can assure you that no vector sequence is included in them. Launch "GelStart. . ." again; remember that once you've used a SeqLab command you can quickly relaunch the same program under the "Windows" menu 'shortcut' listing. Push "GCG Defaults" to reset all of the parameters back to their default settings and then reselect "Begin a new project." Give your project an appropriate name for the protein you are working on and then "Run" GelStart.

Be sure that all of your new fragments are selected in your list and then relaunch "GelEnter. . ." You should see "GelEnter of Selected sequences from Main List." in the program window. Press "Run." GelEnter will read and enter each of the fragments into the database.

Now GelMerge performs the vital job of discovering the overlaps between the fragments and assembling the pieces. The program assembles contigs from individual fragments and previously assembled contigs.

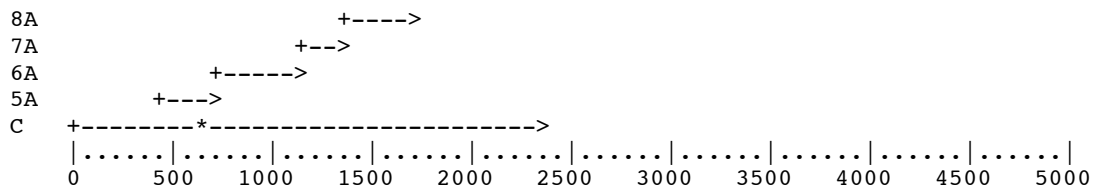
Relaunch **“GelMerge”** to begin the process. Press the **“GCG Defaults”** button again to get rid of the -Excise and -ReportFile options from last time. For the first pass accept the default values for window size, match fraction, and minimum overlap length. The program will read, compare, and assemble the fragment sequences that it can. You may want to check on the status of the job with the “Windows” “Job Manager” depending on the complexity of the dataset, since very complex datasets can take a while to analyze. If no overlaps are found, you will need to decrease the search stringency. However, it is very important not to reduce the stringencies too much ‘right off the bat’ as this would tend to incorporate incorrect overlaps in the contigs.

In order to see how well GelMerge worked we can again use the program GelView. Launch **“GelView,”** this will provide a view of the current status of the project. Your new GelView output file will display the contigs that GelMerge discovered. These are the results of GelMerge at this first pass — don’t get discouraged, with each pass through the system, more will fall into place. Several of the Project Molecule datasets don’t assemble at all on their first pass, and you’ll just see each individual fragments in their own contigs-of-one.

You’ll have to find new overlaps, since, in all likelihood, more than one contig will still be present after your first run through GelMerge. Therefore, run **“GelMerge”** again, only this time decrease the stringency some, for example, by changing the match fraction, that is the **“Minimum fraction of matching words in overlap”** from 0.80 to **“0.60”** in the main program window. Don’t mess with any “Options” parameters. Press **“Run”** and see what happens. Relaunch **“GelView”** to evaluate the results. There may still be several different contigs, rather than just the eventual one that you are striving for. I guarantee that every one of my sample data sets will eventually assemble. I just put a lot of mistakes in the sequences, especially in the 3’ end of them, since that’s where the majority of mistakes really are in sequencing.

Therefore, launch **“GelMerge”** one more time. This time reduce the word size parameter, **“Word size for overlap determination,”** from the default 7 down to **“5”** and change MinOverlapLength, **“Minimum overlap length,”** down to **“10”** from its default 14. The overlap length parameter will likely change on its own to match double the word size. Don’t change the match fraction from 0.60 yet. Press **“Run”** this time and see how successful you were by rerunning **“GelView.”**

GelView can verify the assembly process at any stage. If there is still more than one contig present after this third round, another iteration through the system, using even less stringent parameters should bring in the final alignment. Try decreasing the minimum overlap a little bit more, as well as decreasing the match fraction some. But I would not recommend decreasing the match fraction below 0.4, as this would undoubtedly bring in incorrect alignments. The word size parameter can be decreased to 3 as a last resort, but this dramatically increases computation time. You’ll need to repeat the process as many times as it takes. The key is to never make any big jumps in your parameter adjustments, just reduce them a little at a time. It’s a hassle to undo contig assemblies that are incorrect and much better to slowly merge them in. If things get screwed up so bad that you just want to start over, then the program GelDisassemble can do that for you. Eventually, after you’ve successfully got the entire project to assemble, GelView will show you an entirely different picture:



<Ctrl-D> for Command Mode, or ? for help

Edit Mode: INS

Fragment offset may have to be adjusted slightly to improve alignments in the contig — this is done by adding or subtracting spaces with the space bar and delete key. The space bar will move the sequence right regardless of where the cursor is at, but be careful of the delete key; if it's within a sequence it will delete the character to its left. Adjust the alignment if needed and decide whether to accept or reject the contig. Cursor motion can be controlled with the direction keys and/or with the commands listed in the command summaries of the Program Manual; all GelAssemble commands are also accessible by typing a “?” Several powerful control key commands help you find your way around the edit screen. Some extremely useful cursor Screen Mode commands follow:

- < 1 >< return > and < Ctrl-e > moves you to the beginning and end of a sequence respectively.
 - < relative base # >< return > moves you to that base in the sequence.
 - < # >< arrow key > moves you in that direction # spaces.
 - <<> and <>> moves you one screen at a time left or right respectively.
 - < / > is a find signal; type in the short base sequence you wish to identify, such as a restriction site.
 - < - > rejects the current fragment from the presently displayed contig.
 - < Ctrl-a > and < Ctrl-r > find areas of ambiguity, in the overall contig alignment and in an individual sequence respectively, and < Ctrl-v > finds gaps in the consensus.
 - < Ctrl-l > toggles the “Big Picture” display on and off in order to display larger contigs.
 - < Ctrl-o > toggles Overstrike/Insert mode.
 - < Ctrl-g > causes consensus recalculation.
- Cut and paste in both column and linear mode is available through
 < Ctrl-x > / < Ctrl-l > and < Ctrl-b > / < Ctrl-n > / < Ctrl-p > key combinations respectively.

Several additional features of the screen mode display are worth noting:

- The top line names the current, i.e. cursor position, fragment.
- The second line describes the numeric position of the cursor, both in its fragment and in the entire contig.
- In the Big Picture display an asterisk indicates cursor position and “A’s” at the left margin indicate Anchored fragments; “M’s” indicate Modified ones.
- Finally, in the lower right hand corner the line “Edit Mode:” indicates whether the user is in Insert or Overstrike mode.

Discrepancies are highlighted; check for any areas of highlighting at the junctions. Gaps may have to be introduced to improve alignment of the junctions; insert “n’s” to represent possible deletions in the reading of

the gel. It seems to help if you work on junction problems from the top down and from the right end toward the left; the reasons become apparent as larger contigs are managed — the anchoring concept becomes important and you'll have to worry about it less by working down and in. Any changes made within one sequence will affect all the other alignments as soon as you are working with more than just two. Therefore GCG has built in the anchoring function — pay attention! Any changes made within an anchored fragment are propagated throughout all anchored fragments! This can be a tremendous help and/or a terrible hindrance anytime more than just a pair of fragments are being worked with —be careful. With the cursor on the sequence of concern, use < Ctrl-d > to enter Command Mode to anchor and unanchor sequences with the ANCHOR and NOANCHOR commands respectively. Just press return to get back to Screen Mode. Be careful with anchoring — pay attention to the “A's” in the diagram at the bottom of the screen.

A few tips that I have discovered are:

If you want to add gaps to a single fragment without affecting the rest in the contig, make sure the fragment of interest is unanchored, add your gaps, note your location, move to the beginning of the fragment, delete the same number of spaces as were added, and return to your original place in the sequence to check the results.

The converse strategy, to insert gaps in a whole group but not one sequence, is also helpful. Here, make sure all but the one fragment is anchored and add gaps to one of the anchored group's members, then merely move to the unaffected fragment and space it over into the proper alignment.

Press <Ctrl-g> or issue the command CONSENSUS to generate a new consensus. Always reform the consensus after making any changes. Where discrepancies can not be resolved by the obvious addition or subtraction of bases, just leave the differences and the code of the consensus will indicate the ambiguities.

After all editing, press < Ctrl-d > to reenter command mode, in which global type manipulations can be accomplished; some that I've found very useful follow:

WRITE is the most important command; it is the only way to save changes made to the database! (To get out without saving your work, use the QUIT command.)

[#a,#b] ANCHOR and NOANCHOR performs the anchoring and unanchoring functions respectively on fragments a through b.

CONTIG places you back in the contig preview screen.

SORT is useful if you've drastically changed some fragments' offsets; it reorders them.

[#] OFFSET moves the current fragment to begin at position #.

[#a,#b] SEQOUT writes fragments a through b to output files.

The LOAD command allows a whole different contig to be added onto an existing one and fit in manually.

This can be very helpful in cases where GelMerge fails to discover a known overlap.

After you are satisfied with the assembled contig you must save any changes made while in the editor by issuing the **“write”** command. The contig is written to the database named after the lowermost fragment within it and the cursor returns to its last position in the sequence. Only the currently displayed contig is saved to the database.

Next, if there are more than one contig in the project, you need to perform the same type of manipulations on them also, so reenter command mode and enter the command **CONTIG**. The same contig that you had just worked on will be displayed. However, if there are others to be worked on, you won't want to work on the same one again, so use the cursor direction key to read a different contig and load that one in a similar fashion as before. Again, adjustments are made as necessary, a consensus is reformed, and the contig is saved with **WRITE**.

Finally, a resulting consensus sequence can't do us much good if it's stuck in the assembly system's database. Therefore, enter the command **“segout”** from the command line of Screen Mode with the cursor's current position on the consensus, after you are finished assembling and editing your complete sequence. This will cause the consensus sequence generated by the system to be written out to SeqLab's working directory. GelAssemble will prompt you for a file name; give it something appropriate but do not use the same name as your project. Another couple of handy GelAssemble command functions at this point, or at any other for that matter, are the commands **“prettyout”** and **“bigpicture.”** These write to output files the alignment on a base-by-base sequence and pictorial level respectively. If any of your fragments are anchored, only those anchored fragments will be exported. Therefore, make sure all of your fragments are either anchored or unanchored and then give these two commands, accepting the default file names.

When all of your contigs have been processed and you are satisfied with all the alignments, exit GelAssemble by reentering command mode and typing **“exit.”** The current contig is written and you are returned to SeqLab's Main Window. We won't need SeqLab anymore today so go ahead and **“Exit”** SeqLab.

Corroborate your sequence with the ‘real thing’

Next, we're going to cheat here and bring in the actual sequence. Since any true sequencing project would use more than just one fragment per stretch of DNA, usually sequencing both the forward and reverse strands twice apiece, I suppose it's not really cheating that bad. Remember at the end of last week's tutorial you ran FindPatterns of your probe/primers against the appropriate subdivision of GenBank; you will check and use those results now. It should corroborate what you found in your preliminary database searches in Week Two. Here we are interested in the genomic sequence, not the mRNA/cDNA.

Move over into the subdirectory (**cd ../subdir**) that you created for those results at the beginning of today's tutorials in your xterm window and take a look at your FindPatterns output file. This file is liable to be pretty huge so use the search function in more to try to find the relevant entry. Display the file with the **“more”** command, then once the file has loaded, type a diagonal slash **“/”** and specify the search string. You only

have to type the search phrase once, thereafter, just type the slash and press return. You are looking for entries that refer to the genomic or gene sequence but not mRNA/cDNA entries. This may frustrate you, but I want you to look anyway. It's good practice.

In all cases the sequences you need are genomic and not mRNA/cDNA, so be sure to pick entries that correspond to a genomic DNA sequence for your chosen protein. The entry name will be all uppercase followed by the accession code and title line in the FindPatterns output. You also should have found these genomic entries while database browsing in Week Two. In some cases there may be genomic sequences available for your particular protein from more than one organism of the desired sort, and in other cases the genomic sequence from one organism may be spread over more than one entry. In other cases your FindPatterns file may not even have the correct entry. Therefore, check with your lab instructor before proceeding with the next step to be sure that you are choosing the correct genomic sequence(s); otherwise, it will not work.

After perusing your FindPatterns' results move back to your Week Four subdirectory and use GCG's global pair-wise dynamic programming alignment program Gap to compare your protein's actual DNA databank entry to your assembled sequence's consensus using the appropriate sequence specification. If your sequence's database entry is in multiple pieces, then you will need to run Gap analyses of each one separately using the results of the first to help guide the starting position of the subsequent. Gap will try to align each piece from the very beginning of your consensus sequence otherwise. This command should be issued in your ssh/xterm window, since we've already exited SeqLab and I want you to see how things run at the command line as well as from SeqLab. Remember, don't use my example elongation factor sequence; use your own genomic sequence as verified with your instructor:

```
> gap
```

```
GAP uses the algorithm of Needleman and Wunsch to find the alignment of
two complete sequences that maximizes the number of matches and minimizes
the number of gaps.
```

```
GAP of what sequence 1 ? gb:humef1a (but use your own Project genomic entry!)
```

```
Begin (* 1 *) ? <rtn>
End (* 4695 *) ? <rtn>
Reverse (* No *) ? <rtn>
```

```
to what sequence 2 (* gen1:humprp *) ? ef1a.cons (but use your own FAS consensus!)
```

```
Begin (* 1 *) ? <rtn>
End (* 4695 *) ? <rtn>
Reverse (* No *) ? <rtn>
```

```
What is the gap weight (* 50 *) ? <rtn>
```

```
What is the gap length weight (* 3 *) ? <rtn>
```

```
What should I call the paired output display file (* humef1a.pair *) ? humef1a.g
elpair (but give it a name that makes sense in your own case!)
```

```
Aligning .....
```

```
Gaps:      0
Quality: 46950
Quality Ratio: 10.00
% Similarity: 100.000
Length: 4695
```

The screen trace shows the percent similarity of the two sequences and the output file can be used to locate the exact position of any mismatches.

What else? Restriction enzyme mapping and compositional analysis

In a real lab situation the next step after sequencing often involves cloning your sequence into an appropriate vector where your protein of interest can be overexpressed for biochemical analysis. One of the most important computer analyses for cloning is restriction enzyme mapping. 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. Let's continue to use the command line for this section and see how one of the GCG restriction mapping programs works. Issue the command "map" in your ssh/xterm window. Specify your genomic entry from above (again, not my example!), cut with all enzymes, and tell the program no translations. Follow my example below:

```
> map

Map maps a DNA sequence and displays both strands of the mapped sequence
with restriction enzyme cut points above the sequence and protein
translations below.  Map can also create a peptide map of an amino acid
sequence.

(Linear) MAP of what sequence ?  gb:humef1a      (but use your own Project genomic entry!)

      Begin (* 1 *) ?  <rt>
      End (* 4695 *) ?  <rt>

Select the enzymes:  Type nothing or "*" to get all enzymes. Type "?"
for help on which enzymes are available and how to select them.

      Enzyme(* * *) :  <rt>

What protein translations do you want:

      a) frame 1   b) frame 2   c) frame 3
      d) frame 4   e) frame 5   f) frame 6

      t)hree forward frames   s)ix frames   o)pen frames only
      n)o protein translation   q)uit

Please select (capitalize for 3-letter) (* t *) :  n

What should I call the output file (* humef1a.map *) ?  <rt>

Mapping .....

Writing ..... ..
MAP complete with:

      Sequence Length:   4,695
```

```
Enzymes Chosen:    222
Cutsites found:   1,249
CPU time:         00.18
```

Output file: humefla.map

Check out your new restriction map with “**more.**” Finally, another often helpful analysis calculates the composition of a DNA sequence. Launch this program with the command “**composition:**”

```
> composition
```

Composition determines the composition of sequence(s). For nucleotide sequence(s), Composition also determines dinucleotide and trinucleotide content.

```
COMPOSITION on what sequence(s) ? gb:humefla    (but use your own Project genomic entry!)
```

```
Begin (* 1 *) ? <rtn>
End (* 4695 *) ? <rtn>
```

```
What should I call the output file (* humefla.composition *) ? <rtn>
```

```
COMPOSITION complete:
```

```
Sequences: 1
Total Length: 4,695
CPU time: 00.01
```

```
Output file: /home/thompson/tutorials/BSC5936/Week4/humefla.composition
```

Homework assignment

After the conclusion of today’s tutorial use Mendel’s pine mailer to send me two files:

- 1) your consensus sequence from the Project Molecule FAS session, and
- 2) the **.gelpair** file from the above Gap run.

Furthermore, tell me the restriction “**Enzymes that do not cut**” from your Map output. Please don’t take the time to type all of the names out for me; just copy and paste them into your note to me. Copying and pasting in an X window is quite different than in a normal telnet-style ssh window. Use the first, left mouse button to select; this automatically copies the selection. And then paste with the middle mouse button. If you can’t get this to work, just use a normal telnet-style ssh window on any system and copy and paste the way that you are use to.

Finally, start to be thinking about your Final Semester Project. It really helps to make it more relevant and important if it has anything to do with your own graduate research, but this is not essential. Send me some of your preliminary ideas in your homework assignment for the week.

Conclusion

GCG has provided a powerful utility for building up DNA sequences from individual sequencing fragments with the FAS package. It is hard to imagine trying to put together all this information without help from the

computer — unfortunately some biologists still live in the dark ages, as far as computer technology is concerned, and do not utilize this tremendous tool, or similar ones available. Please check it out; learn the system and spread the word! Next week — database similarity searching strategies — what do you have?

References

Genetics Computer Group (GCG[®]), (Copyright 1982-2003) *Program Manual for the Wisconsin Package[®]*, version 10.3, http://www.accelrys.com/products/gcg_wisconsin_package/index.html Accelrys, a wholly owned subsidiary of Pharmacia Inc., San Diego, California, U.S.A.

Staden, R. (1980) A new computer method for the storage and manipulation of DNA gel reading data. *Nucleic Acids Research* **8**, 3673–6694.

Venter, J.C. et al. (2001) The sequence of the human genome. *Science* **291**, 1304–1351.