Mutation Detection Using the Staden Package

1 pregap4

1.1 Obtaining the course data

This exercise contains some sequencing data for you to experiment with. This is in the same directory as the Staden Package installation and should be copied to a new directory you create called 'exercise'. So firstly point your explorer at the Staden installation root (in many cases this will be *C:/Program Files/Staden Package/)* and append the following directories to the path to reach the desired directory '*course/data/mutations'* and copy all of the data within to your new exercise directory.

You should see lots of files ending in *.scf* and one file ending in .embl. The .scf files are the raw sequencing files produced from an ABI sequencing instrument, including two traces which we will use as references. They are in a different format than ABI use, but the software you will use can either format. The EMBL file is the BRCA1 complete mRNA sequence, as fetched from the EBI using SRS, which we will use as our reference sequence.

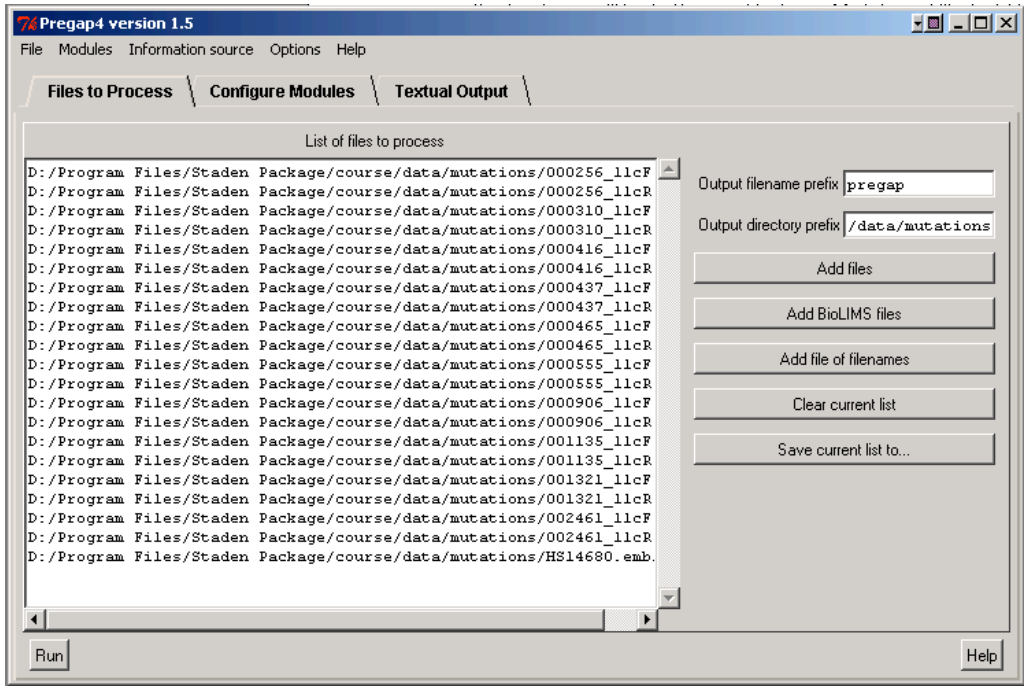
1.2 Starting **pregap4**

pregap4 is the tool for automating the various steps to take a binary sequence trace file and to convert it into something usable by the sequence assembly tools. Along the way it also performs the mutation analysis steps described in the talk.

Start **pregap4** by following this route through the start menu:

Start > Programs > Staden Package > Pregap4

1.3 Loading your files



pregap4 has a series of menus at the top of the window and a large tabbed-notebook underneath this (containing *Files to Process, Configure Modules* and *Textual Output*). We will start with the first tab - *Files to Process*.

Click on the **Add files** button to bring up a file browser. Find your exercise folder then select all the **.scf** files and the **HS14680.embl** file by selecting **Any** to view all files. Use **Ctrl + A** to select every file (as there are no unwanted files in this directory).

Press **OK** to accept your selection and **pregap4** will update to reveal the sequences you have selected.

1.4 Selecting the Modules

Now click on the **Configure Modules** tab to reveal the list of tasks that **pregap4** can perform. The left side of the window is the module name and the right side shows the configuration parameters for the currently selected module. Select different modules by clicking on their name to the left. Click on the [ ] or [x] boxes in the far left to enable or disable a module. Modules with an [x] next to them will be used by **pregap4**. There are a lot of options available in **pregap4**, but for our case we can ignore most of them so follow this simple walk-through. Any modules not listed below should be left disabled.

General Configuration

(This module cannot be disabled.) Accept **No** to the Get entry names from trace files question. **pregap4** is offering to rename the files by extracting the "sample names" contained within them. This is useful if you filled out the sample sheet using the ABI sequencing software, but this is not the case for our data.

Estimate Base Accuracies

Enable this

This simply measures signal to noise ratios as an alternative to re-calling the bases and assigning confidence values.

Trace Format Conversion
Initialise Experiment Files
Augment Experiment Files

Keep these three as they are - **enabled**.

The **Trace Format Conversion** compresses the trace files to save some disk space. The **Experiment Files** are textual copies of the traces, much like **EMBL** files. The **Augment** step applies the "naming scheme", which we come to later.

Quality Clip

Keep this **enabled**. It identifies the poor quality regions of the trace files and hides them from assembly (although the data is not deleted). The average confidence used to determine good data defaults to 15, which is approximately equivalent to a 3% error rate.

Sequencing Vector Clip
Screen for Unclipped Vector
Cloning Vector Clip

Disable all three of these.

These modules are intended for shotgun-sequencing projects where a vector sequence (bluescript, pUC, etc) will be present at the start of every sequence. This is not true with our data.

Reference Traces & Sequences

Enable this.

Here we tell **pregap4** what our reference trace files are and the reference sequence. There are three buttons labeled **Browse**. Click on each in turn and specify the following:

Reference Trace (+ve strand) = **000906 11cF.scf**

Reference Trace (-ve strand) = **000906 11cR.scf**

Reference Sequence = **HS14680.embl**

We shall see later on (in **gap4**) how these files are used.

Mutation Scanner
Heterozygote Indels

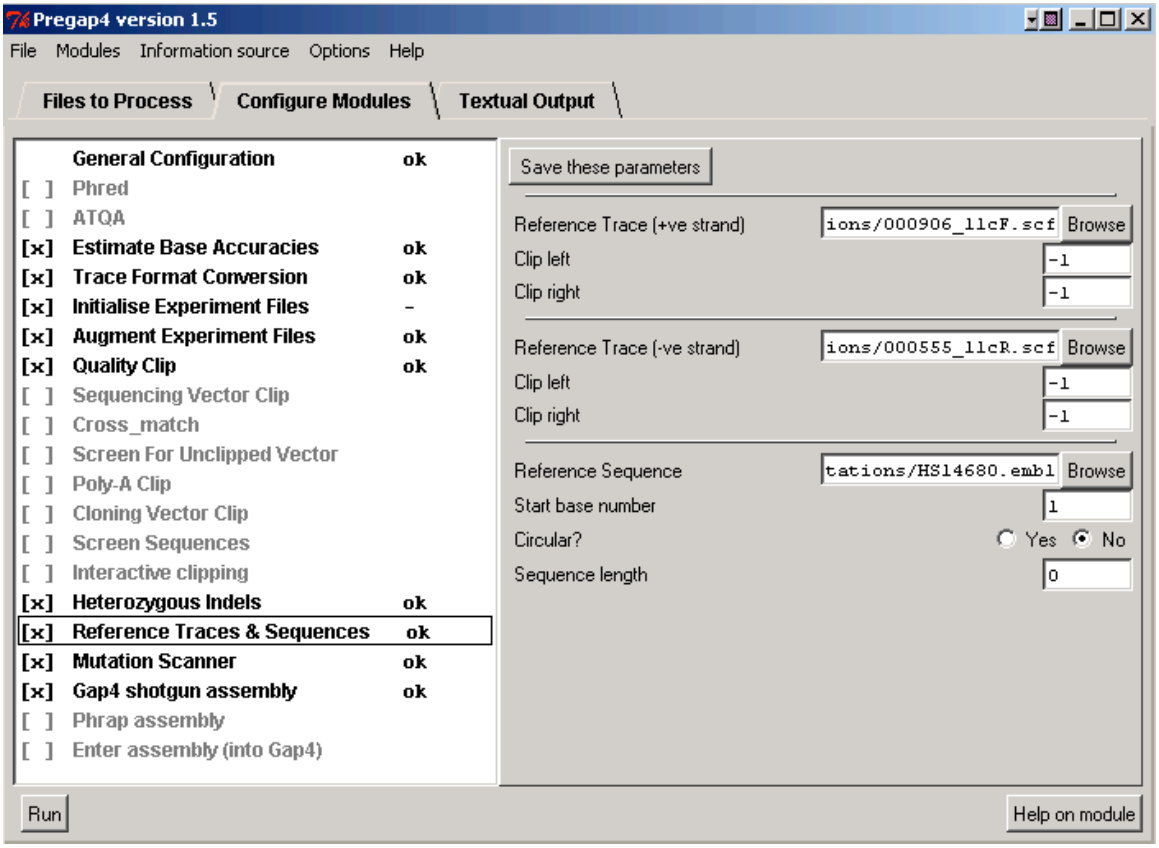
Enable both of these modules.

These perform the primary task of automatically detecting mutations.

gap4 shotgun assembly

Enable this module.

You will also need to specify the sequence database to align (assemble) your sequences into and as this is does not yet exist we shall create a new one. So answer **gap4** database name with **BRCA1** and click on the checkbox next to Create new **database**.



Assuming all has gone to plan you should only have the modules enabled that this picture shows. Here the Reference Traces & Sequences module has been highlighted to show the reference traces and sequences used.

1.5 Naming Schemes

How does **pregap4** know whether a sequence should be compared against the +ve strand or –ve strand reference sequence? We have to tell it. The easiest way of doing this is by careful naming of the files. Notice that our files are always in pairs; e.g. *000310 11cF.scf* and *000310 11cR.scf*. This is exon 11 primer c for patient *000310* in the Forward and Reverse direction. **pregap4** has a choice of several simple naming conventions, including this one.

Use the main **File** menu and select **Load Naming Scheme**. Press **Browse** and double click on the **mutation detection.p4t** file. Then press **OK** to close the Load naming scheme window.

1.6 Running **pregap4**

Before we run **pregap4**, it is wise to save all of our settings so far. Use the **File** menu and select **Save all Parameters (in all modules)**. This creates a configuration file in this directory so that any future **pregap4** sessions will be far quicker to set up.

Now click the **Run** button in the bottom left (or select Run from the Modules menu) to set **pregap4** going. It will automatically switch to the Textual Output window and display the results and ultimately claim:

```
*** Processing finished ***
```

Switch back to your explorer window and view your files once more. You will see lots of new files created, the most important being

pregap.passed - A list of files that passed all processing steps.
pregap.failed - A list of files that failed at least one processing step.
BRCA1.0 - {
BRCA1.0.aux - { A **gap4** assembly database containing your aligned sequences.

Assuming all has worked correctly, exit **pregap4**.

2 gap4

To start up **gap4** follow this route through the start menu:

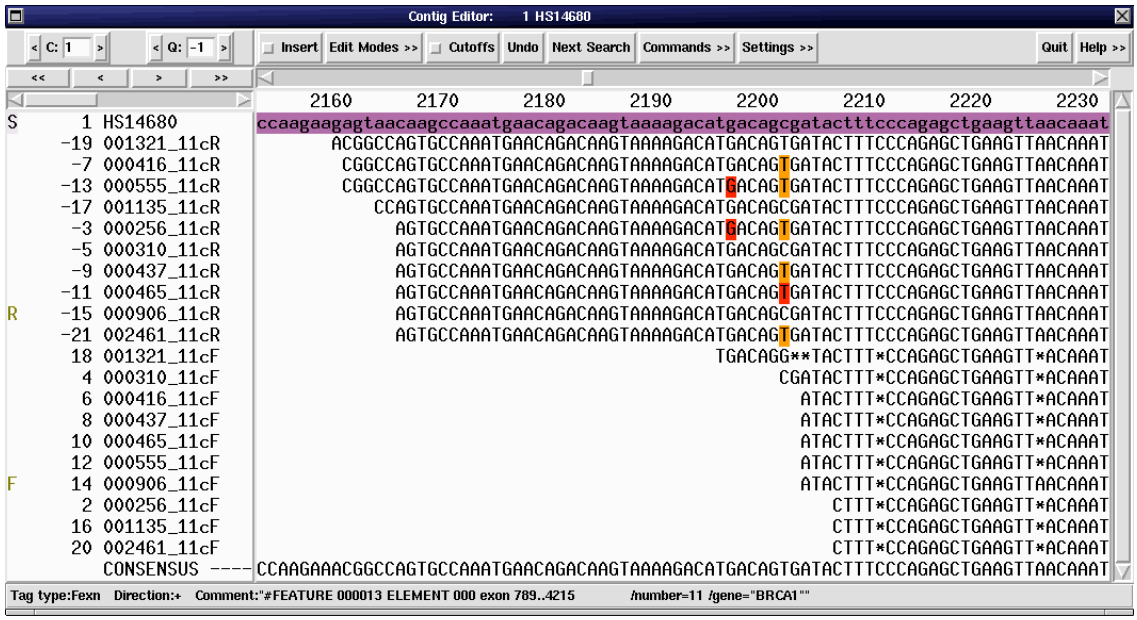
`Start > Programs > Staden Package > Gap4`

We need to open the assembly database produced by **pregap4**, so use the **File** menu and select **Open**. Double click on the **BRCA1.0.aux** file in your exercise directory. A new window titled *Contig Selector* will pop up. A *Contig* is a set of overlapping sequences - in this case our aligned patient and reference sequences. We only have the one aligned set and so the *contig selector* window is showing one single horizontal line representing that set. Move the mouse cursor over it and the line will become lighter and it will report the length and number of sequences underneath.

With the mouse cursor on top of the contig line, click the right mouse button to bring up a menu and select **Edit Contig**.

2.1 Contig Editor

A new window named **Contig Editor: 1 HS14680** will now appear.



At the top of the editor are a series of buttons, menus, and scroll bars. Underneath this we have the sequence names on the left and the DNA sequence on the right. The top most sequence should be HS14680. This is our reference sequence, which can be seen by the S to the left of its name.

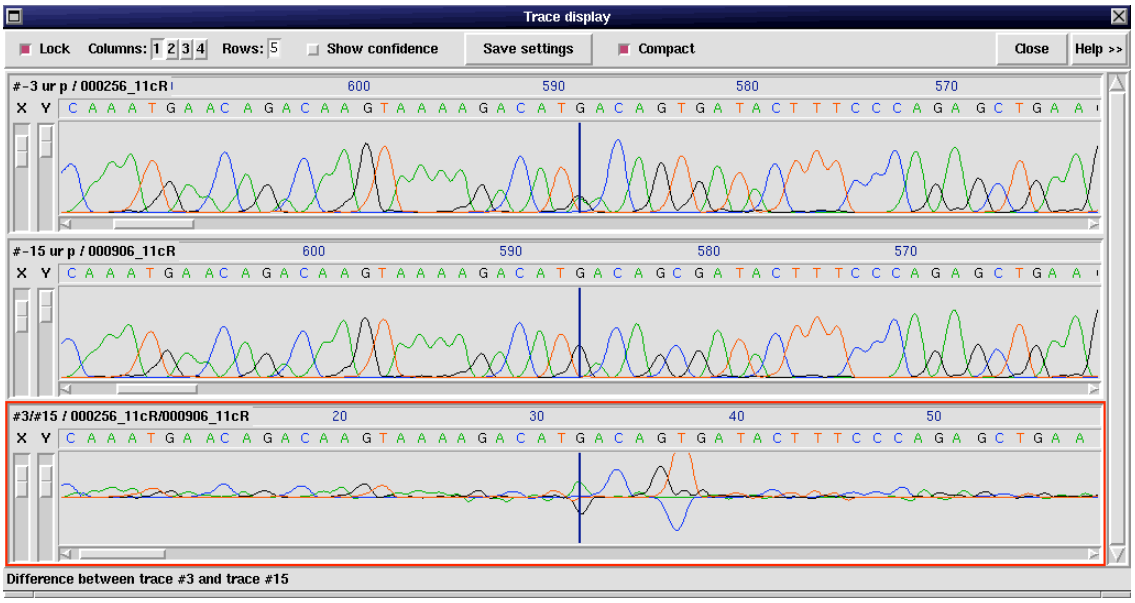
Scroll along so position 2200 is central. Note the reverse strand sequences (those with names ending in R) all have negative numbers before their names (e.g. *-19 001321 11cR*) while the forward strand sequences have positive numbers. This indicates that the reverse strand sequences have been reversed and complemented. (Note that if this is the other way around then it implies that the entire contig needs reversing and complementing, which was one of the other options available from the *contig selector* popup menu we saw a moment ago.) Click on the **Cutoffs** button at the top of the editor window. More sequence should appear, displayed in a light grey colour. This is the poor quality sequence which was hidden by **pregap4**'s **Quality Clip** module. We can see the confidence values assigned by the Phred base caller by using the editor Settings menu and selecting **Show Reading Quality**. The dark bases

indicate low confidence whilst the light bases indicate high confidence. Click on a base to see the confidence (and probability) shown at the bottom of the editor. Double click on a base to view the trace and confirm phred's base-calling decisions. Tidy up your display by clicking on the **Cutoffs** button once more to hide the poor quality data, deselecting **Show Reading Quality**, and closing any trace windows.

Try positioning your editor cursor somewhere by single clicking on a base call with the left mouse button. An inverse base (light text with a black background) represents the current position of the editing cursor. Try typing some new bases and you will see that the default editing mode is to over-type. Initially this is all you are allowed to do, but the editor allows much more than this using the *Edit Modes* menu. We do not have time to discuss all these features here, so for now press the **Undo** button several times until it beeps, informing you that there are no more edits to undo.

Click on the editor **Settings** menu and select **Highlight disagreements**. Now only bases that differ from the consensus (which is the same as the reference sequence in this case) are shown as base calls; all others are displayed as dots. This is a quick way of identifying base calls that differ from the reference sequence.

Double click at editor base position 2196 for sequences *000256_11cR* and *000906_11cR* traces,to show their traces. You should be able to see some differences between the two traces. Right click on the top trace and select **Trace Difference**. Left click on the bottom trace. The differences trace is now visible showing two mutations, one of which is at a heterozygous position. The heterozygous base has been labeled as G by the base caller, which is the same as the reference base call. So looking for base call differences alone is clearly misleading and *Highlight Disagreements* alone is not sufficient. Turn off **Highlight Disagreements** now (by going back into the *Settings* menu).



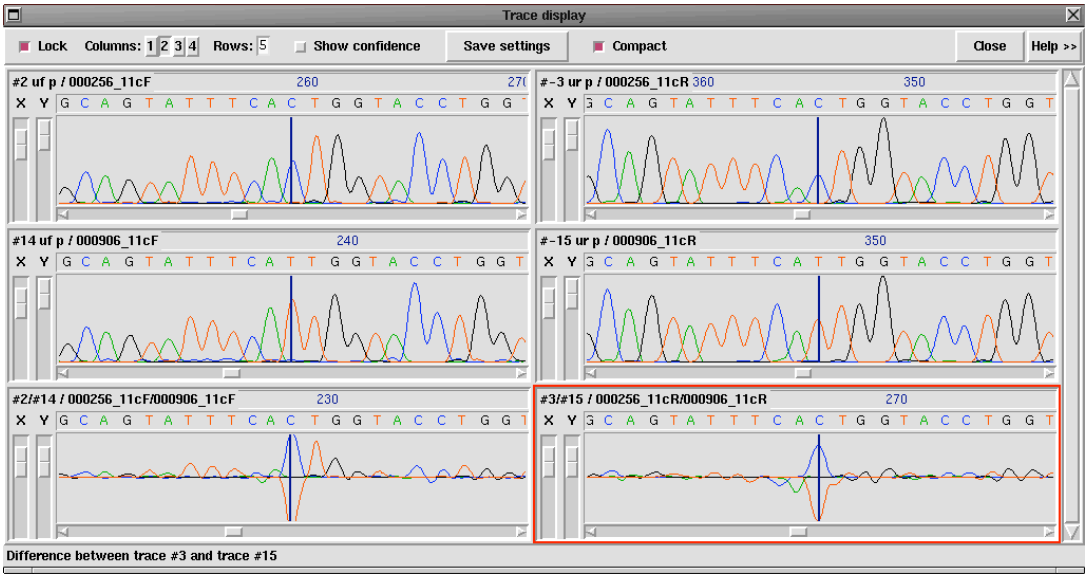
The difference plot is computed by aligning the two traces and subtracting one from the other, without any scaling in the Y axis. Hence when two traces are identical we expect a flat line central to the trace display. A mutation can be seen as a double peak, one above and one below this central line - two are clearly present in the picture above. A peak in one direction implies a change in the signal strength for one dye only without a corresponding change for a second dye. This is not a mutation, but is typically caused by the presence of an adjacent mutation (as the peak heights are strongly dependent on the preceding bases). Searching for double peaks in the difference plot is precisely how the automatic

methods within **pregap4** work. Double peaks sufficiently higher than the background noise will be automatically labeled and mutations and **pregap4** then adds "tags" (the red and orange coloured bases) the bases, which we see in the editor window.

Moving back to the trace display, drag the scrollbar on the bottom difference trace to the right until you see another obvious mutation. Note that the editor window has also been scrolling at the same time. You should be somewhere around editor base 2430 now.

To make it easier to verify when mutations have been automatically detected on both strands we can display the forward and reverse sequences vertically adjacent to one another in the editor. (This is what the reading name convention in **pregap4** was used to determine.) Use the Settings menu to select **Group Readings by Templates**. Note that one pair have an F and R next to the far left of their names - these are the forward and reverse reference traces that we specified in **pregap4**.

Now go back to the **Settings** menu, into the **Trace Display** sub-menu and select **Auto-diff Traces**. Try double clicking on a red or orange tag. You should now see 6 traces displayed in a 2x3 grid. The left hand sides shows the top (forward) strand and the right side shows the bottom strand.



Next turn on Translate **using Feature Tables** in the **Settings -> Status Line** menu. The bottom line of the editor should now be showing an amino acid translation of the BRCA1 gene. This information has been extracted from the CDS records in the EMBL reference sequence *HS14680*. This allows us to determine what the amino acid change is for each mutation, or whether it is a "silent" mutation. We could also use a genomic sequence for the reference (possibly containing multiple genes with multiple exons) in which case each exon is translated as appropriate. This then allows us to determine when mutations are in non-coding or splicing regions.

To get a summary of mutation positions use the **Report Mutations** command in the editor **Commands** menu. This brings up a dialogue offering a choice reporting all tagged bases or simply all bases with base-calls different to the reference sequence. We can also sort the output sequence by sequence or position by position, depending on whether we wish to know details about a particular patient or are looking for SNPs at a particular site. For now just accept the defaults and press **OK**. It may appear that nothing has happened, but if you find your main **gap4** text output window (the first one that came up when you started **gap4**) you should see the following.

```
000256_11cF 2196G>R (expressed D>[ND]) (strand - only)
000555_11cF 2196G>R (expressed D>[ND]) (strand - only)

000256_11cF 2201C>T (silent S) (strand - only)
000416_11cF 2201C>T (silent S) (strand - only)
000437_11cF 2201C>T (silent S) (strand - only)
000465_11cF 2201C>Y (silent S) (strand - only)
000555_11cF 2201C>T (silent S) (strand - only)
002461_11cF 2201C>T (silent S) (strand - only)

000256_11cF 2430T>C (silent L) (double stranded)
000416_11cF 2430T>Y (silent L) (double stranded)
000437_11cF 2430T>Y (silent L) (double stranded)
000465_11cF 2430T>Y (silent L) (double stranded)
000555_11cF 2430T>Y (silent L) (double stranded)
001321_11cF 2430T>Y (silent L) (double stranded)
002461_11cF 2430T>Y (silent L) (double stranded)

000256_11cF 2731C>T (expressed P>L) (double stranded)
000416_11cF 2731C>Y (expressed P>[PL]) (double stranded)
000437_11cF 2731C>Y (expressed P>[PL]) (double stranded)
000465_11cF 2731C>Y (expressed P>[PL]) (double stranded)
000555_11cF 2731C>Y (expressed P>[PL]) (double stranded)
001321_11cF 2731C>Y (expressed P>[PL]) (double stranded)
002461_11cF 2731C>Y (expressed P>[PL]) (double stranded)

000256_11cF 2788G>T (expressed G>V) (strand - only)
```

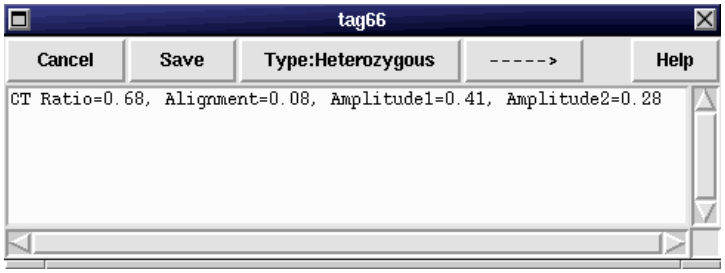
Let us look at the first line in detail.

```
000256_11cF 2196G>R (expressed D>[ND]) (strand - only)
```

000256 11cF is the patient sample (actually this represents both the *F* and *R* sequence as both are considered here). Position 2196 in the aligned reference sequence is base *G*, but in the patient sample it is a mix of bases **A** and **G** (DNA ambiguity code *R*). This is the first base of codon **GAC** which codes for amino acid **D**. Mutating it to AAC yields amino acid **N**. So our mix of **A** and **G** gives either amino acid **D** or **N** depending on the copy of the gene being expressed. This is reported as *expressed D>[ND*]. Finally note that this mutation has only been observed on the reverse strand sequence.

2.2 Tags / Annotations

The Report Mutations function used the coloured tags to identify the mutations. These were automatically added by the **pregap4** modules. Trace Difference produced the orange tags ("Mutation" tag - "MUTA" in short form) short) by computing the difference traces and detecting peaks above and below the base line. Heterozygote Scanner produced the red tags ("Heterozygous" tag - "HETE" in short form) by searching for superimposed peaks.



Move back to the editor window and right-click on a red HETE tag and select the **Edit Tag** command from the popup menu. A new window appears showing the tag type and a text comment, which may be edited to include any details you wish. The very start of the tag will be two base calls,

representing the mixture present (eg "CT").

Cancel this window and try creating a new tag by right clicking on another base-call and selecting **Create Tag**. Click on the **Type:comment** field and scroll down the list to select **HETE** and press **OK**. Now type a comment, remembering to start your comment with two adjacent base calls representing the mixture (e.g. "AG My comment."). Then press **Save**. Run **Report Mutations** once more to verify the new output.

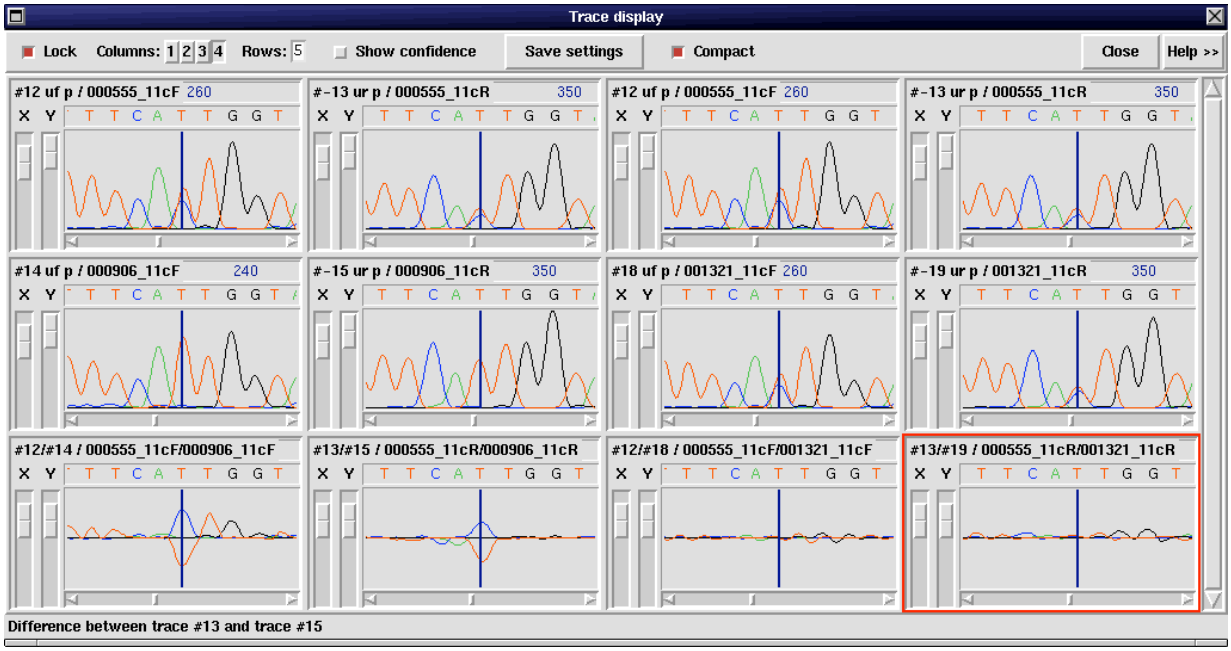
Right click on an orange MUTA tag and select **Edit Tag** again. The format of the first line is slightly different (eg C->T) representing the reference sequence base-call and the patient base-call.

2.3 Positive Controls

We may be looking for a specific mutation and not just any difference to the reference sequence. So far we have used the reference traces as a negative control; a trace in which we expect there to be no mutation. **gap4** also allows us to set sequences to be positive controls.

Right click on the sequence name **001321 11cR** in the left side of the editor. You will get a popup menu containing, amongst other things, **set as reference trace**. Select this, select **Positive control** and press **OK**. Do the same for **001321 11cF** too.You should now see a lowercase *f* and *r* next to these two sequence names. (Notice that this menu also allows you to change the negative control traces should you wish to pick something different than 000906 11cR and 000906 11cF.)

Now try double clicking on another sequence at a site of a known mutation to view the traces, such as position *2731* in sequence *000555 11cF*. After a short pause you will now see 12 traces arranged in a 4x3 grid. The leftmost two columns are the same as before (top and bottom strands compared against the wild-type / negative control reference traces). The rightmost two columns contain the top and bottom strand patient traces compared against the positive control. The difference traces for the positive control should be flat indicating that there is a match between the patient and the positive control.

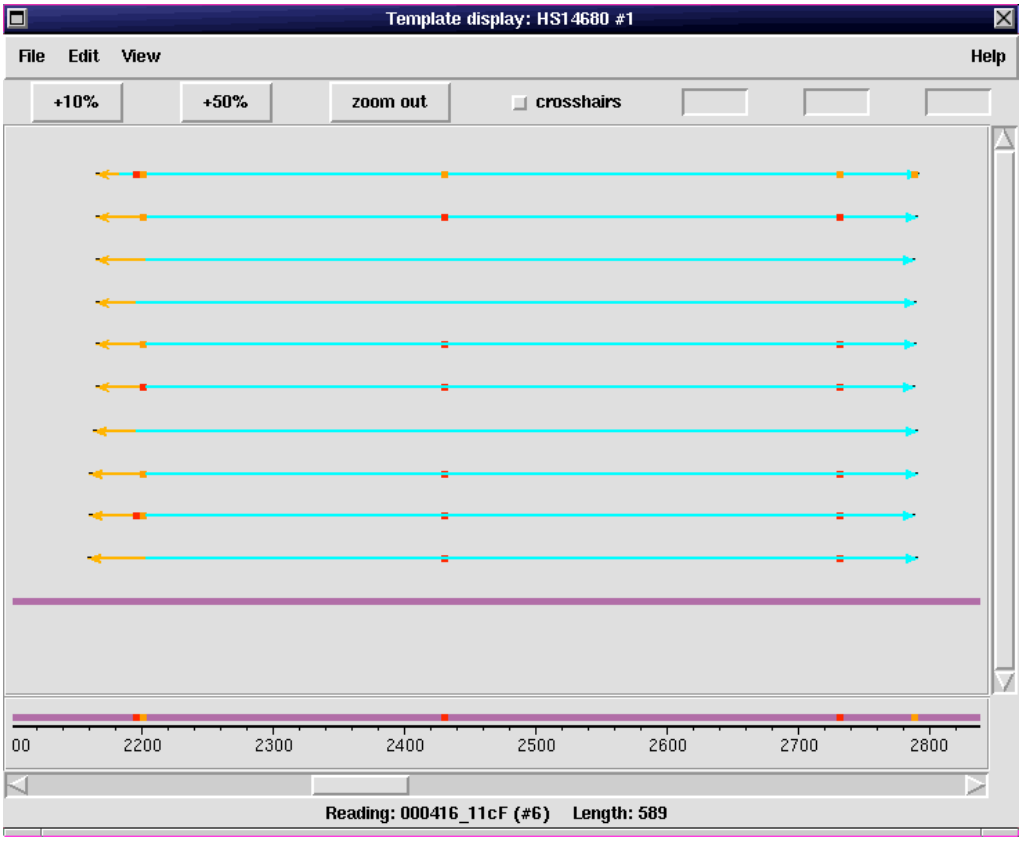


Try the same again, but this time for sequence *000256 11cF*. Here you can see this patient is different from both our negative and positive controls.

Finally, before exiting the editor use the **Settings** menu and select **Save Settings**. This will now remember the various options (such as *Group Readings by Templates*) you have enabled. Exit the editor by pressing the **Quit** button in the top right and save any changes.

2.4 Template Display

To get an overview of your sequences try the *Template Display*. Right click on the contig line in the **Contig Selector** and select **Template Display**. You'll see a graphical representation of your sequences, starting with lots of short lines with arrows on their ends at the top (representing our patient sequences), a longer line underneath this representing the reference sequence, and finally at the bottom (with numbers next to it) is the consensus sequence.



Try zooming up by pressing the **50%** button a few times until the patient data more or less fills the screen.

You will see lots of coloured markers on the sequences indicating the positions of the mutation tags. Try double clicking on one of these tags and you'll get a contig editor launched at that position. Try to rearrange the windows to see the template display and as much of the contig editor window as possible. You'll see that there is now a vertical cursor in the template display.

This indicates the position of the editing cursor in the contig editor. Try clicking on the vertical line in the template display and dragging it left or right. You will see that this scrolls the editor.

As you'll have seen in the introductory talk, the template display is particularly useful when you are dealing with multiple exons and/or multiple sets of primers as it provides an easy visual overview of the sequence alignments against a genomic reference sequence.

2.5 Exiting

Finally, shut down **gap4** by going to the main **gap4** text output window and selecting the **File** menu and **Exit**.

3 Discussion Of Mutation Data Processing Methods

The original version of these methods was described in James K Bonfield, Cristina Rada and Rodger Staden, *Automated detection of point mutations using fluorescent sequence trace subtraction, Nucleic Acids Res. 26, 3404-3409, 1998.*. The more recent work has been done by Mark Jordan and James Bonfield.

At present **pregap4** and **gap4** clearly show their primary usage in the field of genome assembly, but versions tailored to mutation studies can be created once the requirements are agreed.

The automatic mutation and heterozygote detection programs work well on all the test data we have but now they require evaluation by external groups. Such analysis would enable us to improve the algorithms and to tune their parameters. At present we know that sometimes a base will be declared both as a mutation and as a heterozygous position when visual inspection shows that it is one or the other.

There is still much that can be done overall to improve the methods, but the text above summarises their status in July 2002. Although currently valuable for real scientific and clinical work they should perhaps be viewed as prototypes.

DPJ, JKB, PDFJ 2006.05.17