

BSC4933/ISC5224: Introduction to Bioinformatics

Laboratory Section: Wednesdays from 2:30 to 5:00 PM in Dirac 152.

DNA Sequencing Data: Contig Assembly and Restriction Enzyme Mapping

Lab Four, Wednesday, January 28, 2009

Author and Instructor: Steven M. Thompson

The GCG contig assembly system SeqMerge —

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, *a.k.a.* the Wisconsin Package[®] for sequence analysis,
a 'retired' product of Accelrys Inc..
© 2008 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 dollar sign (\$) indicates the system prompt and should not be typed as a part of commands. Really important statements may be underlined.

Contig assembly systems: how to cope 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, known as contigs, from individual sequencing fragments, and manage the myriad of data obtained in DNA sequencing experiments. They turn an often tedious and dreaded job into a manageable proposition.

The very first such package was included in Roger Staden's "electronic notebook" concept (1980), and all subsequent systems built on its foundation. Most all have editing ability as originally developed in William Gilbert's MSE program at the Massachusetts Institute of Technology. One of the best-known packages is from the University of Washington's Genome Center (<http://www.genome.washington.edu/>). It has three main 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 widely used. Sequencher (<http://www.genecodes.com/>) is a very popular commercial, personal computer based approach. And Celera's (<http://www.celera.com/>) Whole-Genome Assembly (WGA) system proved to the world that they could assemble 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 SeqMerge. It is an X-driven GUI based system plenty powerful enough for any research project at FSU, and learning its principles will help you understand how all of these contig assembly software systems work. They all pretty much have the same five objectives:

- 1) to provide a manageable method for storing DNA 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 overlaps between fragments and align them based on that knowledge; yet allow for . . .
- 3) interactive, 'on-screen' user manipulation and editing of these alignments so that one is not 'trapped' into accepting everything that the system suggests;
- 4) to display the alignment and allow export of any part of it to standard sequence file format or graphical representations thereof; and . . .
- 5) to generate a consensus based on your accepted alignment using standard ambiguity codes.

Additionally most contig assemblers have the ability to recognize and excise vector contaminating sequence regions and low quality read areas. SeqMerge includes all these features and consists of a series of steps that need to be run in a logical order. Launching the program initializes the system and builds the database; fragment data is then entered, and then overlaps are discovered and assembled. The Contig Editor allows for checking and manipulating of the alignments. The Project Manager keeps track of the project database. SeqMerge can optionally recognize and mask designated vector and low quality sequences. SeqMerge has a slew of powerful features. Review the documentation carefully if you need to use SeqMerge in your research. GCG provides a very nice overview tutorial for SeqMerge in their Program Manual (http://www.sc.fsu.edu/gcg/seqmerge_tutorial.html)— please read it today, but don't take the time to actually perform the work. Also a complete PDF version of the SeqMerge Guide is available on HPC at `"/opt/Bio/GCG/doc/SeqMerge/SeqMerge_Guide.pdf."`

Contig assembly systems provide an incredible relief from the reams of manual paperwork necessary in 'old-fashioned,' 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 to not utilize the computer to manage the generated data — directly input the fragments to the computer and let it do the work.

A 'real-life' project oriented approach to contig assembly — your lab project molecule 'scenario'

Remember what we're simulating here, the same sort of procedures as one would go through in a 'wet-lab' investigating a new gene sequence in a previously unsequenced species. Last week we designed primers, and in the meantime we'll imagine that they worked great, that the PCR products cleaned up just fine, and that we got good sequence data from the department's automated sequencing facility. This lab begins with methods for assembling that data.

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) primitive 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

Activate and 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 as well as SeqMerge. 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 HPC with an X-tunneled ssh session. When using an xterm window on Mac OSX or UNIX/Linux issue the following command (remember: the X has to be capitalized and “user” is your account name):

```
$ ssh -X user@submit.hpc.fsu.edu
```

Preliminary preparations

Change directory to last week’s subdirectory and list its contents. Files tend to accumulate very quickly, especially while using SeqLab. Remove any that you won’t need later. Be sure to save last week’s RSF file and the results from your FindPatterns search though. Move back to your home directory and then create a new subdirectory for this week’s data and then move into it.

After you’ve taken care of these file maintenance chores, launch SeqLab with the usual command:

```
$ seqlab &
```

Next, it would again be helpful to change your SeqLab working directory to your present location so that everything that you do today will be saved in your new directory rather than last week’s directory. As before, do this with SeqLab’s “Options” “Preferences. . .” “Working Dir. . .” button.

SeqMerge with your lab project molecule fragment data set → an entire consensus 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. SeqMerge can do it from the “Project Manager.” Both offer a trace viewer and editor. However, the sequences that I have created for you to assemble did not come off of an automated sequencer so they have no trace data associated with them; they were modified from existing GenBank files. I purposely placed mistakes in the overlaps of these fragments to force some interaction with SeqMerge; otherwise the system would automatically assemble the entire sequence without any user intervention — not the objective of a learning experience. I’ve put copies of these sequences in GCG’s public training data files directory for your use.

Make sure that you are in the directory created above — stay in it for the duration of this tutorial except where otherwise noted. Also, [be sure to have read the SeqMerge tutorial](#) mentioned above before beginning this laboratory. The SeqMerge Guide PDF file is helpful reference but does not require comprehensive reading;

however, the tutorial is essential! Therefore, if you haven't already done so, launch a Web browser, read the tutorial, and keep it open throughout today's session. If you're real ambitious, go ahead and do the steps described in GCG's practice tutorial. It's really quick and easy, and it illustrates several aspects of SeqMerge that my tutorial ignores.

Now let's begin a SeqMerge session with your lab project molecule data. Select "**New List. . .**" from the "**File**" menu while in SeqLab's "**List**" "**Mode:**" 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.**" We will load and assemble our project fragment data while in SeqLab's "**List**" "**Mode:**" at this point, as there is no advantage to using "Editor Mode" with SeqMerge.

Go to the "**Functions**" "**Fragment Assembly**" menu and select "**SeqMerge . . .**" and then press "**Run**" and "**OK**" in the two new windows produced. An empty "**SeqMerge Project Manager**" window will appear. Select "**Create Project**" off the "**Project**" menu there. Type an appropriate name for your SeqMerge project directory in the space provided in the "**Create New Project**" window. Do not get rid of your working path designation. The first time a new SeqMerge project is started you assign it a project directory name and thereafter always refer to it by that name. SeqMerge can find your project in future sessions in this manner. Now go to the "**Project**" menu and select "**Add Sequences From**" "**Sequence Files. . .**" Replace any and all text in the "**Filter**" text box with "**gentraindata:#*.rsf**" 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 "**Add Sequence Files**" window by selecting the top-most entry and **<shift><clicking>** the bottom-most entry. Press the "**OK**" button to add them all into your new SeqMerge project, and then "**Cancel**" the "**Add Sequences Files**" window. This will place your fragment data into the "SeqMerge Project Manager" entering each fragment into the project database.

As mentioned previously, SeqMerge can recognize vector sequences. This is done through the Project Manager's "Vector . . ." button and SeqMerge's "Mask" operation. It 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 contaminant sequence is included in them. If interested, be sure to do the GCG practice tutorial as it shows you how to take advantage of this feature.

Next SeqMerge tries to discover the overlaps between the fragments and assemble the pieces. It will assemble contigs from individual fragments and from previously assembled contigs. Select all of the fragments in the "**Project Manager**" by dragging your cursor through them all, or by using the "**Edit**" menu "**Select All Contigs**" button. Then use the "**Assemble**" menu and pick "**Assemble . . .**" to discover the overlaps and assemble your contig. For the first pass accept the default values for "**Word Size**," "**Stringency**," and "**Minimum Overlap**" and "**Identity**" lengths. Press "**OK.**" The program will read, compare, and assemble the fragment sequences that it can. The more complex dataset is, the longer it can take 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.

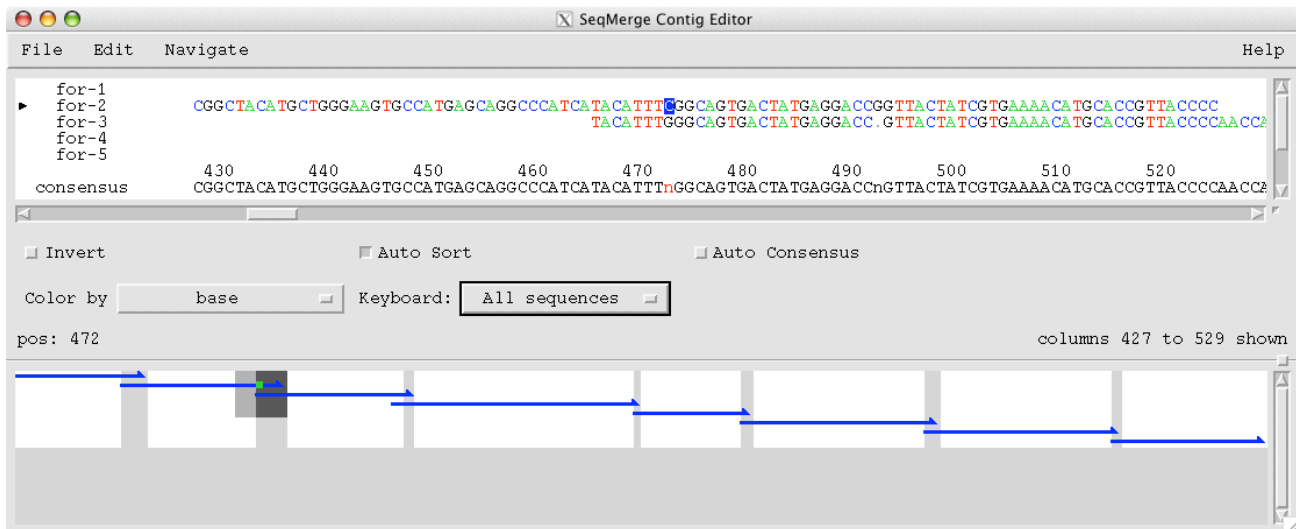
To see how well SeqMerge worked, select a contig that has a plus sign (+) next to it, and then press the **"Edit"** menu **"Edit Selected Contigs"** button (or double-click the selected contig). This will launch the "Contig Editor" where you can check out the assemblies and make modifications if desired. These contig assemblies are the results of SeqMerge's 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 readily assemble; you'll see individual fragments in their own contigs-of-one as well as the assembled contigs in the Manager. Notice the options in the "Contig Editor" menus. This is where vector sequences and low-quality terminal ends can be recognized and masked and even deleted if so desired. You can also launch a trace viewer from this window, though our fragment data does not have any trace data associated with it. Again, if you're interested in this feature, you should go through the GCG practice tutorial, as their fragments do have trace data in them. Click on an overlap area seen in the lower graphical panel of the **"Contig Editor"** to see the exact overlap alignment chosen by the system. Also notice that you can move about your contig alignment using the **"Navigate"** functions. Use the **"File"** menu to **"Close"** the **"Contig Editor"** and return to the **"Project Manager."**

You'll have to find new overlaps, since, in all likelihood, more than one contig will still be present after your first run through the system. Therefore, select all of the contigs in the **"Project Manager"** again, and rerun the **"Assemble"** function, only this time decrease the stringency some, for example, by changing the **"Stringency"** from 0.80 to **"0.70"** in the **"Assemble Project"** window. Press **"OK"** and see what happens. Relaunch the **"Contig Editor"** as above to evaluate the results. There may still be several different contigs, rather than just the eventual one that you are striving for. I guarantee you 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 occur in DNA sequencing.

Therefore, **"Assemble"** your contigs one more time. This time reduce the **"Word Size"** from the default 7 down to **"5"** and change **"Minimum Overlap"** and **"Identity"** down to **"10"** from the default 14. Don't change the stringency from last run's 0.70 yet. Press **"OK"** this time and see how successful you are.

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 stringency some. But I would not recommend decreasing it 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 there is a **"Disassemble"** option that can do that for you.

Eventually, after you've successfully got the entire project to assemble, the "Contig Editor" will show you an entirely different picture, as seen in the graphic here:



Make note of how many runs through the SeqMerge "Assemble" process, and of the final "Assemble" parameters required to create just one final contig. This information will be necessary for the Lab Report.

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 "Navigate" commands. Some useful cursor "Contig Editor" commands follow:

- < Ctrl-b > and < Ctrl-e > moves you to the beginning and end of a sequence respectively.
- < relative base # >< return > moves you to that position in the overall contig assembly.
- < # >< arrow key > moves you in that direction # spaces.
- < < > and < > > moves you one screen at a time left or right respectively.
- < 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.

Several features of the "Contig Editor" are worth noting:

- "Find . . ." in the "Navigate" menu enables you to identify sequence patterns such as restriction sites.
- "Reject" in the "Edit" menu rejects the current fragment from the displayed contig.
- The "Mask" operations in the "Edit" menu allow 'bad' sequence to be ignored or deleted.
- "Overstrike" mode can be toggled on and off through the "Keyboard:" box.
- The numeric position of the cursor in the entire contig is listed at the left of the display.
- Fragments can be moved as a set with the "Group Selected" versus "Ungroup All" "Edit" functions.

“Auto Sort” is useful if you’ve drastically changed some fragments’ offsets; it reorders them.

“Overlay Contig” allows a whole different contig to be added onto an existing one and fit in manually. This can be helpful in cases where SeqMerge fails to discover an overlap you know exists.

The “File” menu “Export” function writes selected fragments or the consensus to RSF format output files.

“Save” in the “File” menu saves changes made to the project database!

Discrepancies can be highlighted with the “Color by” choice box; check for areas of discrepancy 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 helps 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 grouping 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 Group function — pay attention! Any changes made within a grouped fragment are propagated throughout all grouped 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 names of the desired fragments selected, use “Edit” “Group Selected” and you’ll see a group number appear to the left of the fragments’ name. Use “Ungroup All” to sever the connection.

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 not grouped to anything else, 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 grouped and add gaps to one of the group members, then merely move to the unaffected fragment and space it over into the proper alignment.

“File” “Revert” is like an undo button, allowing you to discard recent edits.

The consensus will update as you edit your assembly. Its mode of action can be controlled through the “Consensus Parameters” and “Auto Consensus” functions. Where discrepancies cannot 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 you are satisfied with the assembly save any changes made to write the current contig to the database.

If there are more than one contig in the project that you can’t get to assemble with the others, you’ll need to perform the same type of manipulations on them too (but this won’t be the case with our data).

Finally, a resulting consensus sequence can’t do us much good if it’s stuck in SeqMerge’s project database. Therefore, be sure to use the “File” menu to “**Export**” the consensus after you are finished with the assembly.

This will write the consensus sequence into SeqMerge's current project directory by default, though I suggest that you change the path designation in the "**Output RSF file:**" text box to write it one level up from there, i.e. into your current SeqLab working directory. Also be sure to name it something that makes sense to you. Press "**OK**" in the "**Export**" window. The sequence will be displayed; "**Close**" the file display window. Next, use the "**Output Manager**" to "**Add to Main List**" that consensus sequence. "**Close**" the "**Output Manager.**" Another handy "Contig Editor" function to use at this point, or at any other for that matter, is "Print Contig Graph." This prints or writes to an output file either a PostScript or ASCII representation of your contig assembly graphic.

That's it for SeqMerge — use the "**File**" menu to "**Close**" the "**Contig Editor**" and then use the "**Project Manager's**" "**Project**" menu "**Exit**" button to get out of SeqMerge.

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 at least twice apiece, I suppose it's not really cheating that bad.

Last week's FindPatterns+ search homework may have found the sequence that you need here, but it's a royal pain to find the exactly right sequence amongst all those discovered by the program. You would be looking for entries that refer to the "genomic" or "gene" sequence of your selected molecule, and not mRNA/cDNA entries. You may also have found these genomic entries while database browsing in Lab Two. Remember? In most cases there will 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 still other cases your FindPatterns+ run may not have even found the correct entry. It can all be quite complicated. Therefore, use your SeqLab "**Main List**" "**File**" menu to "**Add Sequences From**" "**Sequence Files**" to bring in the actual genomic sequence. Specify the correct genomic sequence by replacing "# with the correct number of your selected project molecule in the "**Add Sequences**" "**Filter**" text box, as in the beginning of this tutorial with your fragment data:

```
gentraindata:#.genomic
```

Select the entry and then press "**Add**" and then "**Close**" the "**Add Sequences**" window. This will put the sequence specification for your selected project molecule's complete genomic sequence from the GCG public training data directory into your Lab 4 list. "**Save List**" and select both entries there, your new SeqMerge consensus sequence and the 'real' genomic sequence, and then switch to "**Editor Mode.**"

Next we could use GCG's global pairwise dynamic programming alignment program Gap to compare your protein's actual genomic entry to your assembled sequence's consensus, but Gap is another of those programs broken under CentOS 5. Therefore, we'll use EMBOSS's (2000) Needle program. This and GCG's Gap are both implementations of the Needleman and Wunch (1970) global alignment algorithm. You'll learn a

lot more about these types of the dynamic programming algorithm in the coming weeks. Select both sequence names in the Editor display. Go to the “**Extensions**” “**EMBOSS programs**” menu and chose “**Needle**.” We’ll just accept the default program parameters so press “**Run**” in Needle’s main window. The output shows the percent similarity and identity between the two sequences and the exact position of any mismatches between them. Use the “**Output Manager**” to delete the log file, and give your new “.pair” output a name that makes sense, but keep the extension the same.

What else? Restriction enzyme mapping and compositional analysis

In a ‘real’ lab situation the next step after sequencing and contig assembly 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. Unfortunately, of the four, only the ‘plus’ version of Map, “Map+,” continues to work under CentOS 5. Once all cut sites have been mapped SeqLab can be used to actually perform the subcloning operation on the computer before doing it in the wet lab. Let’s see how the GCG Map+ restriction mapping program works. Select just your true genomic sequence (not the exported consensus sequence from SeqMerge) in SeqLab’s Editor window and launch “**Map+**” from the “**Functions**” “**Mapping**” menu. Accept the default “**enzyme.dat**” file that cuts with all enzymes, and tell the program not to perform any translations by checking the “**none**” box under “**Protein Translation Frames**.” Also tell Map+ to “**Save Map cut sites as features in map+.rsf**” by checking the box there. No “Options” are necessary; press “**Run**.”

The top output file will be the RSF feature annotation; “**Close**” it for now. The next file will show Map+’s restriction enzyme site locations; look through it and then “**Close**” it. Use the “**Output Manager**” to give the “**map+.out**” file a better name. Also select the “**map+.rsf**” file in the “**Output Manager**” and then “**Add to Editor**” and “**Overwrite old with new**” when prompted. This will add the newly discovered restriction cite annotation to any that the sequence may already have. “**Close**” the “**Output Manager**.” Now switch “**Display:**” to “**Graphic Features**” and check out all of the little blue arrows that indicate all of the restriction enzyme cut sites discovered by the program. Quickly double-click (or select and then use the “Windows” “Sequence Features” button) on one of the sites, and then select the entry in the new “**Sequence Features**” window. The name of the restriction enzyme that cuts at that particular cut site will be shown in the lower panel. “**Close**” the “**Sequence Features**” window and switch your “**Display:**” back to “**Residue Coloring**.” Close any output windows, “**Save**” the active RSF file in the Editor, and “**Exit**” SeqLab. Also log off the HPC and Classroom computers.

Homework assignment

Use the Lab 4 Web report form to tell me about your SeqMerge experience:

- 1) How many iterations through the system were necessary to make one contig, and

- 2) what were the final “**Assemble**” parameters required?
- 3) What was the percent identity reported in the “.pair” file from your Needle run?
- 4) How many restriction “**Enzymes that do not cut**” are listed at the bottom of your Map output?

Conclusion

Powerful utilities for building up DNA sequences from individual sequencing fragments and managing all the data are widely available. It's 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 these types of tools. Please check them out; learn the systems and spread the word! Next week — database similarity searching strategies — what do you have?

References

- Genetics Computer Group (GCG®), (Copyright 1982-2008) *Program Manual for the Wisconsin Package®*, version 11, <http://www.accelrys.com/> Accelrys Inc., San Diego, California, U.S.A.
- Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48**, 443–453.
- Staden, R. (1980) A new computer method for the storage and manipulation of DNA gel reading data. *Nucleic Acids Research* **8**, 3673–6694.
- Rice, P., Longden, I., and Bleasby, A. (2000) EMBOSS: The European Molecular Biology Open Software Suite. *Trends in Genetics* **16**, 276–277.
- Venter, J.C. et al. (2001) The sequence of the human genome. *Science* **291**, 1304–1351.