

MacVector with Assembler 13.0

for Mac OS X

Contig Assembly Tutorial

MacVector, Inc.
Software for Scientists

Copyright statement

Copyright **MacVector, Inc**, 2014. All rights reserved.

This document contains proprietary information of **MacVector, Inc** and its licensors. It is their exclusive property. It may not be reproduced or transmitted, in whole or in part, without written agreement from **MacVector, Inc**.

The software described in this document is furnished under a license agreement, a copy of which is packaged with the software. The software may not be used or copied except as provided in the license agreement.

MacVector, Inc reserves the right to make changes, without notice, both to this publication and to the product it describes. Information concerning products not manufactured or distributed by **MacVector, Inc** is provided without warranty or representation of any kind, and **MacVector, Inc** will not be liable for any damages.

This version of the sequence assembly tutorial was published in March 2014.

Contents

| | |
|---|----------|
| INTRODUCTION | 4 |
| QUICK START | 4 |
| SAMPLE FILES | 5 |
| GLOSSARY | 5 |
| TUTORIAL | 6 |
| Creating and Populating a Project | 6 |
| The Project Window | 7 |
| Saving and Opening Assembly Projects | 9 |
| Base Calling with Phred | 10 |
| Viewing Base Calls | 11 |
| Masking Vector Sequences with cross_match | 12 |
| Assembling Sequences using phrap | 16 |
| Editing a Contig | 18 |
| Saving the Consensus Sequence | 23 |
| Analyzing Contig Sequences | 23 |
| Dissolving Contigs | 25 |
| Reassembling Contigs | 25 |

Introduction

MacVector Assembler is an add-on module for MacVector. It provides an intuitive interface to the industry standard `phred`, `cross_match` and `phrap` algorithms developed by Phil Green's group at the University of Washington. This tutorial will guide you through the process of assembling sequences using MacVector Assembler. Read the quick start section to quickly get going with a simple assembly project.

Note: see the other tutorials for reference assembly with Bowtie and *de novo* assembly with Velvet.

Quick Start

To assemble a set of sequences follow these steps;

- Start MacVector and choose **File | New | Assembly Project** to create a new empty project file.
- Click on the “**Add Seqs**” tool bar button, then select the sequence and/or chromatogram files you wish to assemble and click on the **Open** button. Note that you can hold down the **<shift>** key to select multiple sequences to import.
- Choose **Analyze | Base Call (phred)** or click on the **phred** toolbar icon to run the phred algorithm on all of the chromatogram-based sequences in the project. Note that if no sequences are selected, phred will be run on ALL of the files in the project.
- (Optional) Chose **Analyze | Vector Trim (cross_match)** or click on the **cross_match** toolbar icon to mask vector sequences in the reads. You will need to import the vector sequences you used in the vectors tab of the cross_match dialog. The files can be in any file format supported by MacVector.
- Choose **Analyze | Assemble (phrap)** or click on the **phrap** toolbar icon to assemble all of the sequences of the project.
- After assembly, overlapping sequences will be removed from the project and be replaced by contigs. Double-click on a contig to open it in a contig editor.

- You can not only edit sequences in the contig editor but you can also directly run any MacVector nucleic acid analysis function on the contig consensus sequence.
- Finally you can save the consensus sequence in MacVector format by choosing **File | Export** from the contig editor window. You can also save the assembly project itself at any time and you will be prompted to save any changes when you close the project window.

Sample Files

After installing MacVector Assembler, you will find example files for this tutorial in the folder;

```
/Applications/MacVector/Tutorial Files/Contig  
Assembly/
```

There is a `Trace Files` folder containing 32 trace files in SCF format along with two vectors (pSG933 and Tn1000) used in the sequencing experiment.

Glossary

There are a few terms that are used in this tutorial that you may not be familiar with;

| Term | Description |
|---------------|--|
| Base Call | The interpretation of the peaks in a trace file to identify the most likely DNA sequence. |
| Chromatogram | The trace information from an automated sequencing machine. The terms “chromatogram files” and “trace files” are used interchangeably in this tutorial to describe the files generated by automated sequencing machines. |
| Consensus | The most likely sequence of a contig, determined by taking all of the overlapping sequences into account. |
| Contig | An assembly of two or more overlapping sequences. |
| Quality Value | A value assigned to a base call to reflect the probability that the base call is in error. Uses a scale from 0 to 99. |
| Reads | A generic term used to describe the collection of DNA sequences that were generated during the sequencing project to be assembled into a contig. Typically these are trace files, but they can also be plain sequences. |
| Trace | The chromatogram data generated by an automated sequencing machine. The terms “chromatogram files” and “trace files” are used interchangeably in this tutorial to describe the files generated by automated sequencing machines. |

Tutorial

Creating and Populating a Project

The first step in the tutorial is to create a new project and add some sequences to it.

Select **File | New | Assembly Project**. An empty assembly project window will open.

Click on the “+” toolbar button. In the dialog that opens, navigate to the MacVector/Tutorial Files/Contig Assembly/Trace Files/ folder.

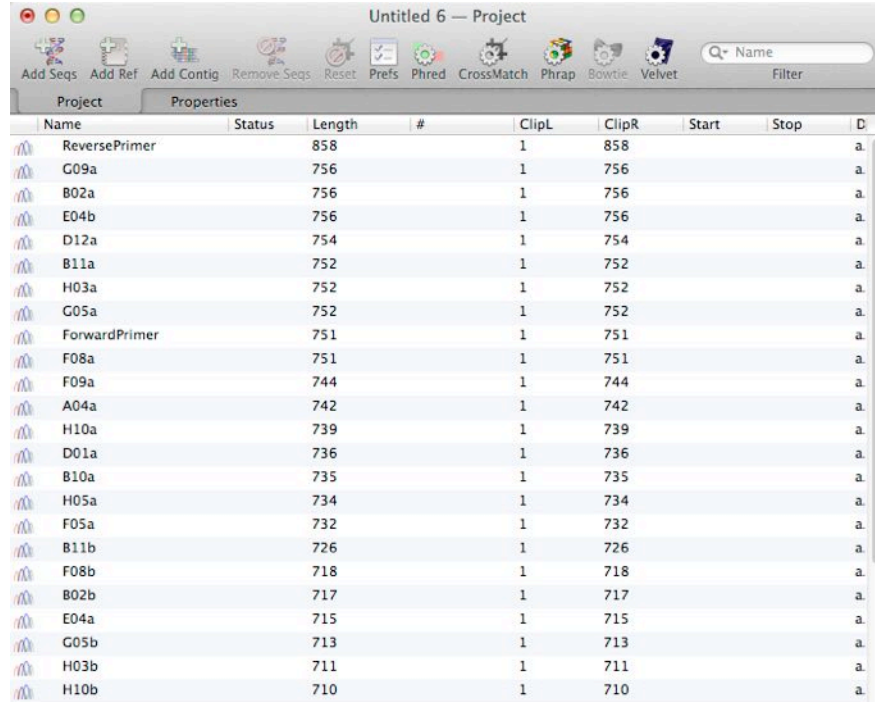
Click on the first file (A04a.scf) to select it, then scroll to the end of the list, hold down the <shift> key and click on the last file (ReversePrimer.scf) to select all of the files in the folder.

Finally, click on the **Open** button to import the selected files into the project.

Note that project makes a *copy* of all of the data that you import. If you subsequently edit the original files on disk, the data in the project will be unaffected. Similarly, any edits you make to the project data will not affect the contents of the original files.

MacVector Assembler can import chromatogram files in ABI, SCF and ALF formats. In addition, you can import plain sequence files in any file format supported by MacVector (including the FastQ format for next generation sequencing projects). There are no specific limits to the numbers or sizes of the sequences that you import. However, you may run into performance problems with projects containing large numbers of sequences. We recommend that you use a computer that has at least 0.5 MB of physical RAM for each chromatogram file you import for optimum performance i.e. use at least 512MB for a 1,000 sequence assembly. If you do see a slowdown importing, editing and saving large projects, adding more RAM to your computer is usually the most cost-effective way to improve performance.

The Project Window



| Name | Status | Length | # | ClipL | ClipR | Start | Stop | D |
|---------------|--------|--------|---|-------|-------|-------|------|----|
| ReversePrimer | | 858 | | 1 | 858 | | | a. |
| G09a | | 756 | | 1 | 756 | | | a. |
| B02a | | 756 | | 1 | 756 | | | a. |
| E04b | | 756 | | 1 | 756 | | | a. |
| D12a | | 754 | | 1 | 754 | | | a. |
| B11a | | 752 | | 1 | 752 | | | a. |
| H03a | | 752 | | 1 | 752 | | | a. |
| G05a | | 752 | | 1 | 752 | | | a. |
| ForwardPrimer | | 751 | | 1 | 751 | | | a. |
| F08a | | 751 | | 1 | 751 | | | a. |
| F09a | | 744 | | 1 | 744 | | | a. |
| A04a | | 742 | | 1 | 742 | | | a. |
| H10a | | 739 | | 1 | 739 | | | a. |
| D01a | | 736 | | 1 | 736 | | | a. |
| B10a | | 735 | | 1 | 735 | | | a. |
| H05a | | 734 | | 1 | 734 | | | a. |
| F05a | | 732 | | 1 | 732 | | | a. |
| B11b | | 726 | | 1 | 726 | | | a. |
| F08b | | 718 | | 1 | 718 | | | a. |
| B02b | | 717 | | 1 | 717 | | | a. |
| E04a | | 715 | | 1 | 715 | | | a. |
| G05b | | 713 | | 1 | 713 | | | a. |
| H03b | | 711 | | 1 | 711 | | | a. |
| H10b | | 710 | | 1 | 710 | | | a. |

Toolbar Buttons

The Project Window has the following functional toolbar buttons relevant to this tutorial;



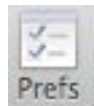
Add Seqs – use this button to add additional sequences to the project. You can also use **Edit | Add Sequences From File**



Remove/Dissolve – removes the selected sequences from the project. You can also use this button to dissolve selected contigs. The text of the button will change depending on the items you have selected. You can also use the **<delete>** key or **Edit | Remove Sequence** to accomplish the same functions.



Reset – this resets the trimming (sometimes called “clipping”) for the selected sequence(s), or for all sequences if nothing is selected in the window.



Preferences – this button opens up a dialog that lets you (a) configure the default appearance of the contig editor and (b) add vectors to the project.



Phred/CrossMatch/Phrap - these buttons invoke the corresponding calculation for base calling (phred), vector screening and trimming (cross_match) or assembly (phrap).

Tabs

The window has two tabs that display information related to the project. Like other MacVector tabbed windows, you can click on them to see a different view of the underlying data.

Project Tab



The project tab has a number of columns that display information about the individual sequences and contigs. Most of the columns can be sorted by clicking on the column header.

Name – the name of the sequence. All sequences and contigs in a project **MUST** have a unique name. If you try to import sequences with duplicate names, you will be prompted to choose how they should be handled. The icon next to the name indicates if the object is a contig, a trace or a plain sequence. You can directly edit this field to change the name.

Status – initially blank, the status field indicates if a sequence has been base called with phred ("P") or masked for vector sequences with cross_match ("X").

Length – the length of the sequence or contig.

- for contigs, this field indicates the number of reads that have been assembled. For sequences in a contig, the field indicates orientation using ">" for forward reads and "<" for reverse reads.

ClipL – the first residue from the 5' end that is not masked. Typically this will be “1”, although cross_match or phrap may change this.

ClipR – the last valid residue at the 3' end of a sequence. Initially, this is simply the last residue of the sequence, but cross_match and phrap may change this.

Start – for sequences in a contig, the start location of the sequence within the contig.

Stop – for sequences in a contig, the location of the last residue of the sequence within the contig.

Definition – any descriptions associated with a sequence.

You can double-click on an item to open up the editor associated with the object, e.g. the trace editor or the contig editor. Note that in this version, you cannot directly edit plain sequences by double-clicking on them – you should complete any editing on these before adding them to the project.

Properties Tab



The properties tab lists a variety of useful statistics about the project, including the number of files and contigs, the average quality value and the number of low quality residues in the assemblies.

```
Total Reads:          32
          Contigs:      1
    Singleton Reads:    0
Sum of Contig Lengths: 3423bp
Average Contig Length: 3423bp
    Avg Contig Quality: 88
    Low Quality Residues: 3
```

Saving and Opening Assembly Projects

You can save assembly projects at any time. They are saved in packages. These are a special type of folder that is treated as a single file by Finder. However, the actual assembly is saved inside this package in an XML format, meaning that the file contents are text and can (potentially) be viewed in any standard text editor such as TextEdit.

Choose **File | Save As...** As this is the first time you have saved the project, you will get prompted for a filename. Otherwise, use **File | Save** so the project will be saved with its current filename.

The small tutorial project should save within a second or two. Large projects may take some time to save – approximately 5 seconds for every thousand trace sequences on an average machine. A progress dialog is displayed during the save. You can cancel this and your original file will not be affected.

Close the project window. You will be prompted to save if you have made any changes since the last save.

Click on the **File** menu. A list of recent files is appended to the bottom of the menu. Select the name you saved the project under.

The project will open. Again, a progress dialog is displayed during the load as large projects may take some time to open. If you have a very large project, it may take a few seconds before the progress dialog is displayed.

Base Calling with Phred

Phred is an algorithm developed by Phil Green's group at the University of Washington. Phred re-evaluates the chromatogram peaks in a trace file, a process known as "Base Calling". Not only is phred typically more accurate than the default base callers used by automated sequencing machines, but it also assigns "Quality Values" to each individual base call. Phred uses a statistically significant logarithmic scale from 0 to 99 where 10 means there is a 1 in 10 chance that the call is in error, 20 means there is a 1 in 100 chance the call is in error, 30 means there is a 1 in 1,000 chance of an error etc. The values 98 and 99 are reserved to indicate residues that have been edited by the user. A phred score of 20 or more is generally considered to be an acceptable score. MacVector Assembler displays phred scores as a histogram above the sequence using colors to indicate the quality – scores below 20 are shown in red, scores of 20 or greater in green and edited residues (score 99) in blue.

Make sure you have no selections in the project window. To toggle a selection off, click on the selection while holding down the command (**⌘**) key.

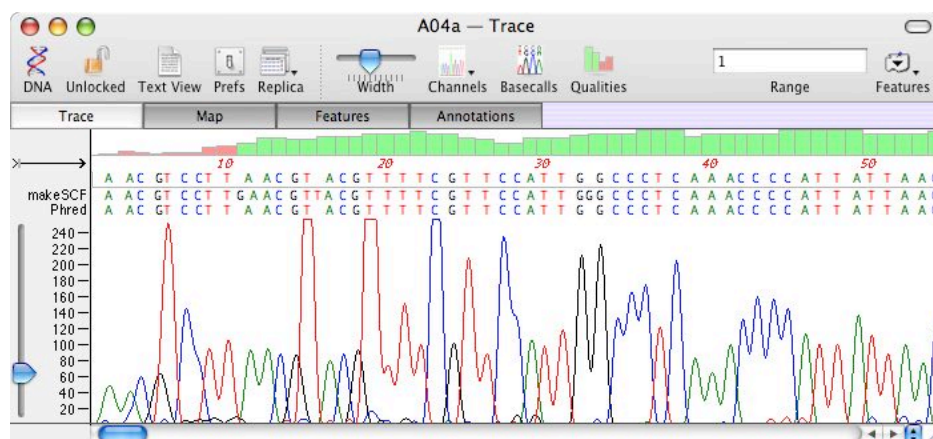
You can select a subset of sequences for analysis if you wish. However, if nothing is selected, all of the chromatogram sequences in the project will be submitted for analysis.

Choose **Analyze | Base Call (phred)** or click on the **phred** toolbar button

You should typically see the Job Manager displayed with a list of the submitted jobs and their progress. When the job(s) have completed, the project window will refresh to reflect the new base calls. The status of each entry changes to “P” to indicate you have run phred on the sequence.

Viewing Base Calls

Double-click on one of the phred-called sequences in the project window to open up the trace editor window.



When you open the trace editor window from an assembly project, two additional toolbar buttons are displayed.



Show Qualities – toggle this button to show or hide the quality histogram displayed over the top of the sequence.



Show Base Calls – toggle this button to show or hide the basecalls displayed immediately below the main sequence line.

You can see that the original base call for this sequence was generated using a utility called “makeSCF”. The phred base call is shown directly underneath this. You cannot edit the base calls – they are read-only. You can edit the upper sequence if you wish – this is the “active” editable sequence that is used in all assemblies and analyses. If you subsequently re-run phred on a sequence, it will replace the phred base call and will also replace any edits you have made to the active sequence.

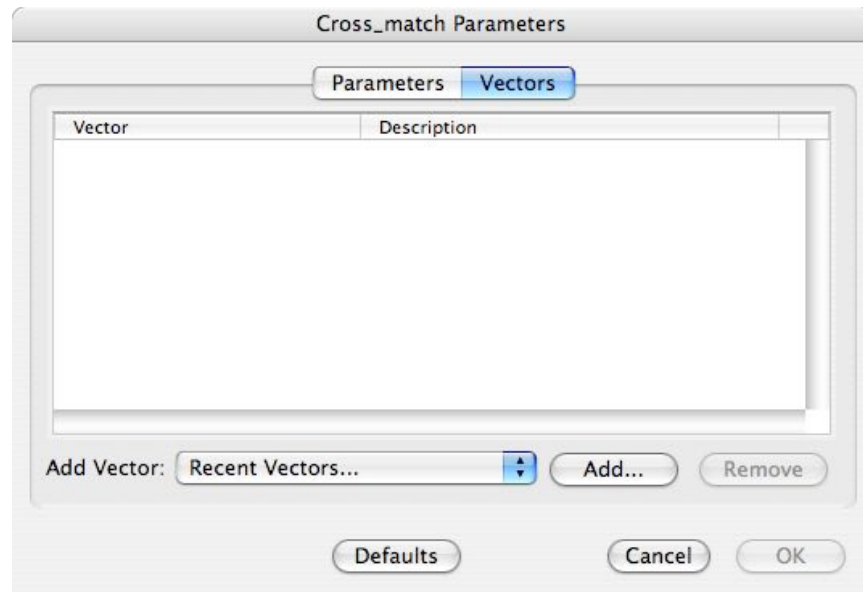
Masking Vector Sequences with `cross_match`

Typical sequencing projects use a directed or shotgun sub-cloning approach to generate short overlapping sequences that are then assembled into a single longer sequence. It is common for the reads to have vector sequences at the beginning and/or end which can interfere with the assembly. `Cross_match` is an algorithm that can be used to mask out any vector sequences to prevent this interference. This is not an absolutely essential step as phrap (the assembly algorithm we will use) can often detect the vector sequences in a collection of similar sequences. However, using `cross_match` is highly recommended to reduce the likelihood of anomalous assemblies.

Make sure you have no selections in the project window. To toggle a selection off, click on the selection while holding down the command (**⌘**) key.

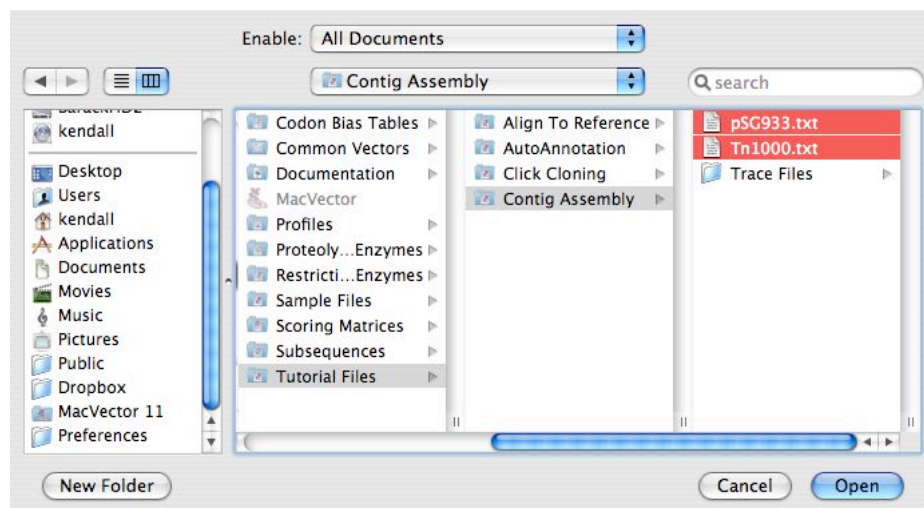
Choose **Analyze | Vector Trim (`cross_match`)** or click on the **crossmatch** toolbar button.

The `cross_match` parameters dialog will appear. The algorithm needs to know which vectors were used in the sequencing experiments, so the dialog initially displays the empty **Vectors** tab.



Click on the **Add...** button to bring up the file selection dialog.

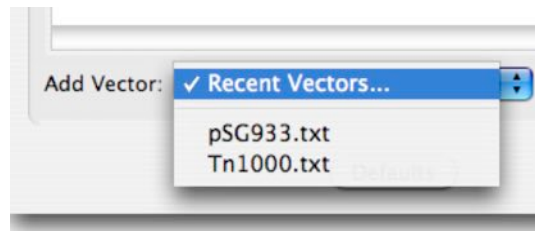
Navigate to the MacVector/Tutorial Files/Contig Assembly/ folder and select the files pSG933.txt and Tn1000.txt



Click on the **Open...** button to add the sequences to the vectors tab.

The tab will refresh to reflect the new vectors that have been added.

Click on the popup menu titled **Recent Vectors...**



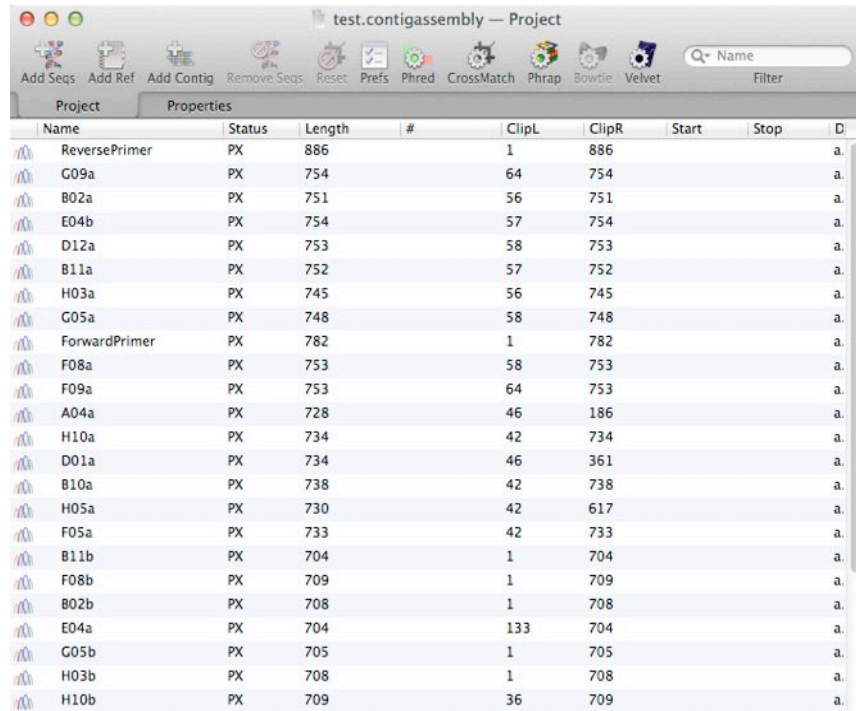
Note how the names of the files have been added to this menu. The menu remembers the last 20 vector files you added to any project, so you can use this as a shortcut to rapidly import common vectors into any new projects you create.

The files you select can be in any sequence format supported by MacVector. FastA, GenBank and EMBL format files can have multiple sequences in them, so you can create just one file with all of the vectors you use on a regular basis to further simplify vector importing.

Click on the **Parameters** tab in the dialog to view the other cross_match parameters. For this tutorial, we will accept the default values. Click on the **Defaults** button to make sure you are using the standard settings.

Click on the **OK** button to dismiss the dialog and run the algorithm.

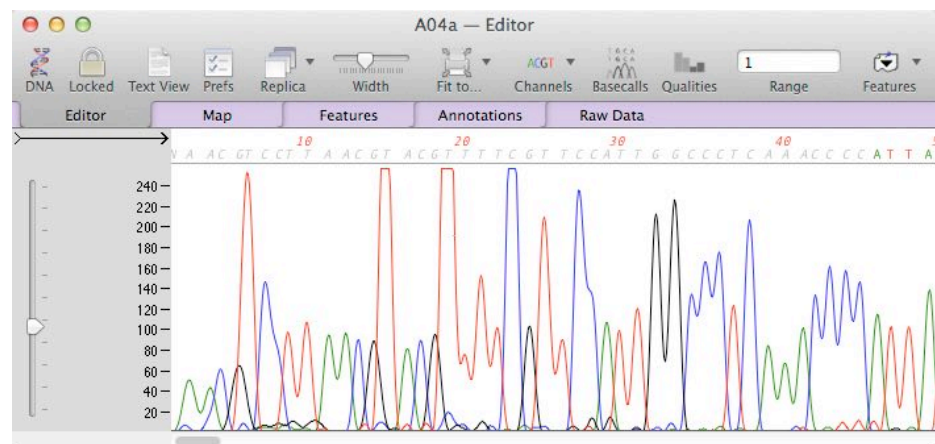
The algorithm should complete within a few seconds. The project window then updates with the new data.



| Name | Status | Length | # | ClipL | ClipR | Start | Stop | D |
|---------------|--------|--------|---|-------|-------|-------|------|----|
| ReversePrimer | PX | 886 | | 1 | 886 | | | a. |
| G09a | PX | 754 | | 64 | 754 | | | a. |
| B02a | PX | 751 | | 56 | 751 | | | a. |
| E04b | PX | 754 | | 57 | 754 | | | a. |
| D12a | PX | 753 | | 58 | 753 | | | a. |
| B11a | PX | 752 | | 57 | 752 | | | a. |
| H03a | PX | 745 | | 56 | 745 | | | a. |
| G05a | PX | 748 | | 58 | 748 | | | a. |
| ForwardPrimer | PX | 782 | | 1 | 782 | | | a. |
| F08a | PX | 753 | | 58 | 753 | | | a. |
| F09a | PX | 753 | | 64 | 753 | | | a. |
| A04a | PX | 728 | | 46 | 186 | | | a. |
| H10a | PX | 734 | | 42 | 734 | | | a. |
| D01a | PX | 734 | | 46 | 361 | | | a. |
| B10a | PX | 738 | | 42 | 738 | | | a. |
| H05a | PX | 730 | | 42 | 617 | | | a. |
| F05a | PX | 733 | | 42 | 733 | | | a. |
| B11b | PX | 704 | | 1 | 704 | | | a. |
| F08b | PX | 709 | | 1 | 709 | | | a. |
| B02b | PX | 708 | | 1 | 708 | | | a. |
| E04a | PX | 704 | | 133 | 704 | | | a. |
| G05b | PX | 705 | | 1 | 705 | | | a. |
| H03b | PX | 708 | | 1 | 708 | | | a. |
| H10b | PX | 709 | | 36 | 709 | | | a. |

In addition to the status of each sequence changing to “PX” to indicate that they have been trimmed with cross_match, many of the **ClipL** entries now show values other than “1” indicating that vector sequences were masked at the beginning. The sequence **A04a** has a particularly short insert and you can see that its **ClipR** value is now only 186.

Double-click on the sequence **A04a** to open up a trace editor window.



Note how the masked residues are shown in gray italics. If you scroll to the right, you will find additional masked residues from 186 onwards. Close the window before proceeding.

Assembling Sequences using phrap

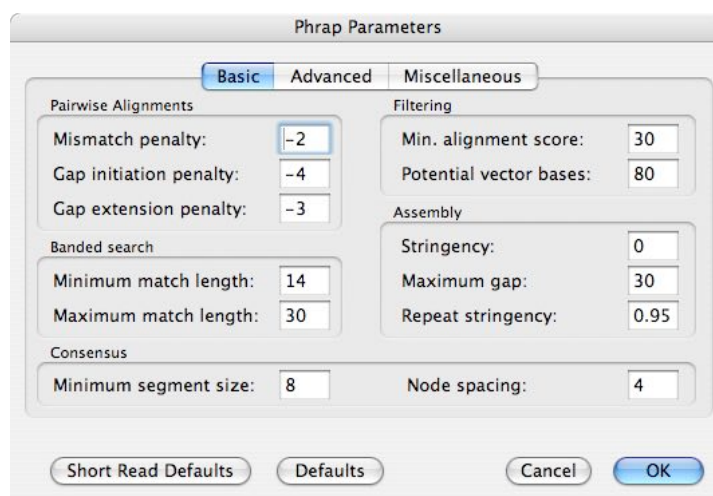
Phrap is the assembly algorithm from the University of Washington that has been incorporated into MacVector Assembler. It is designed to work in concert with phred and cross_match – in particular it understands quality values and will use them to make better assemblies, particularly in areas with repetitive sequences. Phrap also calculates quality values for each residue in the consensus sequence using the same scale as phred. However, for assemblies, a value of 40 (1 error in 10,000) is considered an acceptable value.

Phrap is described in more detail in the `phrap.pdf` document that can be found in the `MacVector 12.0/Documentation` folder. This is the original documentation from the University of Washington. It is somewhat technical in places, but it describes the assembly algorithmic strategy and the effects of changing various parameters in great detail.

Make sure you have the Project window front most, then be sure you have no selections in the Project window. To toggle a selection off, click on the selection while holding down the command (**⌘**) key.

Choose **Analyze | Assemble (phrap)** or click on the **phrap** toolbar button.

The phrap parameters dialog will appear. Not all of the parameters described in `phrap.pdf` are available in the dialog. However, it is unlikely that you will ever need to adjust any parameters other than those displayed in the **Basic** tab.



Click on the **Defaults** button to make sure you are using the correct default settings.

The **Short Read Defaults** button is useful if you are using if you are assembling short (<100 nucleotide) sequences from Next Generation Sequencing machines.

Click on the **OK** button to dismiss the dialog and run the algorithm using the default values.

Phrap is a remarkably fast algorithm and the assembly should be complete within a few seconds. Even with large (>1,000 reads) projects, assembly rarely takes more than a few minutes. Again, you can close the progress dialog and carry on working elsewhere in MacVector if you expect assembly to take a long time. Once assembly is complete, the project window is updated to reflect the data change. In this case, all of the reads should be assembled into a single contig.

Click on the disclosure triangle next to the contig to reveal the contents of the contig.

| Name | Status | Length | # | ClipL | ClipR | Start | Stop |
|---------------|--------|--------|----|-------|-------|-------|------|
| ▼ Contig 2 | | 3450 | 32 | 1 | 3450 | | |
| A04a | PX | 728 | -> | 45 | 186 | 3243 | 3970 |
| A04b | PX | 711 | <- | 1 | 672 | 2631 | 3331 |
| B02a | PX | 754 | -> | 56 | 753 | 1097 | 1847 |
| B02b | PX | 710 | <- | 1 | 676 | 482 | 1190 |
| B10a | PX | 741 | -> | 42 | 738 | 1628 | 2361 |
| B10b | PX | 700 | <- | 2 | 665 | 1011 | 1708 |
| B11a | PX | 763 | -> | 57 | 763 | 1982 | 2732 |
| B11b | PX | 708 | <- | 23 | 674 | 1371 | 2076 |
| D01a | PX | 735 | -> | 46 | 362 | 3068 | 3801 |
| D01b | PX | 708 | <- | 4 | 673 | 2455 | 3152 |
| D12a | PX | 762 | -> | 58 | 762 | 1785 | 2538 |
| D12b | PX | 709 | <- | 1 | 675 | 1175 | 1880 |
| E04a | PX | 704 | -> | 137 | 704 | -127 | 577 |
| E04b | PX | 756 | -> | 57 | 756 | -35 | 721 |
| F05a | PX | 735 | -> | 42 | 735 | 1369 | 2101 |
| F05b | PX | 700 | <- | 1 | 665 | 752 | 1449 |
| F08a | PX | 759 | -> | 58 | 755 | 1489 | 2243 |
| F08b | PX | 710 | <- | 1 | 675 | 878 | 1585 |
| F09a | PX | 764 | -> | 64 | 764 | 2333 | 3085 |
| F09b | PX | 705 | <- | 1 | 670 | 1738 | 2434 |
| ForwardPrimer | PX | 790 | -> | 1 | 698 | 1 | 790 |
| G05a | PX | 754 | -> | 58 | 751 | 645 | 1396 |

The items within the contig are grayed out to indicate that you cannot open them individually. This is to prevent you from

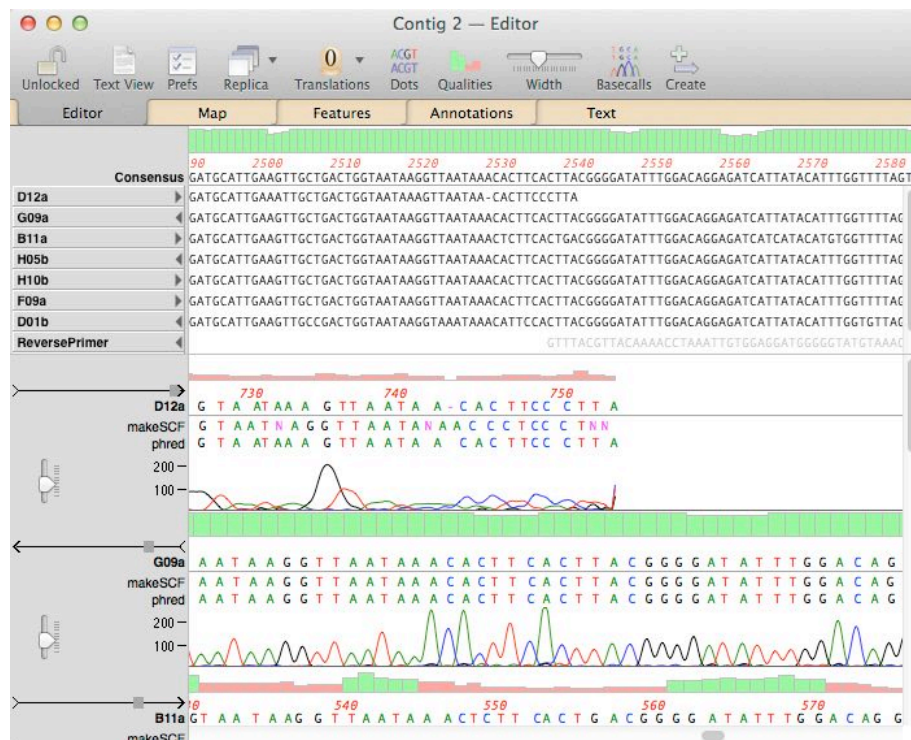
inadvertently changing the sequence of a trace that has been carefully aligned in a contig. However, you do have full editing control from within the contig editor (see below).

Note how the **#**, **Start** and **Stop** columns have been updated to display additional information. The number of reads assembled in the contig is indicated on the top line, while the orientation of each read in the contig is indicated on the other lines. The start and stop locations of each read within the contig are also indicated in the appropriate columns.

Editing a Contig

Although phrap does an excellent job of assembling reads and generating an accurate consensus sequence, there may be times where you need to edit the assembly, particularly if you have poor quality chromatograms, or areas of the contig that have low coverage.

Double-click on **Contig_1** to open up the contig in the contig editor.



The contig editor window is based upon the Align to Reference editor from MacVector, with a number of important differences;

- Base calls and quality values can now be displayed, controlled by the same toolbar buttons used in the trace editor.
- There is no “reference” sequence.
- The overlapping sequences can now be displayed in “tiled” or “untiled” mode.

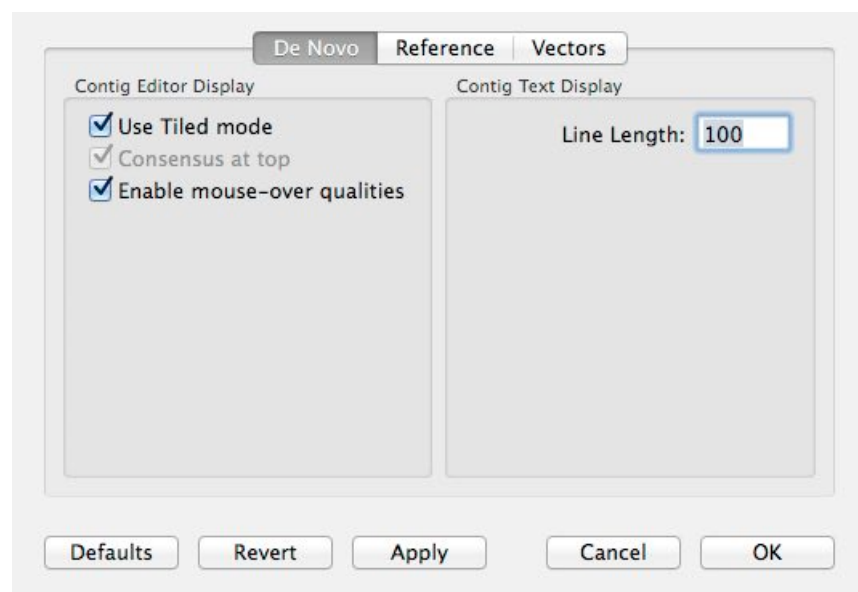
The tiled/untiled mode needs additional explanation.

In “tiled” mode, each component sequence is given its own dedicated line in the upper panel. This is fine for relatively small assemblies (< 50 reads) but for large assemblies, the layout is impractical as there is too much white space and the user spends too much time scrolling to find the right sequences to edit.

In “untiled” mode, only those sequences that actually overlap the currently visible consensus sequence are shown on the screen. This minimizes the amount of white space and reduces the need for vertical scrolling. The downside to this approach is that the reads may “move about” as you horizontally scroll through a contig.

In addition to the tiled/untiled mode, you can also choose to display the consensus sequence at the top of the panel, or in the center of the panel. This is controlled by the project preferences, accessed by clicking on the Preferences toolbar button.

Click on the Preferences icon  to open up the project preferences dialog.

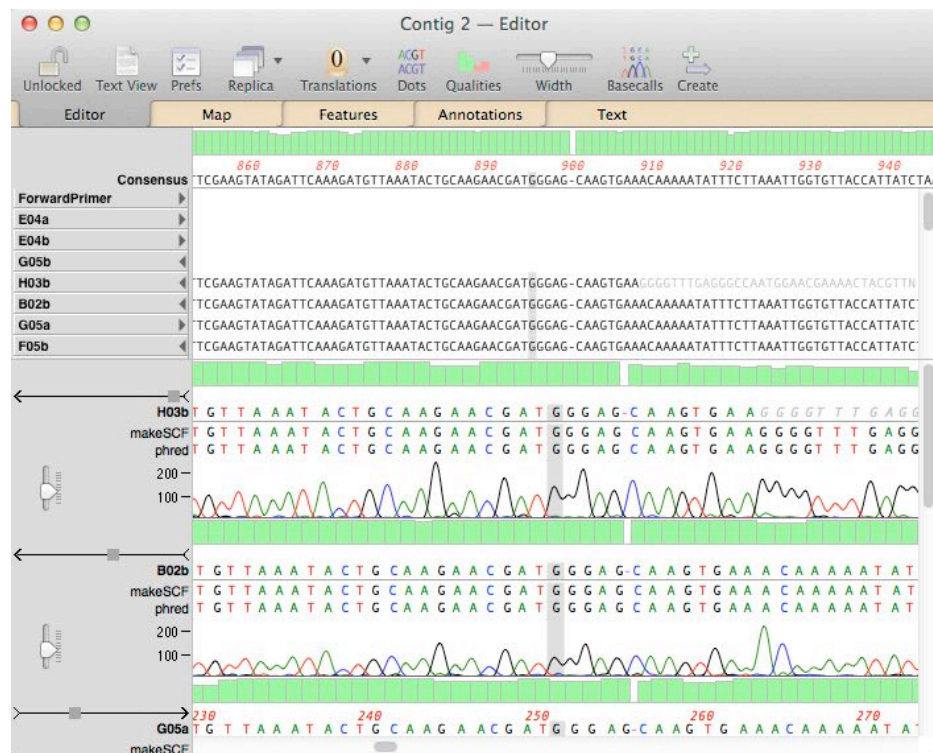


Select the **Use Tiled Mode** checkbox and click the **OK** button. Note that in tiled mode, the consensus is always placed at the top of the panel.

The contig editor display will change to use tiled mode. For the remainder of the tutorial, set the display to non-tiled mode with the consensus at the top of the panel. You may also want to increase the size of the window so that you can see more data at one time. You may also want to adjust the size of the upper panel by clicking and dragging on the vertical bar that separates the upper and lower panes.

Click on any residue on the consensus line.

The residue highlights, but the display also resets so that all of the traces overlapping that residue become centered in the lower multiple trace panel.



You can use this feature to click on any dubious consensus residue and immediately see the overlapping traces aligned at that position. Note that the consensus sequence is highlighted using your primary highlight color (pale purple in the screenshot) while the reads and traces are shown in a secondary highlight color (gray).

Press the right arrow button on the keyboard. Continue to hold it down.

This will scroll the contig to the right. You could potentially slowly scroll through the entire contig this way. Note that whenever the consensus is highlighted, the traces are always aligned and centered at that position.

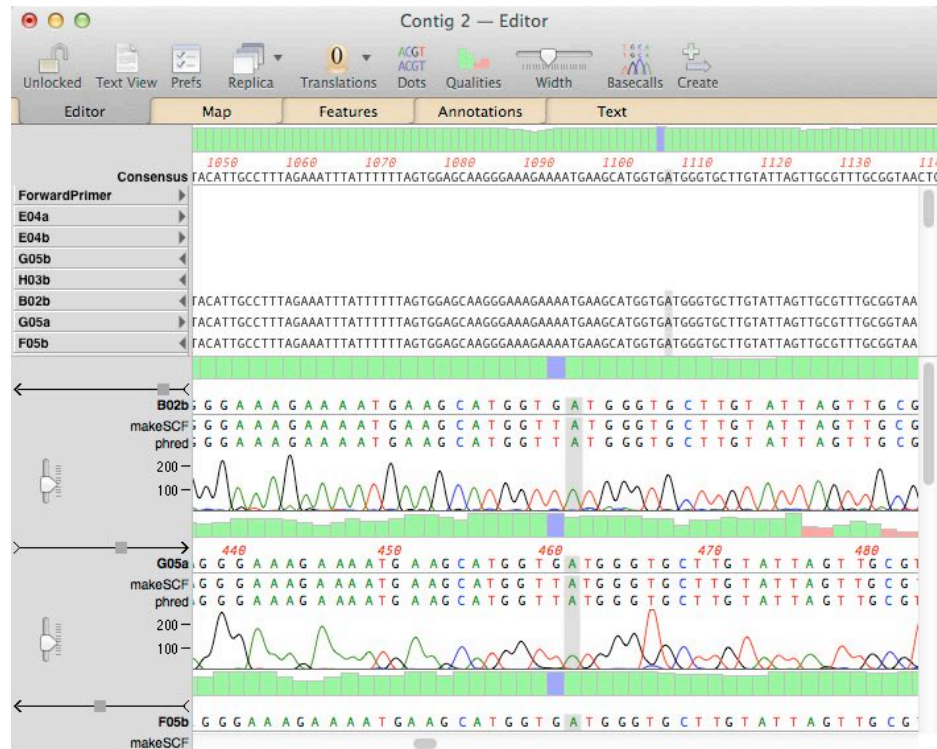
Click on one of the read sequences, either in the upper pane or in the lower multi-trace pane.

In this case the display does not reset to align the traces at the selected position. However, the primary highlight shifts to the selected residue (in both the upper and lower panes) to indicate which character will get changed if you press a valid key.



Select any residue in one of the reads and press a different DNA character key.

Note that you can directly edit the consensus sequence. It is usually calculated *indirectly* from the overlapping reads. However, if you do edit the consensus sequence, all of the overlapping Read sequences are edited to match the consensus.



The residue changes to the chosen character and the quality value changes to 99, represented by the blue histogram. The value 99 is very important for consensus recalculation as it always overrides all other quality values. This has two important implications;

1. If you edit a residue to be a valid DNA character, the consensus will always change to match that character as it overrides all other considerations.
2. If you edit two residues at the same position but in different reads, and they do not agree, the consensus will be given an ambiguity character.

Other than overwriting characters with the correct DNA residue, there are some other editing functions you should be aware of;

- Selecting a residue and typing a **<backspace>**, **<delete>**, **<space>** or a “-” character will replace the residue with a gap – this effectively deletes the residue while retaining alignments to the left of the deletion.
- Holding down the **<option>** key and typing a residue will insert that residue into the read, sliding the entire read beyond that point one space to the right.
- If you make an edit that results in every read at that position containing a gap character, the gap gets closed up.

Choose **Edit | Undo Typing**. As with most MacVector functions, there is just a single level of Undo.

Close the contig editor window.

Changes to contigs in the contig editor are considered to be changes to the project, so you do not get prompted to save those changes until you try to close the project window, rather than the contig window.

Choose **File | Save**. You will be prompted for a suitable filename.

Saving the Consensus Sequence

You can save the consensus sequence of a contig in MacVector format at any time.

Double-click on a contig in the project window to open up the contig editor for that contig.

Choose **File | Export....** You will be prompted for a suitable filename. By default, the file will have an “.axml” extension. Choose ‘MacVector NA Sequence’.

The file will be saved in MacVector single sequence format. Any gaps in the consensus sequence will not be present in the saved file. The locations of the reads ARE written to the file as features, but this behavior may change in future.

Analyzing Contig Sequences

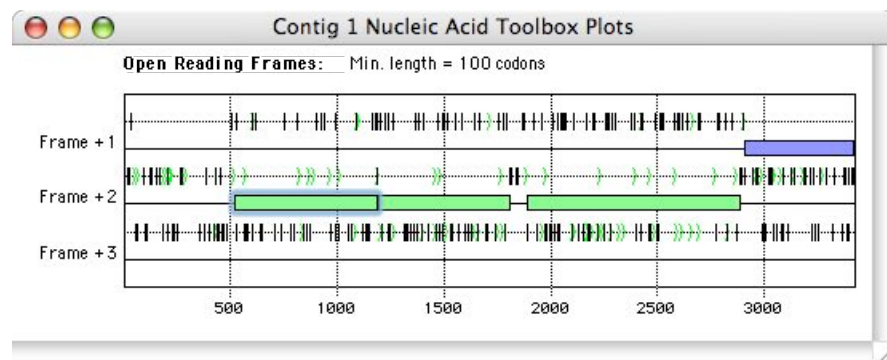
A contig editor window is functionally equivalent to any normal MacVector single sequence window. You can run any nucleic acid sequence algorithm on the contig – in every case, it is the **ungapped** consensus sequence that is analyzed. Any gaps that appear in the contig editor consensus sequence are there only to maintain alignment with the overlapping reads. All analysis functions (along with save and copy functionality) strip out any gaps before analysis.

Double-click on the contig in your project window to open the contig editor if it is not already open.

Choose **Analyze | Nucleic Acid Analysis Toolbox**. The normal Nucleic Acid Analysis Toolbox dialog will be displayed.

Select the **Open Reading Frames** checkbox, then click **OK** to initiate the analysis.

In the graphic window that opens, select the first open reading frame in the “green” frame as shown below (outlined in a pale blue highlight).



With the selection still in place, click on the title bar of the **Contig 1** window (or anywhere in that window). Note that the consensus sequence is selected.

Choose **Analyze | Translation....** The Translation dialog will be displayed. Make sure the **Create new protein** checkbox is selected, then click on the **OK** button.

A new protein window will be displayed. The protein sequence is a translation of the consensus sequence with all of the gaps removed. The same principle applies to all MacVector analysis functions – you can run any nucleic acid analysis (e.g. restriction enzyme analysis, or an online BLAST search) directly from the contig editor window and it is the *ungapped consensus* that gets analyzed. This allows you to get instant feedback on edits affecting the consensus sequence without requiring clumsy export to a different analysis module.

Dissolving Contigs

Make sure you have saved the assembly project you are working on. Many of the functions that dissolve or significantly modify contigs cannot be undone.

In the assembly project window, select a contig and then either click the **Dissolve** toolbar button, press the **<delete>** key or choose **Edit | Remove Sequence**.

The contig will be dissolved into its constituent reads. The project window updates to indicate the fact that all of the reads that were in the contig have now been returned to the root of the project.

Note that any edits you made to the individual reads will be maintained, although all gaps will be removed. It is essential that the edits be retained as this allows you to edit sequences in a “bad” contig and then reassemble them taking your edits into account. This is particularly important when assembling sequences containing closely related repeats. phrap will not assemble overlapping sequences that have mismatched edited bases (i.e. that have quality values of 99) – you can use this to force misassembled repeats to split by editing the mismatched bases and re-assembling.

Reassembling Contigs

In the assembly project window, select a contig and then choose **Analyze | Assemble (phrap)**. Accept the default parameters and click on the **OK** button.

You should see that the contig is dissolved and the assembled sequences temporarily appear in the project window. After assembly is complete, a new contig will appear in the project window.

The contig is dissolved prior to reassembly because there is no guarantee that the contig will reassemble with the same sequences as before. If you have edited sequences in the contig, or chosen different phrap parameters, one or more sequences may no longer assemble.

You can select any combination of contigs and sequences in the project window to (re)assemble just those items.