Babraham Bioinformatics

Sequence Assembly Using the Staden Package

A quick guide to the most commonly used parts of the Staden suite

Version 3.3 (public)

Licence

Introduction

A common task presented to many researchers is to take a series of short overlapping sequences, and assemble them into one large contiguous sequence. The starting sequences may be trace files from a sequencing project, or they may be short EST sequences from the public databases which you are trying to assemble into a full length cDNA, or even a mixture of the two. The underlying problem is the same.

The Staden package provides you with a series of tools to help you to perform tasks like this. It actually provides a whole lot of other sequence analysis tools as well, but we'll ignore them for the time being. Staden is free to academics, and runs under Windows, Unix, Linux and MacOSX.

Getting to Staden

Staden is available from http://staden.sourceforge.net/. There are versions for linux, OSX and windows.

In this course we will be using the Windows version of Staden but they're all pretty much the same.

Staden Overview

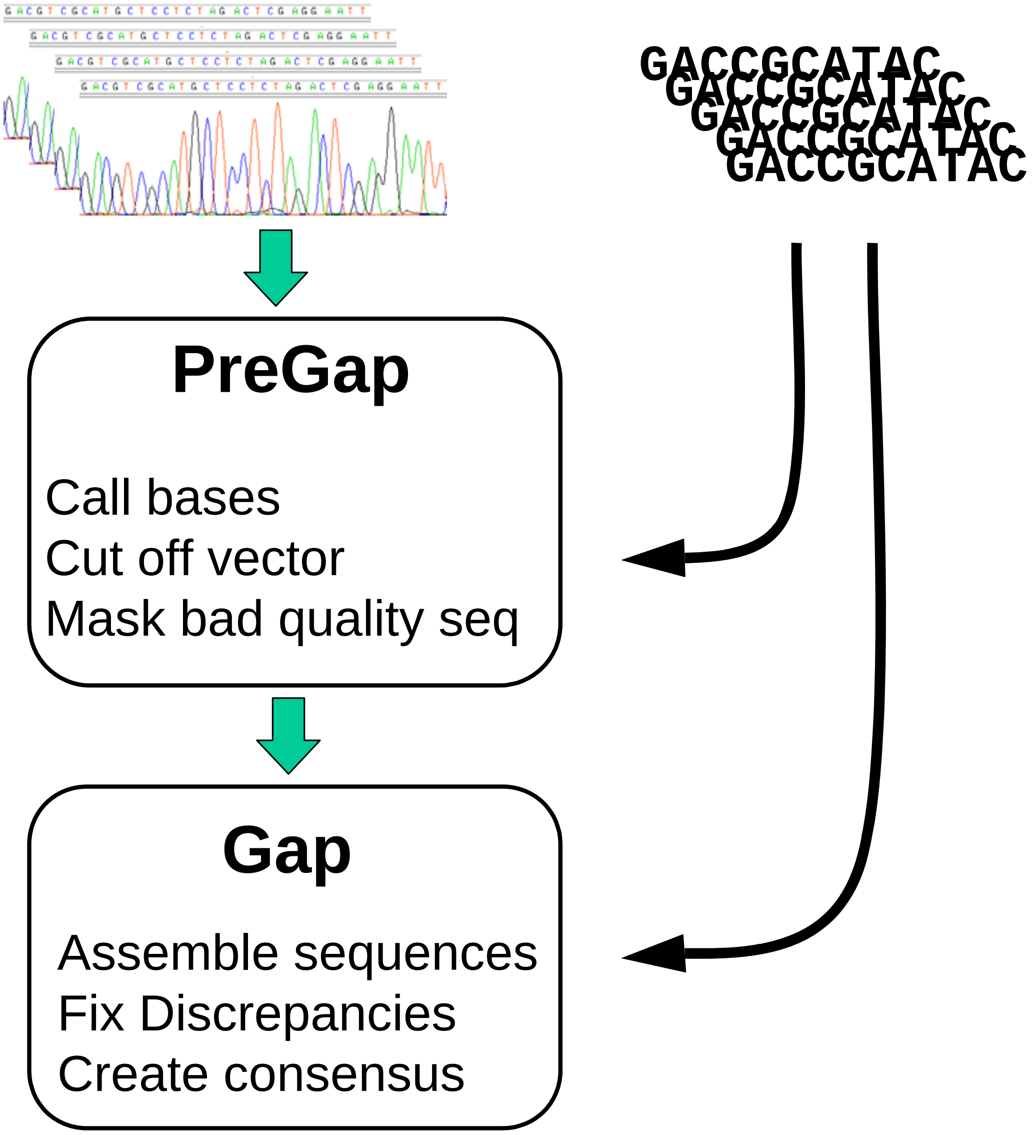
The diagram below shows the two main parts of the Staden program used for sequence assembly.

PreGap

Pregap is used to process raw traces, hot off the sequencer (although it can be used for text files as well). It is used to mask all of the sequence you probably don't want in the final assembly such as bits of vector and poor quality sequence.

Gap

Gap is the Genome Assembly Program – the bit which actually assembles your individual fragments into long contigs. It allows you to edit the assembly, referring back to the starting traces where they are present.



PreGap

Call bases
Cut off vector
Mask bad quality seq

Gap

Assemble sequences
Fix Discrepancies
Create consensus

Formatting Sequence files

In order to enter text sequence (not traces) into the Staden package it needs to be in either Staden or EMBL format. Staden format is simply raw sequence, no numbers, no headers nothing extra – just the sequence itself. EMBL format is the standard format used by the EMBL database, and is the default format exported by the Sequence Retrieval System (http://srs.ebi.ac.uk).

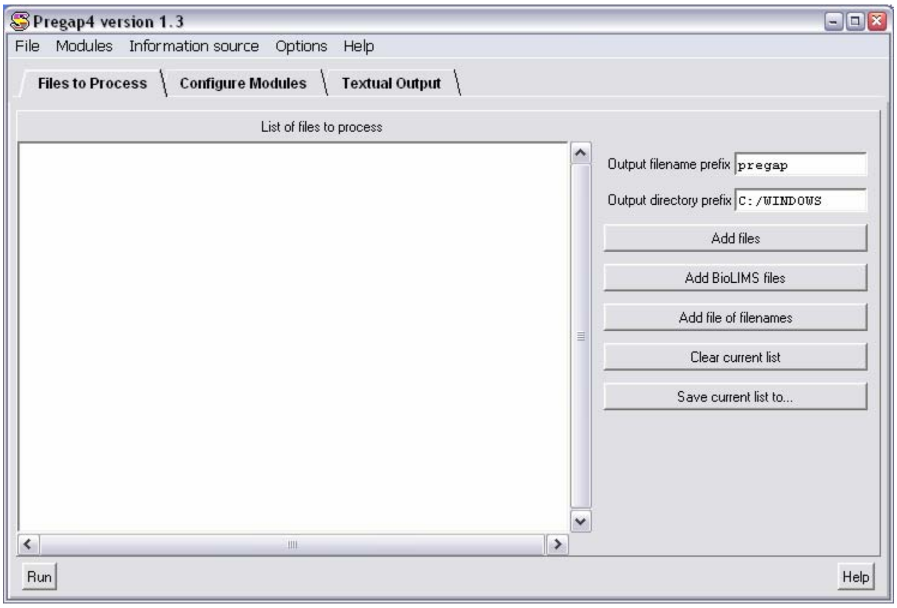
If you are using VectorNTI to manage your sequences you can export in EMBL format by selecting a sequence in the Explorer and selecting DNA/RNA > Export > Molecule into Text File.

PreGap

Pregap is the program you must run on all traces you wish to assemble. Optionally you can also run raw sequence through pregap. Pregap's job is to process your raw files so that they are in a form suitable for assembly. It does this by hiding certain parts of your sequence from the assembly program. These are usually parts which are poor quality, or the end of any vector you may have sequenced. Pregap will only mask these bits of sequence so they don't influence the assembly, it doesn't delete them. This means that later on, if you decide you'd really like to get that slightly dodgy bit of sequence back, then you can.

Starting PreGap

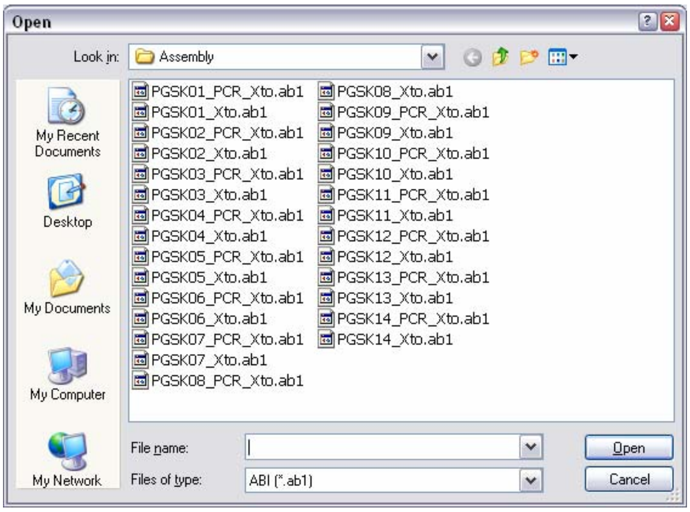
When you start pregap you will see a window like this appear. You will see that there are 3 tabs across the top "Files to Process", "Configure Modules" and "Textual Output". We will work our way along from left to right.



Output filename prefix: pregap

Output directory prefix: C:/WINDOWS

Files to Process



Look in: Assembly

PGSK01_PCR_Xto.ab1
PGSK01_Xto.ab1
PGSK02_PCR_Xto.ab1
PGSK02_Xto.ab1
PGSK03_PCR_Xto.ab1
PGSK03_Xto.ab1
PGSK04_PCR_Xto.ab1
PGSK04_Xto.ab1
PGSK05_PCR_Xto.ab1
PGSK05_Xto.ab1
PGSK06_PCR_Xto.ab1
PGSK06_Xto.ab1
PGSK07_PCR_Xto.ab1
PGSK07_Xto.ab1
PGSK08_PCR_Xto.ab1
PGSK08_Xto.ab1
PGSK09_PCR_Xto.ab1
PGSK09_Xto.ab1
PGSK10_PCR_Xto.ab1
PGSK10_Xto.ab1
PGSK11_PCR_Xto.ab1
PGSK11_Xto.ab1
PGSK12_PCR_Xto.ab1
PGSK12_Xto.ab1
PGSK13_PCR_Xto.ab1
PGSK13_Xto.ab1
PGSK14_PCR_Xto.ab1
PGSK14_Xto.ab1

File name:

Files of type: ABI (*.ab1)

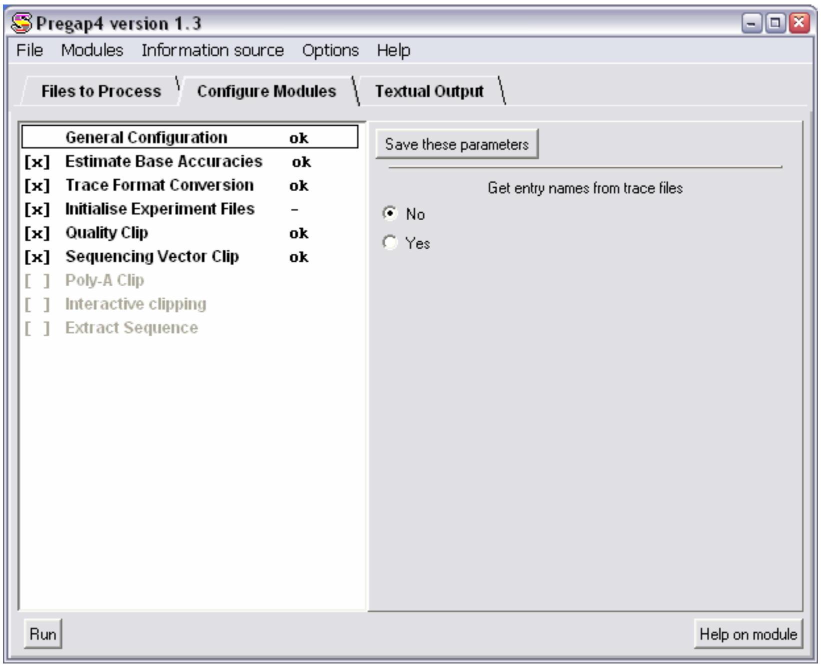
To select the files you wish to process, click on the "Add Files" button on the right on the first window you see. You should be presented with a file selector box. Move to the directory containing the files you wish to process and select them. You can select multiple files by holding down the control key whilst making your selection.

If you can't see your files in the directory you think they should be in, try changing the "Files of type" option to be "Any files".

Having selected your files, their names should appear in the list in the main PreGap window.

Configure Modules

The next step is to tell PreGap what processing you want performed on your files. You do this by selecting the “Configure Modules” tab at the top of the main window. You should see a screen which looks like this;



```
    General Configuration       ok
[x] Estimate Base Accuracies    ok
[x] Trace Format Conversion     ok
[x] Initialise Experiment Files -
[x] Quality Clip                ok
[x] Sequencing Vector Clip      ok
[ ] Poly-A Clip
[ ] Interactive clipping
[ ] Extract Sequence
```

Get entry names from trace files

● No
○ Yes

On the left hand side are a list of all of the modules which you can select, one of these will be highlighted by having a box around it. The highlighted module will display its options on the right hand half of the window.

You can get help on any of the modules listed by selecting the module from the list on the left, then pressing the “Help on Module” button on the bottom right.

On the left of each of the module names you will see square brackets which may have a cross in them. This shows whether this module will be applied to your sequences. Clicking on the box will turn the module on or off. Active modules will be written in black, inactive ones will be grey.

For most sequencing runs you won’t need to change anything in the pregap module setup, but it’s as well to know what it all does anyway.

Going through the modules in order;

General Configuration

You don’t have a choice about whether to use this or not, but you do have an option to set. You can tell the program whether to look inside your trace files to determine the name of each sample, or to just take the name from the file. The safe approach is to always take the name from the file, so you should usually say NO to these options.

Trace format conversion

Although pregap can read ABI format files it will not work with them directly because the ABI file format has not been officially laid out. If you want to work with ABI files you therefore need to convert them to a format which Staden is happy with. This module will do this for

you. Converting your modules to a published format (like ZTR or SCF) has the added advantage of making much smaller files. For the same sequencing run a ZTR file will be approximately 2% of the size of the corresponding ABI file. This can save you a lot of storage space in the long run. Once you've converted your ABI files you can then delete the originals as you won't need them any more.
If you're using ABI files as your starting point this module needs to be turned ON.

Estimate Base Accuracies

This module is required in order to hide bad quality sequence. There is a parameter which tells it how to score its confidence values. You don't need to worry about what this does, but it should be set to Logarithmic. This module should be turned ON.

Initialise Experiment Files

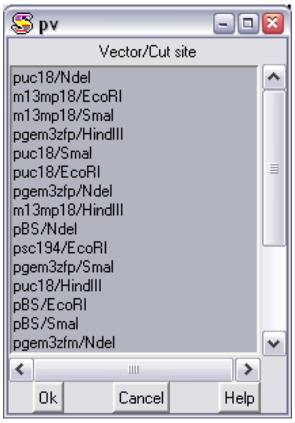
The experiment file is a file written by PreGap for each trace file which tells the assembly program which parts of the sequence have been masked. You need these files to record the output of the other modules. There are no parameters to set, but this option needs to be turned ON.

Quality Clip

The quality clip module will mask off poor quality sequence from the final alignment. You've probably seen that at the start and end of a sequencing chromatogram the quality of the sequencing can deteriorate, such that you can't be confident in the base calling. This module will hide this dodgy sequence away (but you can always get it back later if you're desperate!). There are two clip modes, "By base call" and "By confidence". You should select "By Confidence" as this is the more accurate mode. There are more parameters which you can change, but the defaults are nearly always OK.

Sequencing Vector Clip

If you are sequencing a construct out of a vector then you often find that you end up with a bit of the sequence of the vector stuck on the end of the sequence you really want. This module can remove the vector sequence for you. There are several ways in which this can be achieved, of which only the easiest is shown here. For details about the other ways of clipping off vector, see the bioinformatics department.



puc18/NdeI
m13mp18/EcoRI
m13mp18/SmaI
pgem3zfp/HindIII
puc18/SmaI
puc18/EcoRI
pgem3zfp/NdeI
m13mp18/HindIII
pBS/NdeI
psc194/EcoRI
pgem3zfp/SmaI
puc18/HindIII
pBS/EcoRI
pBS/SmaI
pgem3zfm/NdeI

The easiest way to clip off your vector is to use something called a vector-primer file. This is a pre-built database of vectors and cut sites that the program knows about. To use this select YES to "Use Vector-primer file", then press the button which says "Select Vector-primer subset".

You should see a list of vectors and cut sites. You should select the vector and site you are using from the list. If you are using a vector which isn't listed then you should contact the bioinformatics department who can add more items to this list.

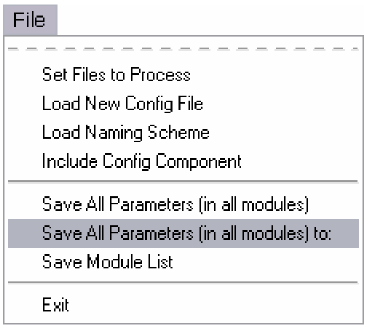
Interactive Clipping

This module allows you to preview each trace file you process and to manually alter the boundaries identified as being the good quality sequence which isn't vector. This module can be useful as a check if you are processing traces but don't intend assembling them, but there's no need to use it if you are going to put your sequences into Gap.

Extract Sequence

The final option is useful only if you want to process traces, but don't actually want to assemble them. If, for example, you were sequencing lots of different clones from a library, and you just wanted to know what each of them was, then you could use PreGap to remove bad quality sequence and any vector contamination. You could then use Extract Sequence to save the good portions of each sequence to a text file. You could then use this text file as input to a Blast search (or whatever). If you want to use this module, then you should specify that you want to output in Fasta format, and that you want to output in one file only. You also need to specify a file name. If you are running PreGap with the intention of assembling your sequences into a longer contig, then you should turn this option OFF.

Saving your options

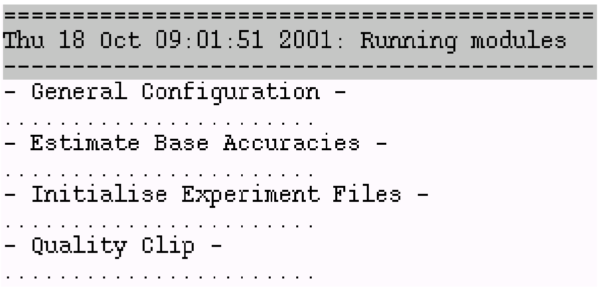


Having set up all of the options in all of the modules, a nice feature is the ability to save this configuration, so that the next time you have a batch of sequences to process you don't need to re-configure all of the modules, you can just load in your last configuration file. To do this, from the "File" menu on the main bar, select "Save all parameters in all modules to". This will produce a standard file selector box, and you can save your preferences to a file.

The next time you come to process a batch of sequences you can reload these preferences using the "Load New Config File" option in the same menu.

Running PreGap

Once you are happy that you've included all of the files you wish to process, and have set all of the options in the modules you wish to use, you can run PreGap by pressing the Run button at the bottom left of the main window. The display will switch to the "Textual Output" tab and the files will be processed.

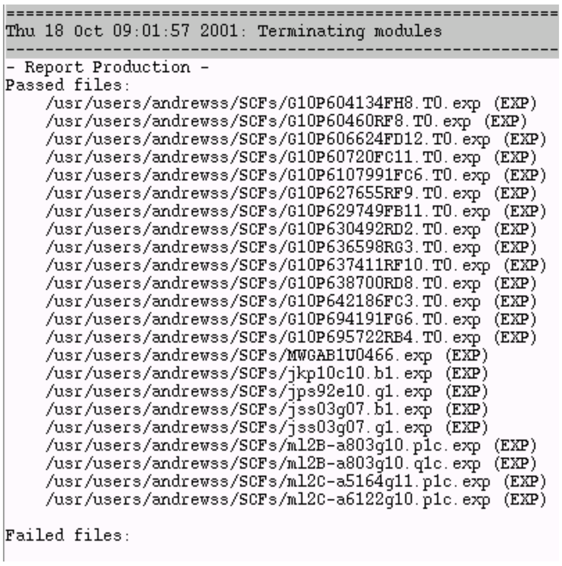


```
==========================================
Thu 18 Oct 09:01:51 2001: Running modules
------------------------------------------
- General Configuration -
.......................
- Estimate Base Accuracies -
.......................
- Initialise Experiment Files -
.......................
- Quality Clip -
.......................
```

You should see a separate text entry for each of the modules you have included and a series of dots will appear underneath each of them. Each dot represents one trace being processed. If you see an exclamation mark rather than a dot, that means that one of the traces has failed that module, and will be excluded from further analysis. This should be a rare occurrence and would usually only be caused by

1) The program not being able to find one of the traces you specified (maybe you deleted or moved it after loading the names into pregap?).

2) The whole of your sequence has been masked. The program may conclude that all of your sequence is either vector, or poor quality, in which case it won't pass it through to be assembled.



```
========================================================
Thu 18 Oct 09:01:57 2001: Terminating modules
--------------------------------------------------------
- Report Production -
Passed files:
    /usr/users/andrewss/SCFs/G10P604134FH8.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P60460RF8.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P606624FD12.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P60720FC11.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P6107991FC6.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P627655RF9.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P629749FB11.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P630492RD2.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P636598RG3.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P637411RF10.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P638700RD8.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P642186FC3.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P694191FG6.T0.exp (EXP)
    /usr/users/andrewss/SCFs/G10P695722RB4.T0.exp (EXP)
    /usr/users/andrewss/SCFs/MWGAB1U0466.exp (EXP)
    /usr/users/andrewss/SCFs/jkp10c10.b1.exp (EXP)
    /usr/users/andrewss/SCFs/jps92e10.g1.exp (EXP)
    /usr/users/andrewss/SCFs/jss03g07.b1.exp (EXP)
    /usr/users/andrewss/SCFs/jss03g07.g1.exp (EXP)
    /usr/users/andrewss/SCFs/ml2B-a803g10.p1c.exp (EXP)
    /usr/users/andrewss/SCFs/ml2B-a803g10.q1c.exp (EXP)
    /usr/users/andrewss/SCFs/ml2C-a5164g11.p1c.exp (EXP)
    /usr/users/andrewss/SCFs/ml2C-a6122g10.p1c.exp (EXP)

Failed files:
```

After the modules have run, the program will give you a list of all of the traces which passed all of the modules, and a list of those which failed.

In this example all of the files processed passed all of the modules. This is normally what you'd expect to see.

Finally, you will see a report for each sequence from each module. Usually you can skip through this as you should have seen the important information by this stage.

Finishing up

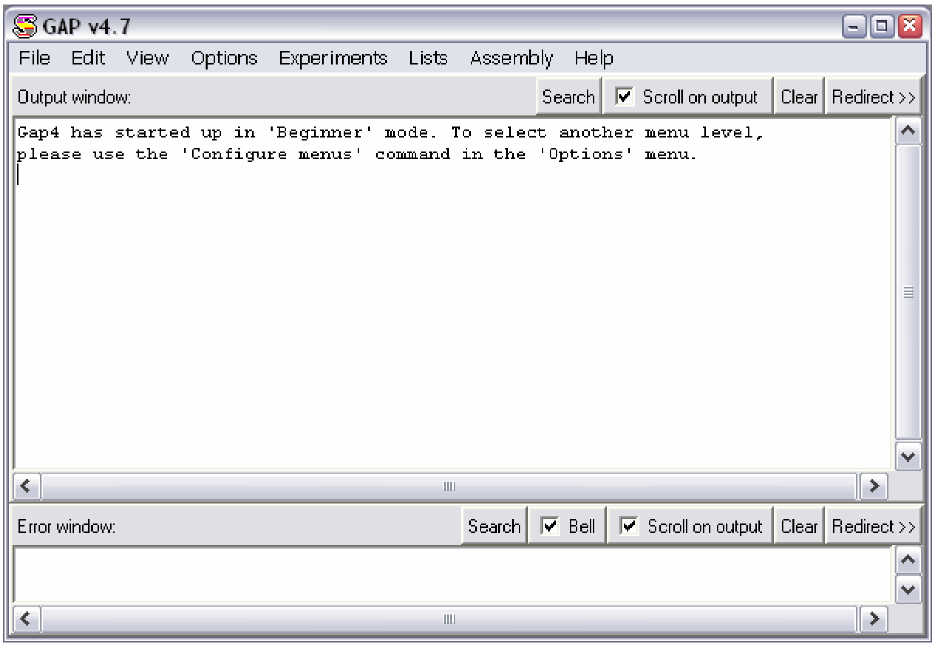
Once you’ve successfully processed your files you’re finished with PreGap, and can simply close it down with “File → Exit”.

At this stage you should also use your standard system file manager to delete your original ABI files from your working directory, as these are no longer required.

Gap

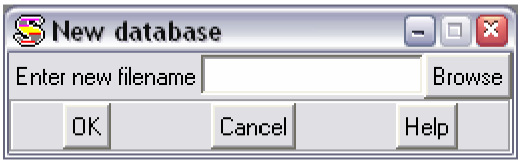
Once you have processed your raw traces through pregap, then you can assemble them into contiguous stretches using the Gap4 program.

When gap opens you should see the main window which looks like this:



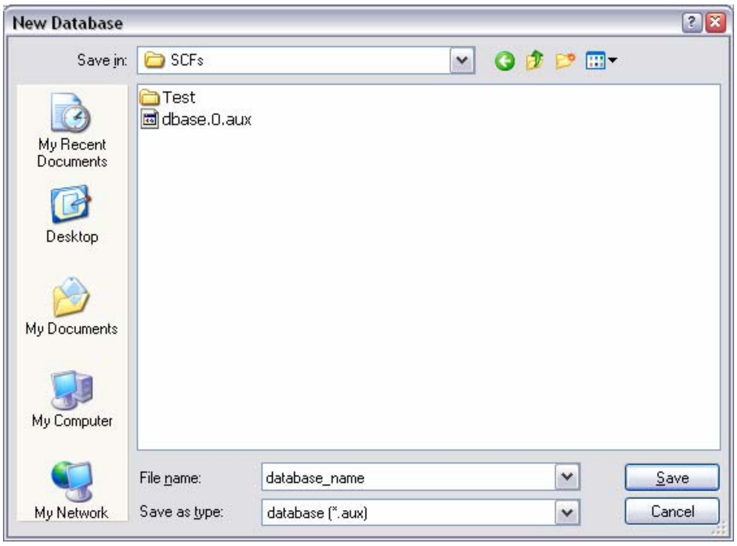
```
Gap4 has started up in 'Beginner' mode. To select another menu level,
please use the 'Configure menus' command in the 'Options' menu.
```

Starting a new assembly



The first thing you need to do after opening Gap is to create a new assembly into which you can enter the traces you have previously processed. To do this, select “File → New” from the Main window.

To specify where you would like your database to be created you should press the “browse” button.



Save in: SCFs

Test
dbase.0.aux

File name: database_name

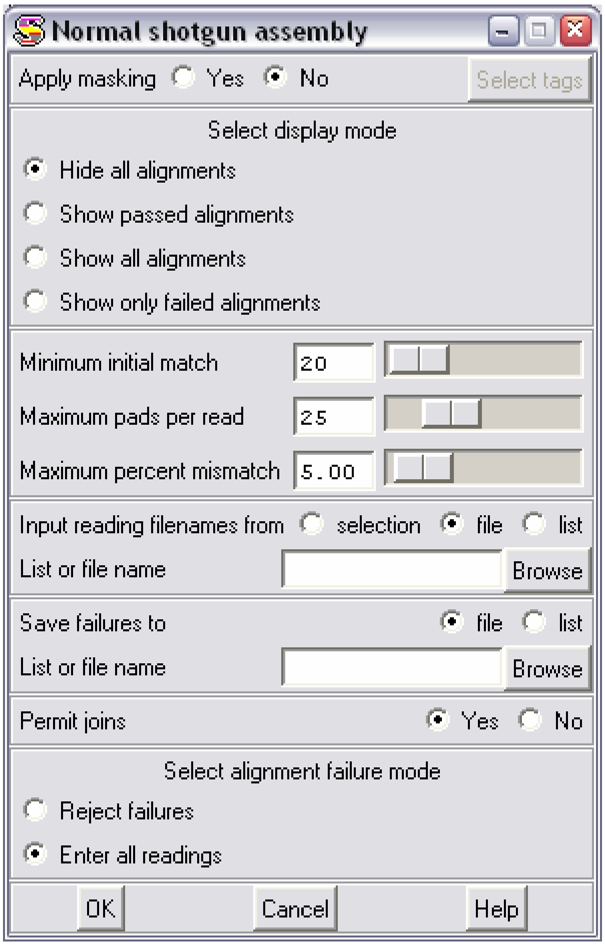
Save as type: database (*.aux)

You should then move to the directory where you just used pregap to process your input files. Once there you should enter a name for your new database in the “File name” box.

To go ahead and create the database press the Save button on the file selector, and then OK on the New database box. You should see a message in the main window confirming that you have indeed opened a database.

Adding sequences into the assembly

Now that we've got an empty database to work with, we need to add the sequences we previously processed into it. To do this we use the shotgun assembly tool. This is launched by selecting "Assembly → Normal Shotgun Assembly" from the main window. This should produce the following dialog box:



Normal shotgun assembly

Apply masking ○ Yes ● No

Select display mode

● Hide all alignments

○ Show passed alignments

○ Show all alignments

○ Show only failed alignments

Minimum initial match 20

Maximum pads per read 25

Maximum percent mismatch 5.00

Input reading filenames from ○ selection ● file ○ list

List or file name

Save failures to ● file ○ list

List or file name

Permit joins ● Yes ○ No

Select alignment failure mode

○ Reject failures

● Enter all readings

Alignment options

There are several options you can set when entering sequences into a database. These allow you to control how much of a match you require before the program is allowed to join one sequence to another. Whilst these settings can be important when you are working on a large sequencing project, you can nearly always leave them at their defaults with no ill effects. The default settings are fairly conservative, and may fail to make some joins which are present, but this can always be corrected later (as we will see).

Selecting files

In order to enter the sequences into the database we need a list of all of the files which came through pregap successfully. Fortunately, pregap creates a file called pregap.passed which contains the names of all of these files. To enter the sequences into Gap we can just point it at the pregap.passed file. To do this, under the "Input reading names from" section, make sure that "File" is selected, then press the Browse button, and select pregap.passed from the file list which appears.

Fails file

Whenever you enter sequences into Gap, it requires you to supply it with the name of a file into which it can save the names of any sequences which it cannot enter into the database. Usually, no sequences will fall into this category, but the program won't let you go on without entering a name. Type something like "fails" into the box, and that will keep it quiet!

Permit Joins

This option tells the program whether you want to allow it to join matching stretches together to make longer contigs. You probably do, so best to leave it set to Yes.

Failure Mode

The final option tells the program what to do with sequences it can't match to others in the database. You have the option of rejecting these (usually to allow you to adjust the alignment parameters and try again), but for modest numbers of sequences (up to 20) it's easier to just enter them all and sort things out later.

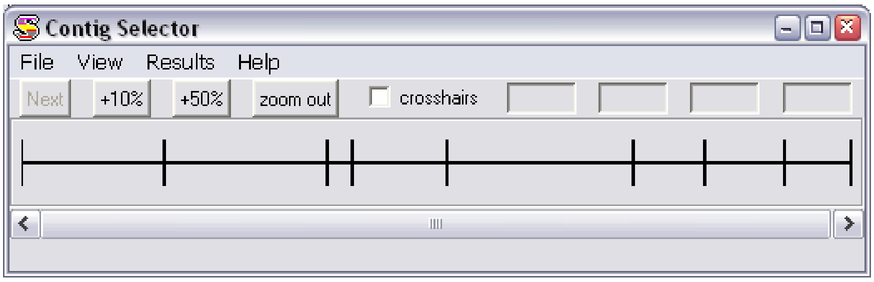
Once you've entered all the information, press the OK button to enter the sequences into the database.

You should see a load of text go flying by in the main window. The last bit should be a summary

```
Batch finished
      23 sequences processed
      23 sequences entered into database
       3 joins made
       1 joins failed
```

This shows you whether any of your sequences failed to be entered. You should also see another window called the "Contig Selector" appear.

The Contig Selector

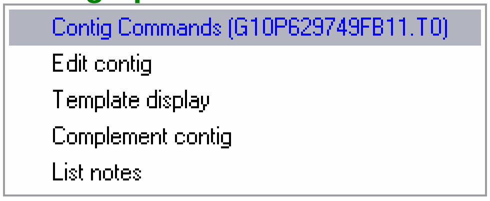


Once you have entered your sequences into the database you need to know how many contiguous stretches of DNA have been formed from them. This information is provided by the Contig Selector. If at any time you can't see this, it can always be retrieved by selecting "View → Contig Selector" from the main window.

The contig selector has a horizontal line across it, separated by a series of vertical lines. The space between adjacent vertical lines represents an assembled stretch of DNA. In the example above there are 8 contigs shown in the selector. For more information about the contig you can place your mouse over it and some information will be displayed at the bottom of the selector (length, and number of readings within the contig).

The order of the contigs in the selector is entirely arbitrary, just because contigs are adjacent in the selector doesn't imply that they are in any way related. You can even move the various contigs about by dragging them with the middle mouse button (left and right buttons together on a two button mouse).

Contig options



From within the contig selector you can perform several operations on your contig. To see the options you must place your mouse pointer over a contig, then **press and hold the right mouse button**.

List notes

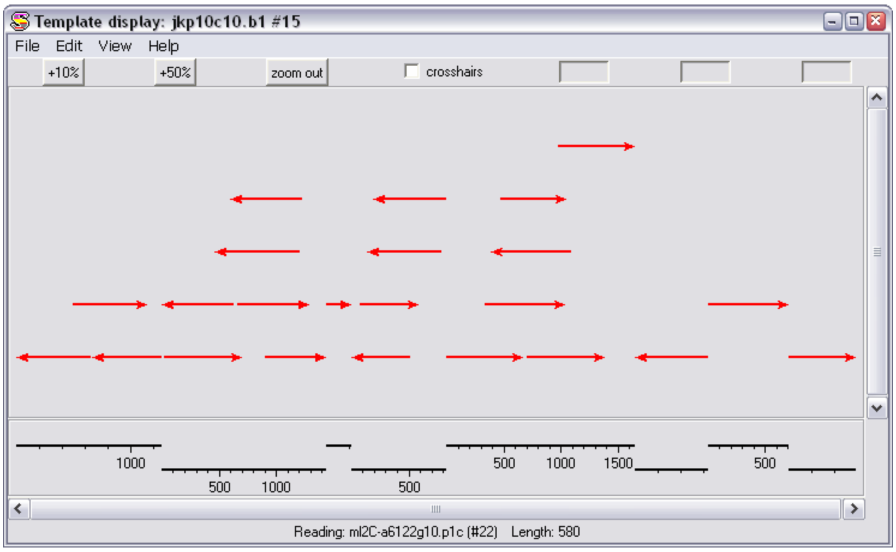
This option allows you to attach a series of text notes to your contig. Possibly useful as a reminder if you are likely to come back to your assembly at a later date.

Complement contig

Short contigs are very self-conscious, and are often bullied by longer contigs. The occasional complement will help to keep them happy. Also useful for switching the DNA strand shown when you view the contig (which is decided arbitrarily when you add in your sequences).

Template display

This option brings up a graphical representation of the positions and orientations of the readings within your contig. The part of the display you are interested in are the arrows (these are the readings). The display also initially shows dark blue extensions to your readings. These do not signify any useful information in this context, and should be removed by selecting "View → Templates". The template display view is interactive, and putting your mouse over a reading should identify it at the bottom of the display. As a side note, you can create a template display for all contigs by selecting “View → Templates” from the main window.



Reading: ml2C-a6122g10.p1c (#22) Length: 580

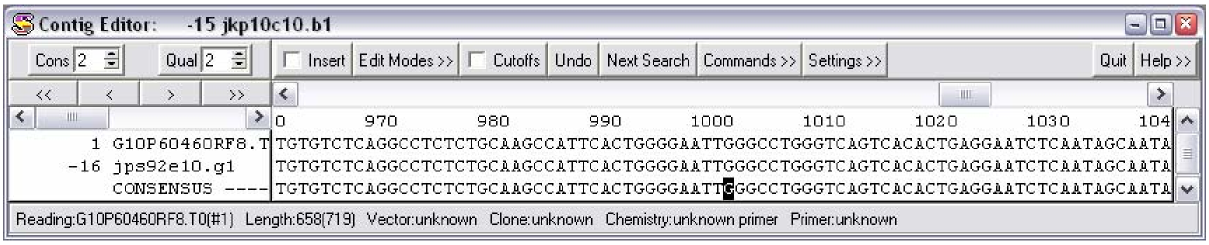
Edit Contig

This launches the contig editor, which is where the real work begins….

The Contig Editor

The contig editor works like an interactive multiple sequence alignment, containing all of the readings in each contig. It allows you to view the assembly of the readings, and look for differences between the different reads. The really clever bit is that the assembly is interactively linked to the sequencing chromatograms so that for any discrepancies you find, you can quickly refer back to the raw data to assess whether they result from a genuine sequence difference, or merely a miscalled base.

If you select contig editor from the contig selector options you should see something like this:



```
                      0       970       980       990      1000      1010      1020      1030      104
   1 G10P60460RF8.T  TGTGTCTCAGGCCTCTCTGCAAGCCATTCACTGGGGAATTGGGCCTGGGTCAGTCACACTGAGGAATCTCAATAGCAATA
 -16 jps92e10.g1     TGTGTCTCAGGCCTCTCTGCAAGCCATTCACTGGGGAATTGGGCCTGGGTCAGTCACACTGAGGAATCTCAATAGCAATA
     CONSENSUS ----  TGTGTCTCAGGCCTCTCTGCAAGCCATTCACTGGGGAATTGGGCCTGGGTCAGTCACACTGAGGAATCTCAATAGCAATA
```

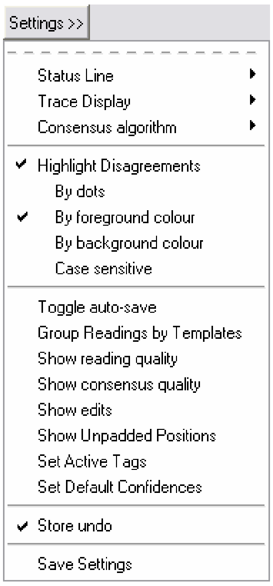
Reading:G10P60460RF8.T0(#1) Length:658(719) Vector:unknown Clone:unknown Chemistry:unknown primer Primer:unknown

The first thing to point out about the editor is that ***you should never try to manually adjust the size of the window***. The width of the window is fixed (though this isn’t a good thing and will hopefully be fixed in the next release). The height of the window is adjusted automatically. The editor will resize the height of the window to show you all the sequences at that point in the assembly. If you only see 2 sequences (as above), that’s because there are only 2 sequences there to see!

You can scroll through the contig using the scrollbar immediately below the top row of buttons. You will see the window resize as the number of sequences at each point in the assembly changes. There will always be at least two lines of sequence, as the very bottom line is a consensus, derived from the sequences above.

You should see that you have a series of different sequences visible, each of which has its own line. The sequence name is on the left hand side (eg G10P638700RD8). This will be the same as the name of the trace file you originally started with. To the left of the name is a number, this is just the number the sequence has been given within the assembly. More importantly, some of these numbers have a minus sign next to them. This indicates that this sequence has had to be reverse complemented to fit into the assembly. The program will have done this automatically, so you don't need to worry about the orientation of the sequences you enter.

Highlighting differences

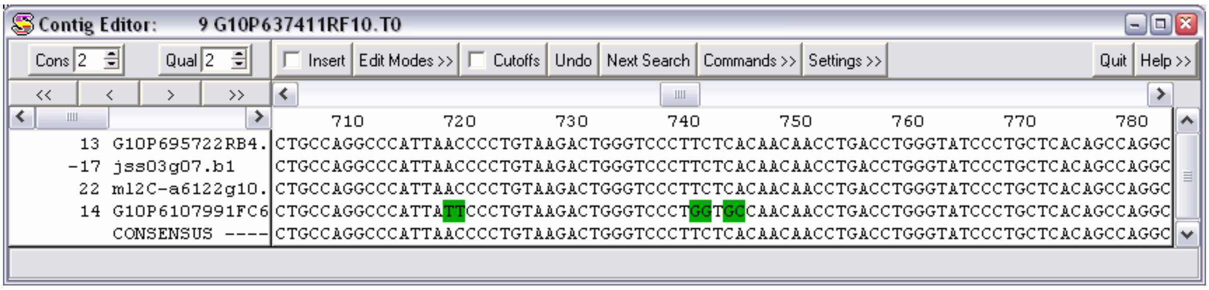


One useful feature offered by the editor is the ability to highlight any differences between the different sequences within the alignment. There are several different ways offered to do this, and they can be found by pressing the "Settings" button at the top of the editor.

You can highlight differences by dots (only showing differing bases, all others are replaced by a dot), or by changing the foreground or background colours of bases which do not match. I generally find that highlighting the background colour shows the differences up best.

The last option you may wish to change here is whether the program considers upper and lower case letters to be different. Unlike in other programs you don't need to use a case change to show where you have made changes (you can use the Show Edits function in the same menu to do this), so generally you want the disagreements to be case insensitive, so turn off the "Case sensitive" option.

You should now be able to see any differences clearly, as shown below.

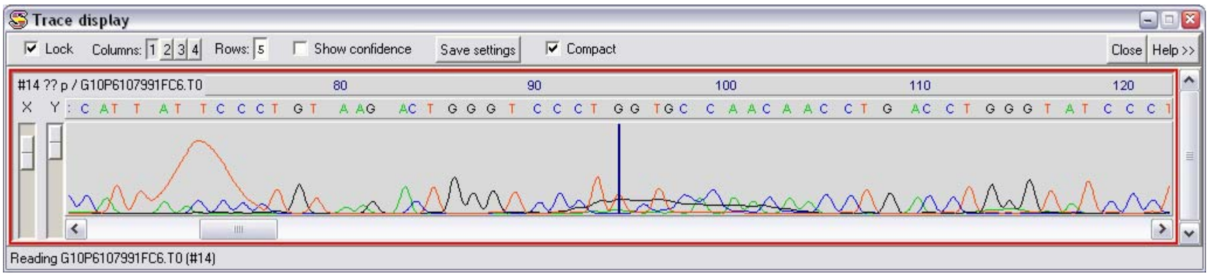


```
                            710       720       730       740       750       760       770       780
    13 G10P695722RB4.CTGCCAGGCCCATTAACCCCTGTAAGACTGGGTCCCTTCTCACAACAACCTGACCTGGGTATCCCTGCTCACAGCCAGGC
   -17 jss03g07.b1   CTGCCAGGCCCATTAACCCCTGTAAGACTGGGTCCCTTCTCACAACAACCTGACCTGGGTATCCCTGCTCACAGCCAGGC
    22 ml2C-a6122g10.CTGCCAGGCCCATTAACCCCTGTAAGACTGGGTCCCTTCTCACAACAACCTGACCTGGGTATCCCTGCTCACAGCCAGGC
    14 G10P6107991FC6CTGCCAGGCCCATTATTCCCTGTAAGACTGGGTCCCTGGTGCCAACAACCTGACCTGGGTATCCCTGCTCACAGCCAGGC
       CONSENSUS ----CTGCCAGGCCCATTAACCCCTGTAAGACTGGGTCCCTTCTCACAACAACCTGACCTGGGTATCCCTGCTCACAGCCAGGC
```

Once you have your view settings the way you like them you can preserve them by selecting the Save Settings option. This will restore your preferences whenever you open the contig editor in future.

Viewing the Chromatograms

When you see a difference between two sequences in an assembly, the next thing you would want to do is to refer back to the original sequencing chromatograms to see whether the error is genuine, or just a base calling problem. You can look at the chromatogram for a sequence by double clicking any letter in that sequence. The chromatogram will be opened, and will be centred on the base on which you double clicked. For example there is a disagreement at base 740 on the previous alignment. If I double click on the G in the bottom sequence I see this:



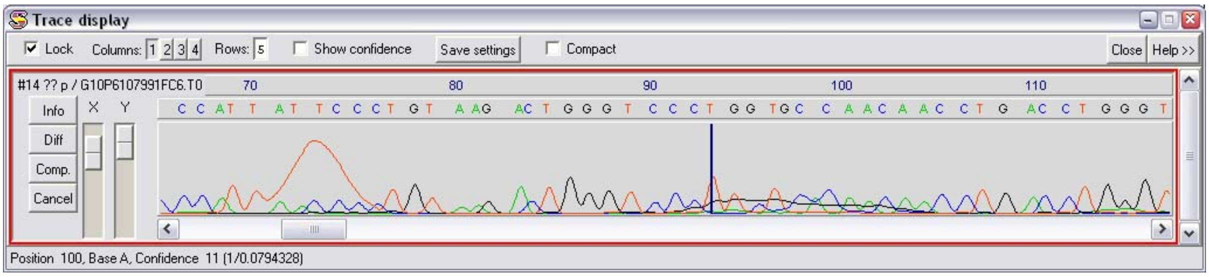
☑ Lock Columns: 1 2 3 4 Rows: 5 □ Show confidence ☑ Compact

#14 ?? p / G10P6107991FC6.T0

Reading G10P6107991FC6.T0 (#14)

The dark blue vertical line shows the base I was interested in. I can quickly see that this is a miscall, and the base should be a T (which agrees with all of the other sequences). If you leave this window open, and then click on a different base in the same sequence, you will see that the trace display automatically updates to show you this new position. Also, if you scroll along the trace display you will see that the cursor in the alignment also moves to keep the two displays aligned.

You can open muliple chromatograms by clicking on more sequences in the contig editor. You can close all chromatograms by pressing the “Close” button at the top right of the Trace display window. If you want to close an individual chromatogram you need the expanded chromatogram view, which you get by turning off the “Compact” option. You can then press “Cancel” to remove an individual trace.

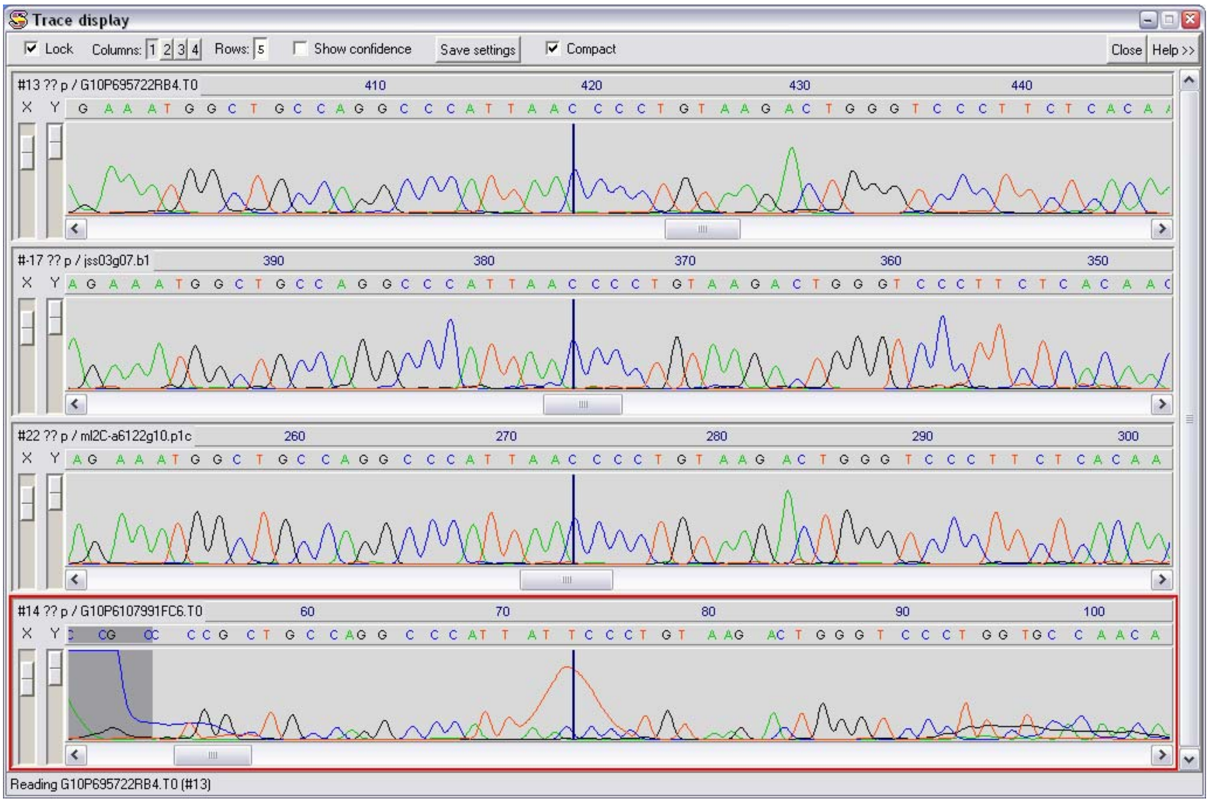


☑ Lock Columns: 1 2 3 4 Rows: 5 □ Show confidence □ Compact

#14 ?? p / G10P6107991FC6.T0

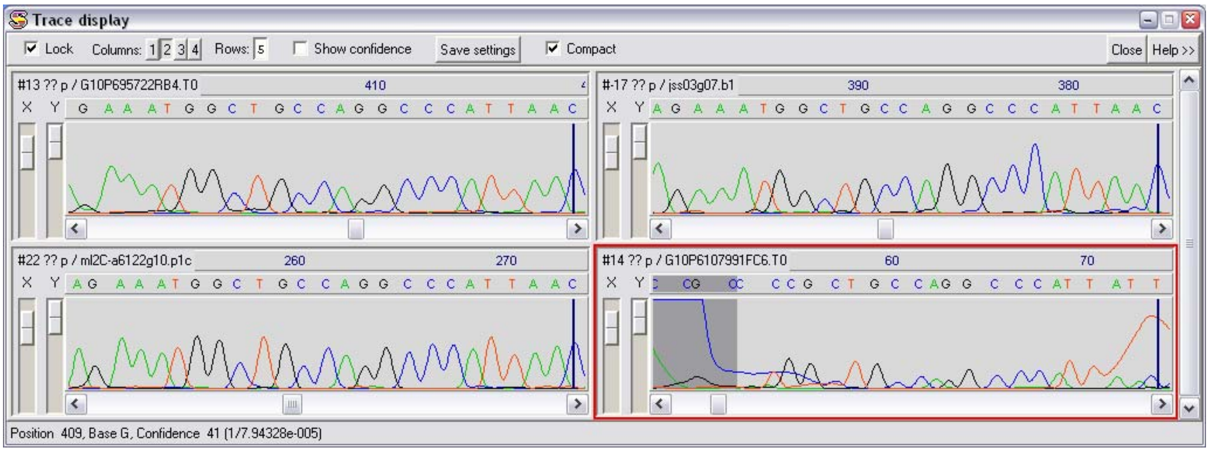
Position 100, Base A, Confidence 11 (1/0.0794328)

Another nice feature is the ability to quickly open all the traces available at a particular point in the assembly. To do this simply double click on a base in the consensus sequence. This will open all the traces at that point, and will close any old ones you may have had open. The traces will all be aligned where you clicked, and area all locked together, so any movement you make in one of them is reflected in all of them.


Trace display
☑ Lock Columns: 1 2 3 4 Rows: 5 □ Show confidence Save settings ☑ Compact Close Help >>
#13 ?? p / G10P695722RB4.T0
#-17 ?? p / jss03g07.b1
#22 ?? p / ml2C-a6122g10.p1c
#14 ?? p / G10P6107991FC6.T0
X Y

Reading G10P695722RB4.T0 (#13)

If you open a lot of traces, you may want to reduce the amount of space they take up on the screen. You can do this by arranging the traces in columns. You can select the number of columns by pressing the appropriate number button at the top of the trace display.

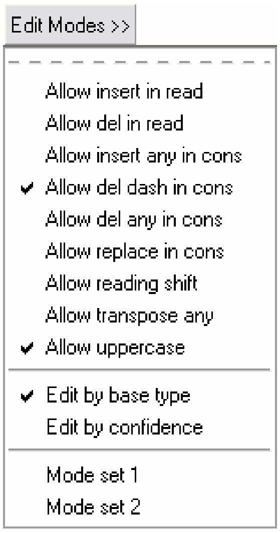

Trace display
☑ Lock Columns: 1 2 3 4 Rows: 5 □ Show confidence Save settings ☑ Compact Close Help >>
#13 ?? p / G10P695722RB4.T0
#-17 ?? p / jss03g07.b1
#22 ?? p / ml2C-a6122g10.p1c
#14 ?? p / G10P6107991FC6.T0
X Y

Position 409, Base G, Confidence 41 (1/7.94328e-005)

Editing the contig

Having looked through the discrepancies between the different readings in the assembly you will usually find some bases which have been miscalled, or parts of the contig which are misaligned. You will therefore want to make changes to the contig. There are a couple of different tools you need to know about in order to do this.

Edit Modes



The contig editor by default will not allow you to do whatever you want to your assembly. It operates a scheme whereby you can grant yourself permission to make ever more radical changes to your sequences. The permissions you have can be viewed and changed using the “Edit Modes” button at the top of the Contig Editor.

Within this menu, you can perform any action which has a red box next to it. You can make any box change state by clicking on it.

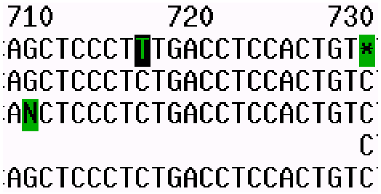
There are two predefined edit modes which should suffice for the majority of your editing. These are shown at the bottom. Mode 1 is for simple editing (adjusting the positions of gaps, calling uncalled bases and correcting disagreements). Mode 2 is for potentially more damaging actions (deleting or inserting bases etc).

Insert Mode

At the top of the contig editor is a box which says insert. This is activated when the box inside it is red. Under nearly all circumstances you should not use the insert mode when editing your contig. Inserting or deleting bases from individual reads can do really nasty things to the alignment of any downstream sequences. If you wish to insert or delete a base you should (nearly) always do the deletion/insertion in the consensus, which will preserve the relative alignment of all downstream sequences.

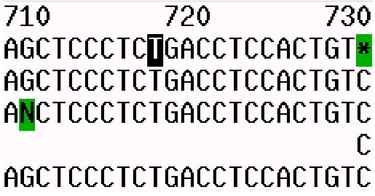
Re-calling bases

The most common type of edit is to change the call of a base.



```
710       720       730
AGCTCCCTTTGACCTCCACTGT*
AGCTCCCTCTGACCTCCACTGTC
ANCTCCCTCTGACCTCCACTGTC
                      C
AGCTCCCTCTGACCTCCACTGTC
```

To do this simply place the cursor on top of the base you wish to change,



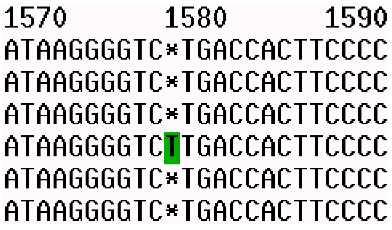
```
710       720       730
AGCTCCCTCTGACCTCCACTGT*
AGCTCCCTCTGACCTCCACTGTC
ANCTCCCTCTGACCTCCACTGTC
                      C
AGCTCCCTCTGACCTCCACTGTC
```

then type the letter you wish to change it to. You should see the letter change, the consensus update (if necessary), and the disagreement highlighting should be removed.

The same technique also works if you want to add in an extra letter where a gap character (*) has been inserted to preserve the alignment of the reads. Simply place the cursor on the star and press the base you would like inserted.

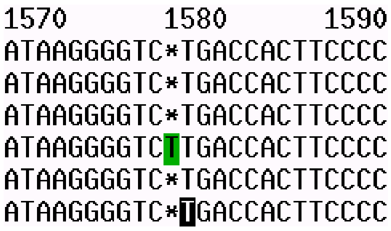
Removing an extra letter

The second common edit you will make is the removal of an extra base in a sequence. This results in a star being put in all but one of the readings in the contig, and a star appearing in the consensus.



```
1570      1580      1590
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTCTTGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
```

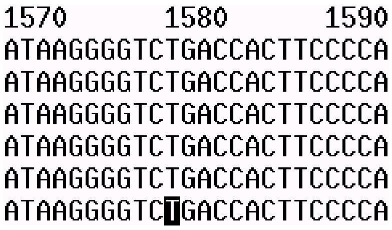
You should not correct this sort of mistake by individually deleting the stars and the extra T from all of the reads. This will break the alignment of sequences further downstream.



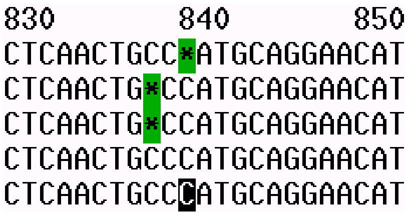
```
1570      1580      1590
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTCTTGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
ATAAGGGGTC*TGACCACTTCCCC
```

Instead you should place the cursor on the base after the star in the consensus (in this case a T), and then press backspace.

This will delete the previous base (or space) in all of the readings at that point, and will preserve the alignment. Note that if the consensus has been sufficiently biased that it displays the extra base instead of a star, you should still delete it this way, but you must use edit mode2, and you will also need to enable "Delete any in Cons".



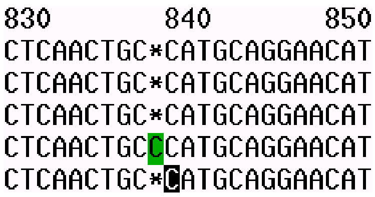
```
1570      1580      1590
ATAAGGGGTCTGACCACTTCCCCA
ATAAGGGGTCTGACCACTTCCCCA
ATAAGGGGTCTGACCACTTCCCCA
ATAAGGGGTCTGACCACTTCCCCA
ATAAGGGGTCTGACCACTTCCCCA
ATAAGGGGTCTGACCACTTCCCCA
```



```
830        840        850
CTCAACTGCC*ATGCAGGAACAT
CTCAACTG*CCATGCAGGAACAT
CTCAACTG*CCATGCAGGAACAT
CTCAACTGCCCATGCAGGAACAT
CTCAACTGCCCATGCAGGAACAT
```

As a final mention in this section, one extra complication you may see is a situation like this, where although two reads have the same sequence, they have been spaced differently to fit into the contig. There isn't a deletion you can make in the consensus which will fix all of these gaps in one go.

In cases like this where an inefficient alignment is responsible for your problem you can get the program to look through the alignment for a more sensible arrangement of the gap characters, so that they all line up. To do this, select "Commands → Shuffle Pads", and the program will try to sort this out for you like this.

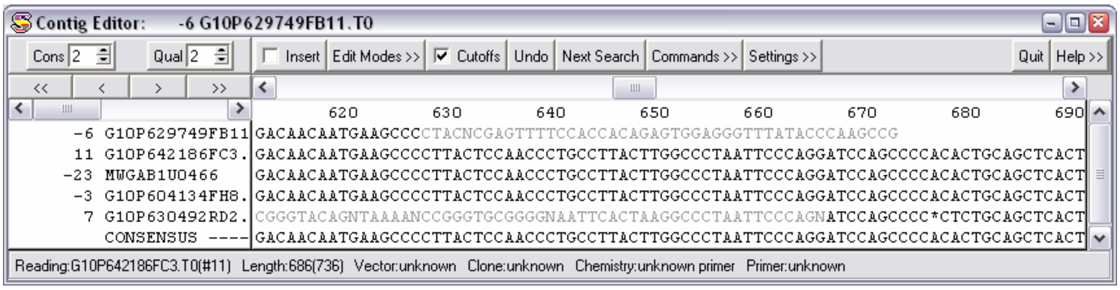


```
830        840        850
CTCAACTGC*CATGCAGGAACAT
CTCAACTGC*CATGCAGGAACAT
CTCAACTGC*CATGCAGGAACAT
CTCAACTGCCCATGCAGGAACAT
CTCAACTGC*CATGCAGGAACAT
```

Hidden Data

By default, when you are using the contig editor you are not seeing all of the sequence which was present in your original trace files. This is because when Pregap processed your files it decided that some of the data was of too poor quality to use, additionally it may also have decided that some of the data was vector sequence. This data was therefore masked, so that it would not interfere with the creation of your contigs. However, there are a few situations when you may wish to alter the boundaries of this masked data by hand from within the editor.

Firstly, to view the data which has been masked from your sequences simply press the "Cutoffs" button at the top of the Contig Editor. You should see the extra sequence reappear in light grey (much of it may also be highlighted in green if you have disagreement highlighting turned on).



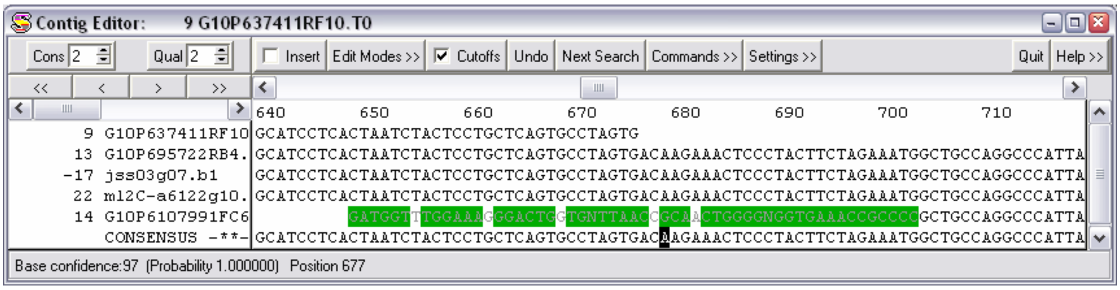
```
                           620       630       640       650       660       670       680       690
   -6 G10P629749FB11 GACAACAATGAAGCCCCTACNCGAGTTTTCCACCACAGAGTGGAGGGTTTATACCCAAGCCG
   11 G10P642186FC3. GACAACAATGAAGCCCCTTACTCCAACCCTGCCTTACTTGGCCCTAATTCCCAGGATCCAGCCCCACACTGCAGCTCACT
  -23 MWGAB1U0466    GACAACAATGAAGCCCCTTACTCCAACCCTGCCTTACTTGGCCCTAATTCCCAGGATCCAGCCCCACACTGCAGCTCACT
   -3 G10P604134FH8. GACAACAATGAAGCCCCTTACTCCAACCCTGCCTTACTTGGCCCTAATTCCCAGGATCCAGCCCCACACTGCAGCTCACT
    7 G10P630492RD2. CGGGTACAGNTAAAANCCGGGTGCGGGGNAATTCACTAAGGCCCTAATTCCCAGNATCCAGCCCC*CTCTGCAGCTCACT
      CONSENSUS ---- GACAACAATGAAGCCCCTTACTCCAACCCTGCCTTACTTGGCCCTAATTCCCAGGATCCAGCCCCACACTGCAGCTCACT
```

Reading:G10P642186FC3.T0(#11) Length:686(736) Vector:unknown Clone:unknown Chemistry:unknown primer Primer:unknown

Because this cutoff data was masked from the contig assembly it will not be properly aligned with the rest of the contig, so don't be concerned if the alignment appears incorrect.

Adding to the hidden data

It is relatively common for the very end of a read to be incorrect. This may just be because it is poor quality sequence, but is more often due to the vector clip in pregap failing to remove part of the surrounding vector.

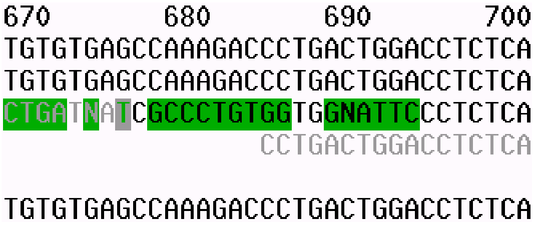


```
                     640       650       660       670       680       690       700       710
    9 G10P637411RF10 GCATCCTCACTAATCTACTCCTGCTCAGTGCCTAGTG
   13 G10P695722RB4. GCATCCTCACTAATCTACTCCTGCTCAGTGCCTAGTGACAAGAAACTCCCTACTTCTAGAAATGGCTGCCAGGCCCATTA
  -17 jss03g07.b1    GCATCCTCACTAATCTACTCCTGCTCAGTGCCTAGTGACAAGAAACTCCCTACTTCTAGAAATGGCTGCCAGGCCCATTA
   22 m12C-a6122g10. GCATCCTCACTAATCTACTCCTGCTCAGTGCCTAGTGACAAGAAACTCCCTACTTCTAGAAATGGCTGCCAGGCCCATTA
   14 G10P6107991FC6          GATGGTTTGGAAAGGGACTGGTGNTTAACCGCAACTGGGGNGGTGAAACCGCCCCGCTGCCAGGCCCATTA
      CONSENSUS -**- GCATCCTCACTAATCTACTCCTGCTCAGTGCCTAGTGACAAGAAACTCCCTACTTCTAGAAATGGCTGCCAGGCCCATTA
```

Base confidence:97 (Probability 1.000000) Position 677

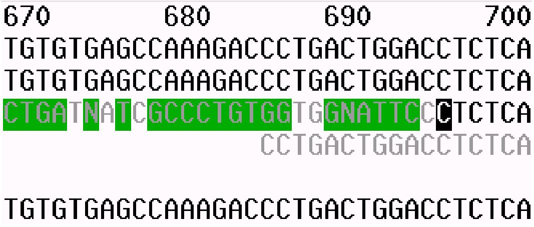
In this example you can see that the G10P6107991FC6 sequence suddenly becomes incorrect near its end.

In cases such as this you could go through and re-call every base at the end, but it is easier and more correct to move the boundary of the hidden data so that the bad data at the end of the read becomes masked.



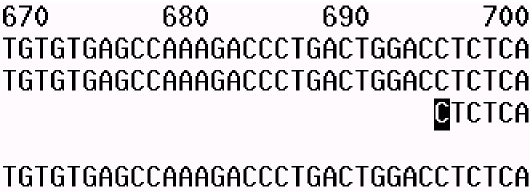
```
670       680       690       700
TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
CTGATNATCGCCCTGTGGTGGNATTCCCTCTCA
                CCTGACTGGACCTCTCA

TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
```

To do this you first need to make the hidden data visible by pressing the “Cutoffs” button. You should then position the cursor at the first hidden base after the end of the read you want to mask.



```
670       680       690       700
TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
CTGATNATCGCCCTGTGGTGGNATTCCCTCTCA
                CCTGACTGGACCTCTCA

TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
```

You can then make more data hidden by holding down the control key and using the left and right arrow keys to move the cursor to where you want the new end of the hidden data to be.



```
670       680       690       700
TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
                           CTCTCA

TGTGTGAGCCAAAGACCCTGACTGGACCTCTCA
```

Having moved the cutoff position you can re-hide the hidden data by pressing the “Cutoffs” button again.

Adding back hidden Data

Sometimes you will want to do the reverse of the above process. You will often find that sequence which has been masked as poor quality can be carefully read by eye, and used to extend the end of a contig. Those few extra bases may be of vital importance to you, so you should know how to resurrect that hidden data.

The process of unhiding hidden data is very similar to that shown above. First show the hidden data. Next, place the cursor on the last base of ***good*** data, then use control and the cursor keys to extend the good data into the previously hidden data.

Making mistakes

Since we all make mistaks at some point it is useful to have a way of correcting them. You will soon find that a few accidental key presses can have the disagreements in your contig lighting up along most of its length. Fortunately the program is in a position to help you. At the top of the Contig Editor is an Undo button. This not only allows you to undo your last action, but can be used repeatedly to undo all the changes you made since you started editing the contig. At some point you will be very glad the program has this feature!

Saving your changes

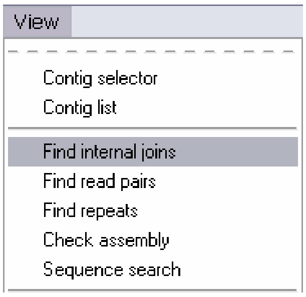
When you have finished editing your contig you will want to commit those changes back to your assembly database. To do this simply press the Quit button at the top right of the Contig Editor, and you will be asked whether you want to save the changes you have made. If you say yes then your database will be updated, if you say no then you are back where you started at the contig selector.

Joining Contigs Together

Quite often, when you assemble your sequences you will find that you are left with more than one contig. This may be because all of your sequences do not overlap, or it may be because there are some errors in your sequences and the program is not confident enough to join them together automatically.

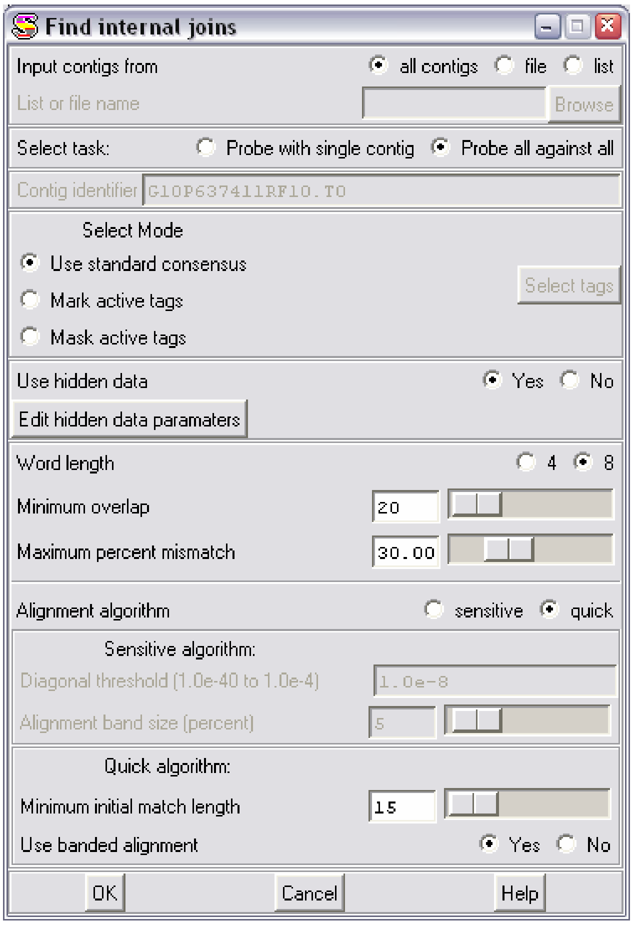
If there are more joins between contigs which can be made, then you can perform this operation manually. There are two step to doing this, finding potential joins, and using the join editor to make the connection between separate contigs.

Finding Joins



The first step in joining two contigs is to get the program to search for potential overlap sites between contigs. This is done by using a tool called “Find Internal Joins”. This tool can be started from the “View” menu of the main window.

Once you start the tool you should be presented with a list of options which looks like this.



Input contigs from ● all contigs ○ file ○ list

List or file name

Select task: ○ Probe with single contig ● Probe all against all

Contig identifier G10P637411RF10.T0

Select Mode

● Use standard consensus

○ Mark active tags

○ Mask active tags

Use hidden data ● Yes ○ No

Word length ○ 4 ● 8

Minimum overlap 20

Maximum percent mismatch 30.00

Alignment algorithm ○ sensitive ● quick

Sensitive algorithm:

Diagonal threshold (1.0e-40 to 1.0e-4) 1.0e-8

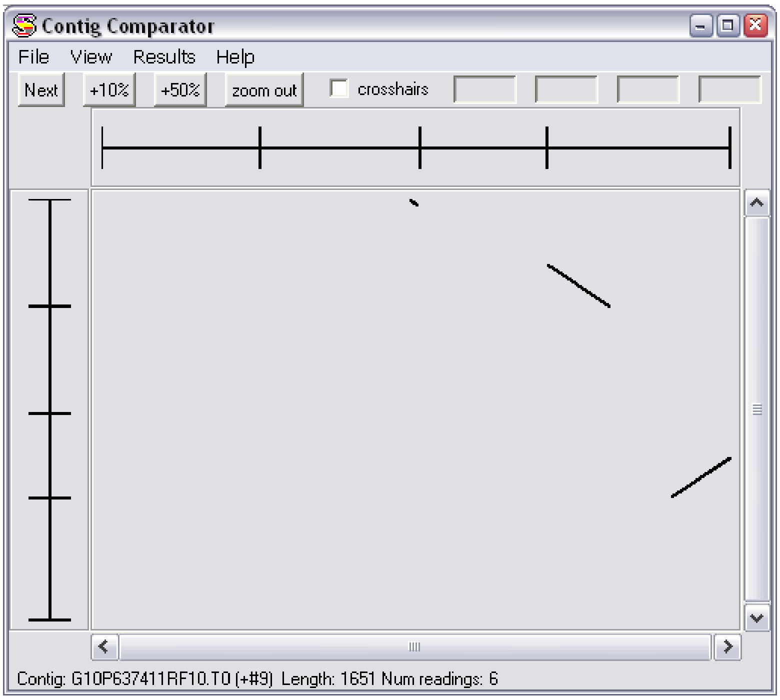
Alignment band size (percent) 5

Quick algorithm:

Minimum initial match length 15

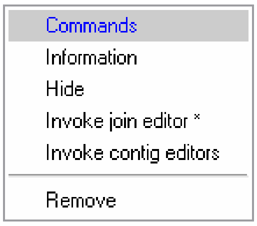
Use banded alignment ● Yes ○ No

There are several options here, but you will nearly always leave everything at its default value. The only option you may wish to change, is that if you are looking for very small overlaps, potentially with mistakes in them then you may wish to try using the Sensitive alignment algorithm. For most cases the quick algorithm should be OK though. Press OK to start the search, and be prepared to wait for a few seconds for the output to appear.



Contig: G10P637411RF10.T0 (+#9) Length: 1651 Num readings: 6

The window which appears consists of two contig selectors set at right angles to each other. In the middle are a series of black lines which show potential regions of overlap between the contigs. If you hold your mouse over one of the black lines then information about the join will be displayed at the bottom of the window.

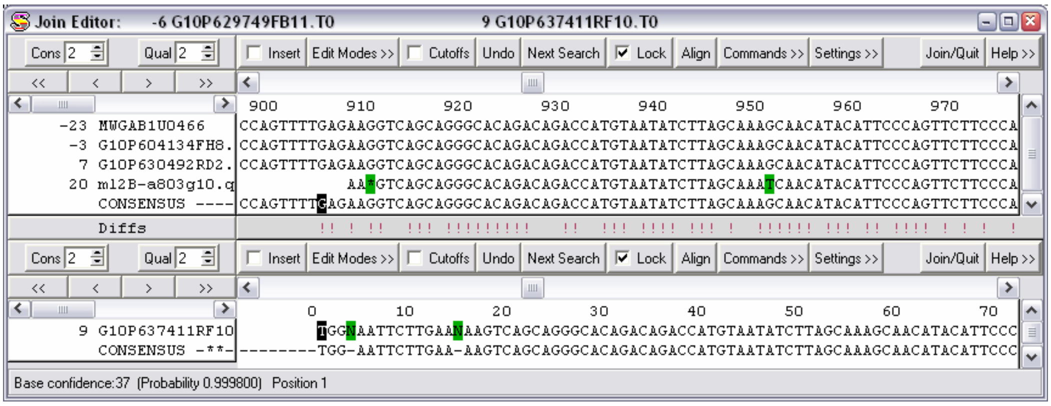


To view one of these joins in more detail you have two choices. You can press the right mouse button whilst you are over one of the joins. This will produce a small menu, which contains the option to “Invoke join editor”.

Alternatively you can use the “Next” button to step through all the potential joins, starting with the most solid one, until there are none left. Unless you have a good reason not to, then this is the approach we suggest you take.

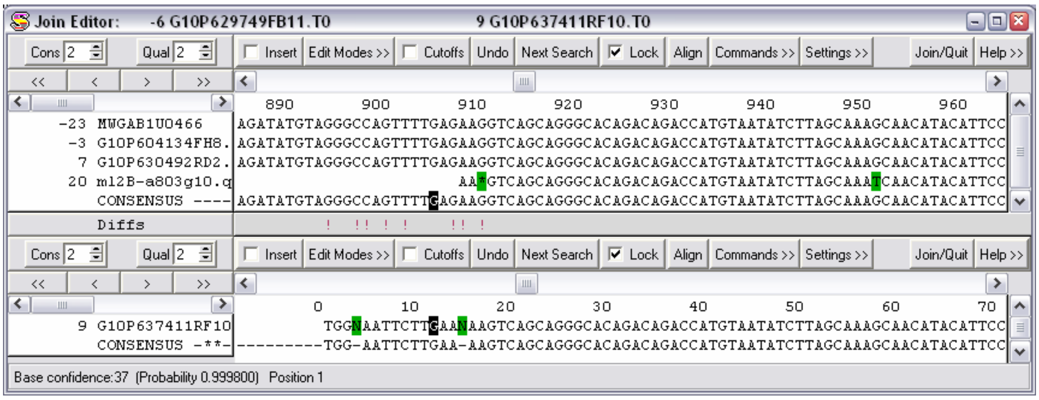
The Join Editor

When you invoke the join editor you should see a window appear which consists of two Contig Editors placed over the top of each other.



```
           900       910       920       930       940       950       960       970
 -23 MWGAB1U0466   CCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCCCAGTTCTTCCCA
  -3 G10P604134FH8.CCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCCCAGTTCTTCCCA
   7 G10P630492RD2.CCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCCCAGTTCTTCCCA
  20 ml2B-a803g10.q          AA*GTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAATCAACATACATTCCCAGTTCTTCCCA
     CONSENSUS ----CCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCCCAGTTCTTCCCA
     Diffs                 !! ! !!  !!! !!!!!!!!!   !!   !!! !!!! !!! !  !!!!!! !!! !! !!!! ! ! !  !

           0         10        20        30        40        50        60        70
   9 G10P637411RF10        TGGNAATTCTTGAANAAGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCCC
     CONSENSUS -**---------TGG-AATTCTTGAA-AAGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCCC
Base confidence:37 (Probability 0.999800) Position 1
```

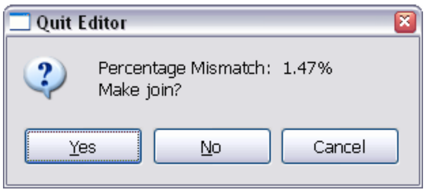
The two contigs should be roughly aligned so that their portion of overlap is displayed. You will notice that between the two editors is a line which says Diffs. On this line are a series of exclamation marks. Each of these represents a difference between the two contigs. When the join editor is first opened the contigs will be very poorly aligned, so the first thing you should do is to align them properly. To do this press the “Align” button in either of the contig editors. You should see the alignment improve, and many of the exclamation marks disappear.



```
             890       900       910       920       930       940       950       960
 -23 MWGAB1U0466   AGATATGTAGGGCCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCC
  -3 G10P604134FH8.AGATATGTAGGGCCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCC
   7 G10P630492RD2.AGATATGTAGGGCCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCC
  20 ml2B-a803g10.q                       AA*GTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAATCAACATACATTCC
     CONSENSUS ----AGATATGTAGGGCCAGTTTTGAGAAGGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCC
     Diffs                   !   !! ! !     !! !

               0         10        20        30        40        50        60        70
   9 G10P637411RF10         TGGNAATTCTTGAANAAGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCC
     CONSENSUS -**----------TGG-AATTCTTGAA-AAGTCAGCAGGGCACAGACAGACCATGTAATATCTTAGCAAAGCAACATACATTCC
Base confidence:37 (Probability 0.999800) Position 1
```

The tools available to you are now exactly the same as those you had in the Contig Editor previously. In this case though you are not really interested in fixing mistakes within a contig (although you can do this if you like), but looking at the differences between the contigs. You don’t need to change things so that both contigs are exactly the same where they overlap, they just need to be similar enough that you can convince yourself that they should overlap.

You should note that the join editor will also look into your hidden data to find potenital overlaps. If you need to use hidden data to make an overlap then you should unhide it before trying to make a join (as explained before). If you don’t then the program will tell you that the contigs do not overlap when you try to join them.



Percentage Mismatch: 1.47%
Make join?

When you are happy that the contigs should be joined together, you can perform the join by pressing the “Join/Quit” button. You will see a box appear which shows you the percentage mismatch, and asks if you want to make a join. Saying Yes will merge the two contigs to form one larger one. Saying No will take you back to the contig selector.

You can continue joining contigs until no more potential joins are found.

Adding more data

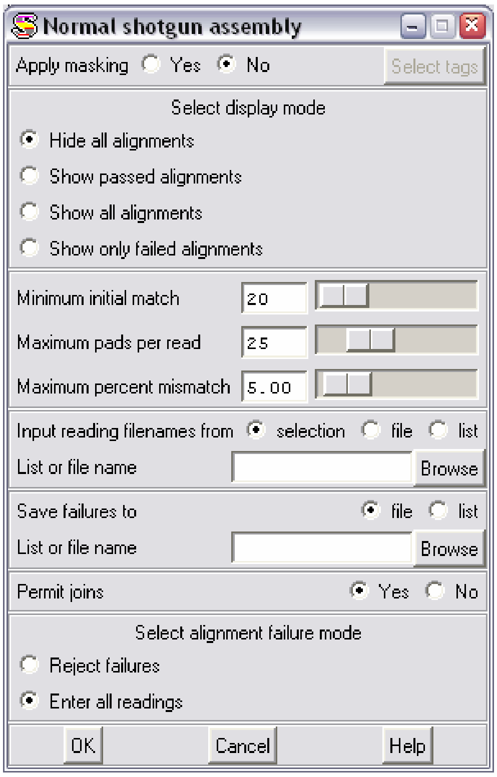
Traces

Despite your best efforts at joining your contigs, there will often come a point when your only recourse is to go out and get some more sequence to add into your database. Adding new traces into the database works exactly the same way as adding the one you used to start the database. The new sequences must be passed through Pregap as before, and then added using the “Assembly → Normal Shotgun assembly” tool and pointing it at the pregap.passed file. The only difference to the process previously described for creating a new database is that instead of creating a new database with “File → New”, you just open an existing one with “File → Open”.

Raw Sequence

All of the data we’ve added into our assembly so far has been derived from traces which have passed through Pregap. Staden however can also use raw sequence as input into its database. It can also import sequences individually, rather than using something like the pregap.passed list file we saw previously.

Any sequence being brought into Staden must be in staden format. It’s a bit of a cheek for Staden to claim this as its own format, since all it is is completely raw sequence. No header, no numbers, no spaces – just sequence. If you’re interested, you can convert any GCG format file to staden format using the “tostaden” command. You can also easily create a staden format file from a fasta file by simply deleting the first line (the one that begins with >).


Normal shotgun assembly

Apply masking ○ Yes ● No

Select display mode

● Hide all alignments
○ Show passed alignments
○ Show all alignments
○ Show only failed alignments

Minimum initial match 20
Maximum pads per read 25
Maximum percent mismatch 5.00

Input reading filenames from ● selection ○ file ○ list
List or file name

Save failures to ● file ○ list
List or file name

Permit joins ● Yes ○ No

Select alignment failure mode

○ Reject failures
● Enter all readings

To add in a single sequence (whether raw sequence or processed trace file), you again use the “Assembly → Normal Shotgun Assembly” tool from the main Gap window.

To input a single sequence, change the “input reading names from” option to “selection”, then use the browse button to select your raw sequence file. If you are importing a single processed trace then you need to point to the .exp file which was created by Pregap, and not to your original chromatogram file.

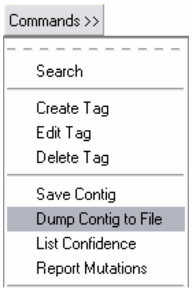
You will again need to specify a name for a fails file, then just press OK and your raw sequence should be imported. The program will add it to an existing contig if possible, and will join contigs together if the new sequence spans two or more existing contigs.

The sequence file will appear in the contig editor the same as any other sequence, the only difference will be that it will not open up a trace file when you double click on it, because there is no trace file associated with it.

Exporting information

When you’ve finished assembling your sequences you will want to be able to export information out of the database so that you have something to show for all your hard work. There are two different types of export you may be interested in, a dump of the whole contig assembly – showing all the reads, and an export of just the consensus, which you may want to use for further analysis.

Dump a whole contig



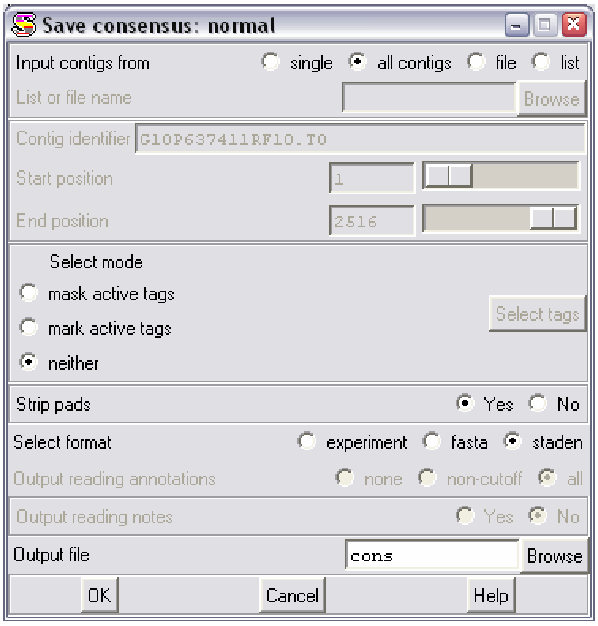
The tool which allows you to output a whole contig to a text file is an option in the “Commands” menu of the contig editor. This has an option “Dump Contig to File” .

This brings up a dialog box which allows you to set some basic parameters about the width of lines you want in your output, and lets you specify a filename. The file created will be a plain text file and will look something like this.

```
                                   10        20        30        40        50
 24 rawseq.txt             AGTCCCTGCGAAGCGCCCCCGCTGGCAGCCAGATTTGCAGAAGGTCGTCC
 -4 G10P60720FC11.T0            CTGCGAAGCGCCCCCGCTGGCAGCCAGATTTGCAGAAGGTCGTCC
    CONSENSUS              AGTCCCTGCGAAGCGCCCCCGCTGGCAGCCAGATTTGCAGAAGGTCGTCC

                                   60        70        80        90       100
 24 rawseq.txt             GCGGCTGCAGCGGGCAGGGGGATGCGAAACCTCCGACGCCAGAGGGTCGG
 -4 G10P60720FC11.T0       GCGGCTGCAGCGGGCAGGGGGATGCGAAACCTCCGACGCCAGAGGGTCGG
 20 ml2B-a803g10.p1c               AGCGGGCAGGGGGATGCGATACCTCCGACGCCAGAGGGTCGG
    CONSENSUS              GCGGCTGCAGCGGGCAGGGGGATGCGAAACCTCCGACGCCAGAGGGTCGG

                                  110       120       130       140       150
 24 rawseq.txt             CTCCGGCCTGCGGTAGGCTCCTTATCTGGAGAGGAAGCTATAGGACCCCC
 -4 G10P60720FC11.T0       CTCCGGCCTGCGGTAGGCTCCTTATCTGGAGAGGAAGCTATAGGACCCCC
 20 ml2B-a803g10.p1c       CTCCGGCCTGCGGTAGGCTCCTTATCTGGAGAGGAAGCTATAGGACCCCC
-13 G10P694191FG6.T0                                                  GACCCCC
    CONSENSUS              CTCCGGCCTGCGGTAGGCTCCTTATCTGGAGAGGAAGCTATAGGACCCCC

                                  160       170       180       190       200
 24 rawseq.txt             CCCC**ACTCTTCGACTCGCATGGCTGTCCCGTCACCCCCACTTCATGGT
 -4 G10P60720FC11.T0       CCCCA**CTCTTCGACTCGCATGGCTGTCCCGTCACCCCCACTTCATGGT
 20 ml2B-a803g10.p1c       CCCC**ACTCTTCGACTCGCATGGCTGTCCCGTCACCCCCACTTCATGGT
-13 G10P694191FG6.T0       CCCCCCCCTCTTCGACTCGCATGGCTGTCCCGTCACCCCCACTTCATGGT
    CONSENSUS              CCCC**ACTCTTCGACTCGCATGGCTGTCCCGTCACCCCCACTTCATGGT
```

Save a consensus sequence

The other export option you have is to extract the consensus from one or all contigs, and to save that to a text file. The option to do this can be found from the main Gap window. You should select “File → Save Consensus → Normal”.



Save consensus: normal

Input contigs from ○ single ● all contigs ○ file ○ list

List or file name Browse

Contig identifier G10P637411RF10.T0

Start position 1

End position 2516

Select mode

○ mask active tags

○ mark active tags

● neither

Select tags

Strip pads ● Yes ○ No

Select format ○ experiment ○ fasta ● staden

Output reading annotations ○ none ○ non-cutoff ● all

Output reading notes ○ Yes ● No

Output file cons Browse

OK Cancel Help

You have a couple of options you can set. You can choose to extract the consensus for all contigs or just one. If you choose all contigs (and you actually have more than one!), then the multiple consensuses will be placed one after another in the same file. If you want to extract only one contig then you must supply the contig identifier. You can see this by holding the mouse over the contig you are interested in, in the contig selector. Its identifier will be shown at the bottom of the display.

You also have a couple of other options. You can choose to strip the pads from the consensus sequence. Pads are the gap characters (stars) which are used to space the alignment. You usually don’t want any of these left in your consensus, so you can choose to have them removed. You can also choose the format for your sequence. You have the choice of experiment, fasta or staden. It is nearly always best to choose fasta format, as this is the most universally recognised format. Finally you get to choose your output filename. The file will be created in the same directory as your database, unless you have used the “Browse” button to select a different directory.

Troubleshooting

There are a couple of problems you may come across when trying to use Staden, which can be quickly diagnosed and corrected. If you are having a problem then look through the list below to see if it is listed. If you are still having trouble, then contact Simon Andrews in Bioinformatics on x6223 (or simon.andrews@bbsrc.ac.uk).

Errors importing from a pregap.passed file

If you try to import a series of files from a pregap.passed file and when you press OK the computer beeps continually for a few seconds, and the output window shows errors like this:

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Processing       22 in batch
File name ml2C-a5164g11.p1c.exp
Failed file ml2C-a5164g11.p1c.exp written to error file
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

The usual reason for this behaviour is that the pregap.passed file along with all the other trace files were in a different directory to your database. To successfully import files using a pregap.passed file they must all be in the same directory as the Gap database. If you really need to bring in files from another directory then you can do this by using the "Input reading names from Selection", then manually selecting all of the .exp files.

Errors importing from a single file

Another common error occurs when you have selected a single file to import (either an exp file or a raw sequence file). If, after you have pressed the OK button you hear a lot of beeping and see errors like this:

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Processing       94 in batch
File name GCGGCGCCTTCAGCAGGGAAAAGTGACAATTGCTGAGCTT
Failed file GCGGCGCCTTCAGCAGGGAAAAGTGACAATTGCTGAGCTTwritten to error
file
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

The reason for this type of error is that although you are trying to import a single file, Gap still thinks you are trying to import a list. To correctly import a single file you must change the "Input reading names from" option of the Normal Shotgun Assembly to read ***"Selection"***.

Normal Shotgun Assembly is greyed out

If you try to import sequences with Normal Shotgun Assembly, and find it is greyed out, and cannot be selected this is because you have not yet opened a database for the sequences to go into. Either open an existing database with File → Open, or create a new one with File → New.