FPC V4: User's Manual
FingerPrinted Contigs

C. Soderlund
The Sanger Centre
Cambridge, UK
cari@sanger.ac.uk

Technical Report
SC-01-99
January 1999

Notice

If you use FPC results on a poster or slide, please reference FPC, C. Soderlund, The Sanger Centre on the poster or slide. If FPC results are in a manuscript, please reference this manual or either of the two following papers:

Soderlund, C., I. Longden, and R. Mott. FPC: a system for building contigs from restriction fingerprinted clones. *CABIOS*, Volume 13, No 5, pp. 523-535.

Soderlund, C., S. Gregory and I. Dunham. Sequence ready clones. In Bishop, M.J. (ed) *Guide to Human Genom Computing*, Academic Press.

NB This software is run regularly on Dec Alphas using data generated by the Coulson et al fingerprinting method. We do not necessarily use all function provided. Therefore, there may be problems I do not know about. If you tell me about them, I will fix them. In fact, I feel it is a courtesy to me that you let me know about problems (and results) in exchange for obtaining this software for free.

A. Overview

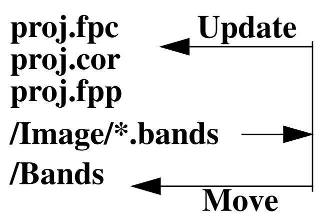
FPC (fingerprinted contigs) is an interactive program to create contigs from fingerprinted clones, view associated markers with the clones, and select a minimal tiling set for sequencing. This system is the replacement for contig9 [Sulston *et al.* 1988] which was used for the mapping of *C.elegan*. The success of this project [Coulson *et al.* 1986] has led us to scale up this technique for large scale fingerprinting [Gregory *et al.* 1996]. FPC supports scaling up of fingerprinting by providing advanced features for editing, viewing, positioning and verification. The system is built on the ACeDB graphics library [R. Durbin and J. Thierry-Meig, personal communication].

The Sanger Centre uses a program called Image [Sulston *et al* 1989] to scan and process the autoradiographs of the clones which are processed by restriction digest and isotopic labelling to yield mixtures of radioactive DNA fragments and are separated on thin polyacrylamide gels. These fingerprints use a fixed tolerance (i.e. two bands may be the same if they are within a constant tolerance, regardless of the size of the band). The output of Image is a file per gel which contains the set of bands for all clones on the gel; these files, called band files, are input to FPC. FPC can also be used for data with variable percent uncertainty or a file of uncertainty values (see §A.2) as is often used with complete digest fragments run on agarose gels.

To accompany this document is a <u>User's Guide</u> found at http://www.sanger.ac.uk/Users/cari/ffpc_aq.shtml. It will be updated for any future releases and in response to additional FAQ's.

A.1. FPC files and data flow.

The following document, the word "proj" should be replaced with your project name. An '*'indicates any string of characters. All FPC files and sub-directories shown below are in a project directory.



The FPC files are:

- proj.fpc is created, read and updated by FPC. It contains all clone, marker and contig information. It uses a format similar to ACeDB. This is the only file necessary to view the contigs. This file and read and written using the Save .fpc and Read .fpc functions. See §D.
- proj.cor is created, read and updated by FPC. It contains all the bands. It is in an unreadable format. The file is necessary to view fingerprints and run most of the analysis routines.
- proj.fpp is created, read and updated by FPC. It contains configuration information (see §F). If the file is not present, FPC will use the default configurations. The user never explicitly reads or writes this file.
- *.bands files contain newly created fingerprints along with the clone names. The files should be placed in a directory called /Image. The FPC function Update .cor reads all files in the /Image directory which contain the suffix ".bands"; it updates proj.fpc and proj.cor with the new bands, and then moves the files to /Bands where they can be subsequently deleted by the user. See §D and §L.

- /Gel (optional input) contains image data files (the format is not specified in this document).
- *.ace (optional input) a file of markers along with associated clones can be dumped from ACeDB V4, and subsequently merged with FPC. The marker file can also be created by SAM V2.5a [Soderlund 95, Soderlund and Dunham 95]. See §D.3.
- /Pace (optional output directory) Contigs with clone selected for sequencing can be saved in pace files which are put in this directory. These files are loaded into Pace, which is the Sanger Centre data tracking database (see Simon Dear for details). See §D.3.
- /Ace (optional output directory) Contigs can be saved in ace4 format; these files are put in the Ace directory. The ace models used are from the chromosome 22 database (http:ftp.sanger.ac.uk/pub/human/chr22, see Ian Dunham for details). See §D.3.
- *.gram (optional output) A file of clones and bands can be written in GRAM format for a contig so that a detailed restriction map can be created [Soderlund and McGavran 94, Soderlund and Burks 94]. See §J.3.

A.2. Notes for External Users

The demo2.tar contains a proj.fpc, proj.cor, an /Image directory with two bands file.

FPC expects all clone names to be a maximum of 12 characters and markers are a maximum of 16 characters long. FPC is NOT case sensitive. A maximum of 200 bands (or fragments) is allowed per clone. A maximum of 1000 markers per clone are allowed.

The Update .cor function automatically maps clones to clone types based on the first letter of the clone name, as follows: ‘b’ maps to BAC, ‘d’ maps to PAC, ‘c’ maps to cosmid, ‘f’ maps to fosmid, and ‘y’ maps to YAC.

In order to use FPC with variable uncertainty (e.g. complete digest run on agarose gels), select Configure from the main window to set this option. See §D.3 for more information.

For all ace files, the C routines are in separate files so that they can be easily altered for different laboratories (i.e. ace_merge.c, pace_save.c are in the fpc/file directory).

Marker data is used in the analysis as follows: (1) A markers hybridizing to one or more clones in a contig is positioned over the deepest stack of clones. Therefore, the salient positioning is on the fingerprint, but possible problems can be elucidated by the markers. (2) Markers are used to determine neighbouring CBmaps by the OKALL routine. (3) Framework markers can be read in from a file (see Merge .ace file) and order of the contigs based on the framework markers can be display via the Project window.

B. Startup

To execute FPC, type the following (<> indicates optional arguments):

fpc <project name> <ctg number>

If you start up FPC with no arguments, the Main Menu will appear with no project name. Select Load .fpc. The file chooser window will start up; it will show all the directories of your current directory (these will be suffixed with "/") and all the files which are suffixed with ".fpc" (this suffix will not be shown). Double clicking a directory name will cause the file chooser to move to that directory and show the files as just described. Clicking on a file name will write it into the yellow box beside the label File. Select OK and the project will be loaded.

If you are in the project directory, you can specify the project name on the command line and it will automatically be loaded. If you provide a contig number, it will be automatically displayed.

When specifying a file, do not include the suffix ".fpc" as FPC will automatically append it.

Beware - If fpc does not show colours correctly, you have too many applications running. You need to remove other applications. Alternatively, you can start fpc with a "- install" option; this installs its own colour map so the colours are correct but your entire screen becomes funky when moving the mouse around

❖ Starting a new project

The following explanation assumes an understanding of FPC; so if you are a first time user, you should read through this manual while trying the functions on the demo2.fpc file. To start a project:

- Create a project directory, create a subdirectory in it called Image, and put your *.bands files in the Image directory.
- Under File..., select New Project; you will be prompted for the name.
- Execute Update .cor which will read the clones and bands and update your FPC files.
- Under Analysis, select Build Contigs. Edit them, etc. The User's Guide gives more information on how to proceed (see page 3).

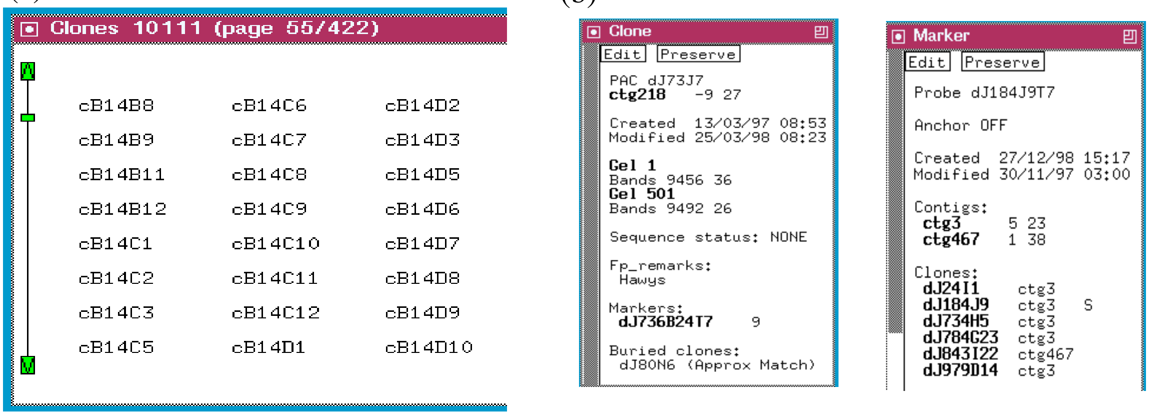
C. Windows, Entities, and Hotspots

The main entity is the project and all other entities belong to the project. There are currently three other types of entities: contigs, markers and clones (these are often referred to as *classes).* A contig contains one[1] or more clones and zero or more markers. A clone can only belong to one contig. Clones that do not belong to a contig are called *Singletons.* Markers must have a positive hybridization to one or more clones. There are six types of windows:

- Command windows - Main, Analysis, Configure, Edit Map, Merge.
- Keyset - There are keyset windows for contigs, clones, markers. These may contain all the entities of the class or just a subset of them.
- Text - There are text windows for the project, contigs, markers, and clones.
- Contig display - horizontal contig and vertical contigs (sometimes referred to as *maps*).
- Bands - fingerprint, gel image, and consensus band map.
- Edit - for editing clones and markers

1. A contig is defined to be two or more clones, but it is easiest for various reasons to allow contigs of one.

(a) (b) (c)



(a) Clone Keyset (b) Clone Text (c) Marker Text window.

✧ Rules about windows and hotspots

- Only one Keyset window can be active. A new Keyset replaces an existing one. A Keyset window for a single entity is never displayed; the text or contig for the entity is displayed instead.
- Only one active clone text window and one active marker text window can be open; an active text window has hotspots. The Preserve button will make an inactive copy of a text window which does not have hotspots and will not be replaced; it must be explicitly removed.
- Only one edit window can be active at a time regardless of its type. The user must accept/reject the edit (which exits the window) before editing a new entity.
- Only one contig window can be open.
- Only one of each type of bands window (fingerprint, etc.) can be open.
- A boldface name on a window is a hotspot; it can be double clicked to view the entity. Also, all the items in a keyset window can be double clicked to view the entity.
- If a window gets hidden, click on its hotspot or command to bring it to the front.

✧ On menus and windows:

- A black circle is an on/off switch; when it is filled in, the corresponding feature is on.
- A stop/go button turns red when selected. While it is red, any clones selected are used as input to the function. Turn the function off by re-selecting the button. If a stop/go button turns light-red on selecting it, the function automatically turns off after one clone selection.
- Pull-down menus are accessed by pressing the right mouse button and pulling down. (1) Doing this in the white space of a window results in a menu with the options Print and Quit; and occasionally, other minor stuff. (2) On a contig display, the clone names and clone interval lines have pull-down menus. (3) A button with a "..." in the label is a pull down menu.

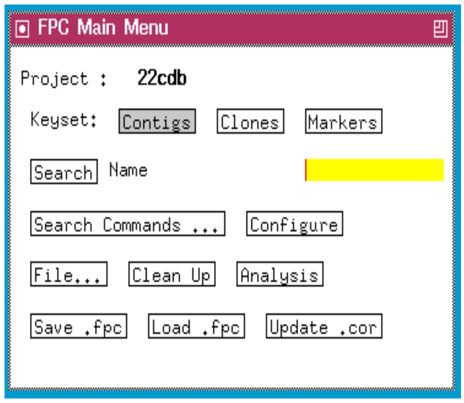
✧ Drag and drop

You can drag the mouse over text on stdout (your terminal window) and drop it in any of the FPC yellow text boxes.

✧ Quit

Some windows have Quit buttons; most windows have Quit and Print on the pull-down menu in the white space of the window.

D. Main Menu



Load .fpc A file chooser window will be opened; it will list all files that have a ".fpc" suffix. Once a file is chosen, the prefix (i.e. project name) will be listed beside the Project label (no other options will work until a project is chosen).

Save.fpc Writes all data to your proj.fpc file. A backup file will be made before the new file is written. FPC will ONLY write this file at your request (or on Update .cor as described below). It is recommended that when you do one or more changes that you want to keep, do a save; then, if there is a system crash, your work is not lost. If someone else has written the file since the file has been loaded, it will not over-write the file; instead, it writes a file proj.userid.fpc where userid is your userid. (See Lock in §D.3 for exclusive access).

Select Update .cor to update the proj.fpc and proj.cor. This command looks in your /Image directory for all files called *.bands (where * is any prefix). Each file will be read, the bands concatenate to your proj.cor file, the clones added to your proj.fpc file, and then the file is moved to your /Bands directory.

When a clone is added to the proj.fpc file, it is stored with the clone name, the gel number, the number of bands, and the position of the first band in the proj.cor file. Referring to the Gel entry as the {gel number, number of bands}: If a clone name is already in the proj.fpc file with a Gel entry, FPC will do the following: it checks to see if the gel number is not equal to 0 and if the new Gel entry is equal to an existing gel, it assumes that this file has already been loaded and it terminates the reading of it. If the Gel entry is unique for the clone, it will be added. A clone can have any number of gels (see §G and §H).

Clean Up: removes all but the main window.

KeySet: <class> where class is Contigs, Marker, or Clones. Selecting a KeySet button will turn it grey, making it the current class. A double click will list all entities in a keyset window. The Search Command will be applied to the current class.

NB Contigs can be listed in either the KeySet window or the Project window (see §E); the keyset window is good for listing a subset of the contigs, the project window contains more information. Contig numbers are hotspots on both.

D.1. Search Commands

Selecting Search Command (with the left button) will bring up a window of searches appropriate for the current class; using the right button to get a pull-down menu gives the 'most often used' subset of commands. If there is no keyset for the class, the search will be applied to all entities of the class. *If a keyset window is already open for the class, the search command will search the entities in the keyset. In this way, you can create subsets of subsets. To reset the keyset to all entities, click on the Keyset <class> button.*

For class Clone: For commands requiring an argument, double click the command and the search name will go in the main window. Enter the argument and either hit carriage return or select the Search button. The argument type is in {brackets}.

- Name {substring}- all clones with the substring (see below) in their clone name.
- Remark {substring} - all clones with the substring in the remarks.
- !Remark {substring} - all clones without the substring in the remarks.
  NB A clone has two types of remarks, "Remark" and "Fp Remark". Both of the Remark commands search for the substring anywhere in either type of Remark.
- Before creation date {day/month/year hour:minute}
- After creation date {day/month/year hour:minute}
- Before modified date {day/month/year hour:minute}
- After modified date {day/month/year hour:minute}

e.g. 28/2/96 00:00. If there is no hour:minute, the default is 24:00. Use 00:00 to include the day.

- Gel {integer} - all clones that belong to the gel specified.
- Fingerprint {name} - all clones that match the fingerprint for a clone.
- Contig {integer} - all clones that are in the contig specified.

The following search commands do not require an argument. Each will execute immediately after you double click it. All searches are carried out on the current keyset except for Cancelled.

- Singletons - clones that do not belong to a contig.
- Multiple Fingerprints - clones with more than one {Gel/Bands} data.
- No fingerprints - clones with no {Gel/Bands} data.
- Selected - clones that are selected in the current contig.
- Cancelled - clones that have been cancelled. Cancelled clones are stored in the proj.fpc file with a "!" and are ignored (see the clone edit functions).

For class Marker: Name and Dates as described above are available and the following search commands which do not take arguments:

- Multiple Contigs - a marker is in more than one contig
- No Contigs - a marker is in no contig.

For class Contig: only the Name option is available.

✥ Specifying a substring

A "*" at the beginning of a string will select all entities with the suffix specified. Likewise, a "*" at the end of a string will select all entities with the prefix specified. Asterisks can be specified at both the beginning and end of a string, but not within it. For example, when the <class> is Clone, a search string of *11* will display all clones that have this string anywhere within the clone name. If the string is only an "*", it is equivalent to selecting Keyset <class>. The search for the substring is NOT case-sensitive.

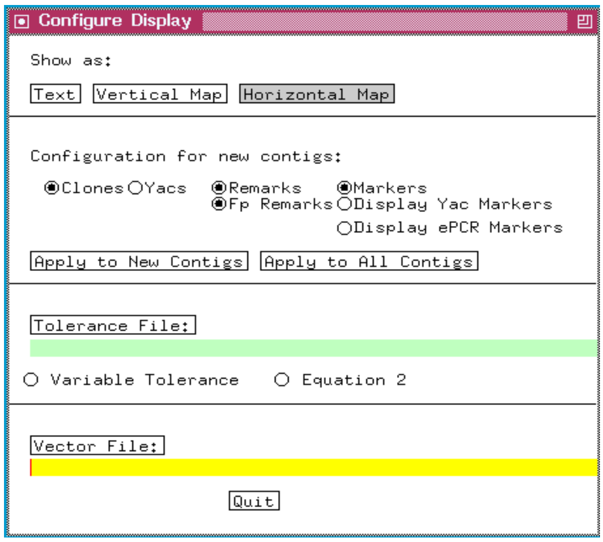
❖ Add Fp Remark and Remove Fp Remark

On a clone keyset window, the pull-down menu (in white space) has an option to:

Add Fp Remark: Enter a word to be added as a Fp Remark to all clones in the keyset.

Remove Fp Remark: Enter a word. All clones 'in the keyset' are searched and each Fp Remark that is the same as the input word is removed.

D.2. Configure



Show as: A contig can be displayed as a vertical map, horizontal map, or text object; select the type of desired display. All subsequent contig displays will be according to the selected type.

Configuration for new contigs: Each contig can be configured to not display one or more of the following types: Clones, YACs, Remarks, FpRemarks, Markers, ePCR and YAC Markers. These are configured per contig by the Configure command on the contig window (see §F.2 and §F.3) and are stored in the proj.fpp file. This menu sets the default for new contigs; i.e. every time a new contig is created, it uses these defaults. To apply them to all contigs already created but not used, select Apply to New Contigs. You can change the defaults on all existing contigs with the Apply to All Contigs. (Bug: Does not work properly for a displayed contig).

The Tolerance File and Variable are for projects that have a tolerance (i.e. uncertainty) based on the size of the band (i.e. fragment). If the uncertainty varies with the length of fragment, a file of values can be entered. An example file is:

```
14000 0.0363
6400  0.0198
1100  0.021
600   0.0378
```

FPC will use linear interpolation (the format and the interpolation are exactly as used in GRAM).

Alternatively, turn on the option "variable"; wherever the tolerance parameter appears in FPC (i.e. the fingerprint and analysis windows), the number entered will be used as a percent of the band size (e.g. if 3 is entered, the uncertainty will be 0.003 * band-size).

The file name or variable flag only have to be set on the first time as they are stored in the fpc file. When changed, the change takes effect immediately for all subsequent analysis operations.

Equation 2 uses an alternative equation for comparing clones [Soderlund *et al.* 1997]. Toggle the equation 'on' to use it.

The Vector File holds the band values of vector bands that should be ignored. The Vector file has a simple format of the type of clone followed by the band values, one values per line. Multiple sets of clone type and vector values may be specified,. e.g.

```
BAC
10
100
350
PAC
50
75
400
Other
40
550
1300
```

The tolerance used on these bands is the value listed in the FPC Analysis window. 'Other' can be used to specify a none specified clone, i.e. type Clone. See §A.2 for our naming convention to typed clones. NB The file is read and tolerances computed at the same time that the .fpc file is read, i.e. a change of tolerance in a current session will not affect the vector tolerance.

D.3. File...

NB Most of these commands are not relevant to the new user except for 'Create New Project. Plus, most commands are specific to the Sanger Centre dataflow.

❖ Merge .ace file

(1) A file generated from SAM as an .fpc file but renamed with an .ace suffix can be merged; markers are added to FPC if they hybridize to one or more clones in the FPC database. The markers are added to clones if they do not already exist. No markers are deleted on a SAM merge. The marker order is used as the framework and the markers become anchored. If multiple files are loaded, insure that the Start and End given in SAM for the different files do not overlap. The description of the format is given in File Formats, *note that this format could be generated from some other mean in order to enter markers and/or frameworks.*

(2) Alternatively, an ACE dump can be read. The following description is what we use at the Sanger Centre [Soderlund *et al*. 1997b]: This command reads in a file of markers dumped from an ACeDB V4 database. This function does the following: for each clone in the file, if the clone name exists in the FPC database, it reads the list of markers. If it finds a marker that is not in the database, it adds the new marker. Otherwise, it just adds it to the clone if it does not exist for the clone. Markers can have a value of "weak"; e.g. Positive_probe or Positive_probe_weak indicate a strong versus weak signal, and are highlighted in the display accordingly. When the file is saved, the markers become permanent data in the FPC database.

After reading the marker file, it looks for the following two files (this is a bit of a kludge): it looks for a file name 'framework'; if found, this file is read to get the positions of the framework markers. Additionally, the map name is found and is thereafter used as a prefix for the map name in ace files. Then it looks for a file called sequencesnew.ace, and if it exists, it reads it to update the sequence status of the clones. See file format.

✥ Submit for sequencing

Should be executed when clones are ready to be sent for sequencing. Submit will request a contig to submit; a pace file will be written (called proj-ctg#.ace where proj is the project name and # is the contig number). The file will be written into a sub-directory called Pace (it will be created if it does not exist). The pace file contains all clones with the sequence flag set to a value other than NONE; once submitted, this flag will change from TILE to SENT.

✥ Save as Ace

This command provides a menu with the following 3 options: Save All Contigs, Save Sequenced Contigs, and Contig followed by a text box for the contig number. The corresponding contig(s) are written in an ace4 formatted file. The files are written in the Ace directory with names proj-all.ace, proj-seq.ace, and proj-ctg#.ace, respectively. (SC data flow: If a map name has been read from a framework file, the Save Sequenced Contigs option first writes the necessary ace commands to remove all maps with names ‘mapname || ctg#’ from ACeDB).

✥ Save Contig as FPC

Saves a contig in a new or existing FPC file. This file must be used with the original .cor file if any analysis is necessary (see §L for a way to reduce the original cor file to the existing set). This is not user friendly as you could write the same contig twice.

✥ Un/Lock write access

Executing this command will give you exclusive write access; i.e. no one else can write to the file. The words WRITE LOCK will be next to the project name when you have exclusive access. Remove the lock by executing this command again or by exiting FPC.

In summary: When a project is saved (i.e. Save .fpc) two checks are made, firstly that the project is not locked by another user and secondly that the project has not been saved by another user since you started. If the checks fail, the project is saved as proj-userid.fpc (where proj is the project name and userid is the user’s id). So if you are doing many edits, make sure to lock the database; you may save any number of times before unlocking it.

✥ Merge Projects(not tested in V4)

This command takes two projects and merges them into a third directory, as follows: (1) Load your first project to be merged. (2) Click on this command. (3) You will be asked for a project name; it will create a new directory; e.g. if you are in directory /nfs/cari/demo/x1 and you give a directory name x2, it will create a directory /nfs/cari/demo/x2 and the files x2.fpc and x2.cor. (4) It will ask you for the second file to merge. Then it will execute. (5) If the same clone is in two different contigs (from the two projects) it will not be put in either contig; instead, it will be given an Fp-remark of ‘Merge’. Search for all such remarks and resolve these conflicts. It is a good idea to give the various projects Fp-remarks (given to all clones in a keyset as specified in the previous section) before merging so that you can later tell what is from where.

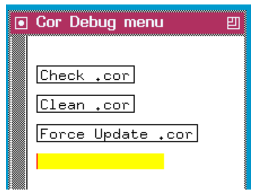
❖ Create New Project

To start a new project, select this command, then Update .cor to read in your first set of bands.

❖ Load contigC

Loads a contigC ace file (this command is probably obsolete).

❖ Check Cor



This set of functions were kludged together to fix some cor files that got corrupted (though we have not had a problem in years). They are included because they can be useful.

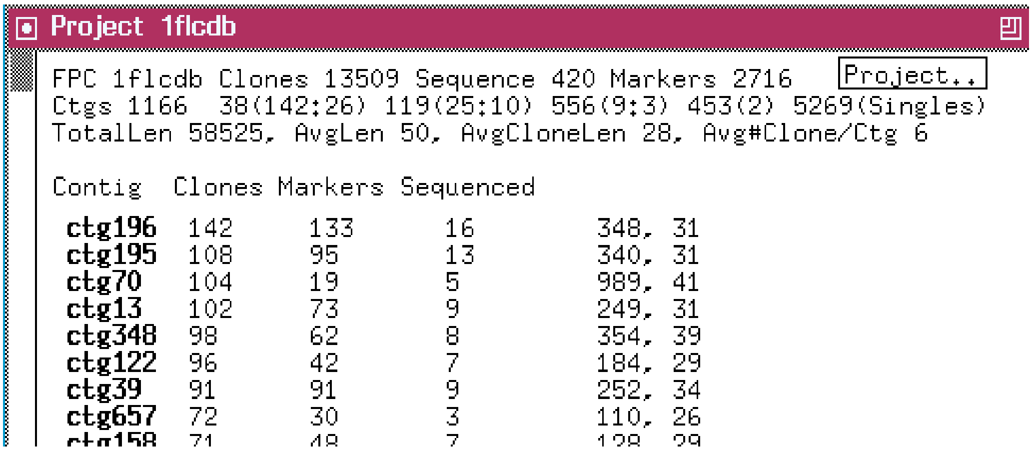
Check .cor tell you if there are any gaps or overlaps in coordinates pointing in the cor file; overlaps are serious.

Clean .cor gets rid of any bands that are not referenced by a clone in the corresponding fpc file. This is useful if you have physically removed clones from your fpc file, or, if you have used the Save Contig as FPC, you can make a corresponding cor file by make a copy of the original and then having it cleaned.

Force Update will add any bands file in the Image directory without checking to see if its already but added; i.e. you can overwrite an existing gel entry for a clone.

A clone name can be put in the yellow box and upon <CR>, the bands for the clone will be written to standard out.

E. The Project Window



On the main window, the project name is a hotspot; click it to get the project status window. The window contains a pull-down button which has options for displaying the contigs in various orders and displaying messages on the right side of the window. The options are:

Sort by contig: the contigs are listed by ascending order of contig number. There is no message computed for this option; though a message from a previous option will remain.

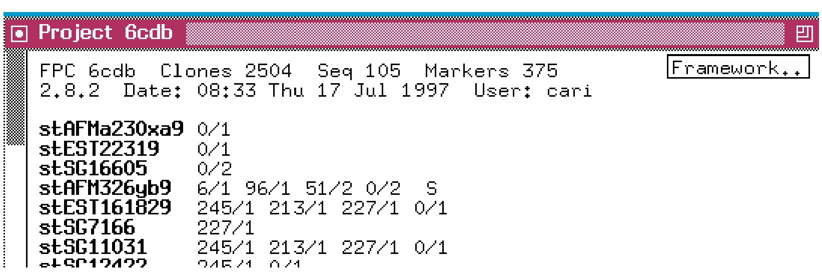
Project totals: as shown above. The second line gives the number of contigs for each of the following groups: greater than 26 clones, between 25 and 10 clones, between 9 and 3 clones, 2 clones and singletons. The contigs are sorted by number of clones. On each contig line is the contig length in bands and the average clone length in bands.

Sort by Length: If fixed tolerance, the length is the number of bands in the contig multiplied by 3; our estimates for the average size of band is 3000, so multiplying by 3 gives the length in KBs. If variable tolerance, it computes the average length fragment per contig and multiplies that by the number of bands.

Clone counts: show the number of buried clones, cosmids (c), BACs (b), PACs (p), fosmids (f), YACs (y), and other; the small letter will be next to the number of each type, e.g. c160 is 160 cosmids. The contigs are ordered by size groups as described for project totals.

By Sequence: contigs with sequenced clones will be displayed first and the number of clones of Tile, Sent, Ready, Shotgun and Finished are listed for each contig.

By Keyset: the contigs which contain one or more clones in the keyset will be listed first. This only works when the Keyset contains clone names.



Framework list the markers in order and their respective contig.

SC summary: shows clone and band totals.

F. The Contig Map

The contig can be displayed as a vertical or horizontal map (see §D.2). The horizontal map has the display of markers and YACs which the vertical map does not have. It also has the capability of merging two contigs. Otherwise, they have the same functionality.

F.1. Buried, canonical, sequenced clones, and singletons (ctg0)

Buried clones is a concept inherited by contigC; if a clone shares all (or most) of its bands with another, it can be buried within it. A clone that has one or more buried clones is called a canonical clone (canonical/buried relation is often referred to as parent/child). A canonical clone can not be buried within another. If a buried clone does not have a position, by default, the position is the parents left coordinate plus the length of the clone. When the canonical is moved, its buried clones are automatically moved. When a marker is selected on a map, if the marker is positive for a buried clone and not positive for the parent, the parent is highlighted bright green.

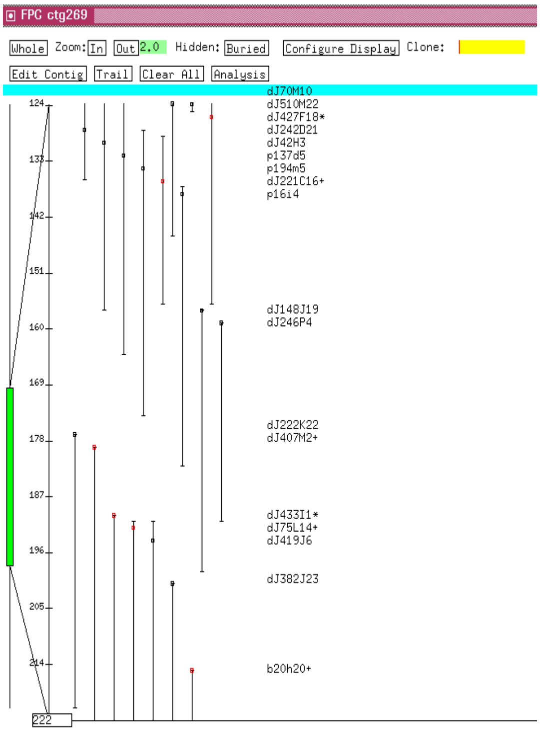
Clones can be buried with a status of exact, approximate, or pseudo where exact has all the same bands as the parent, approximate has a user-defined percentage of the parent bands (see Analysis), otherwise, the clone is buried as pseudo. The pseudo status allows poor clones or clone in areas of great depth to be buried, hence, only the best clones are kept for display. On map displays, the canonical/buried clones have a character following their names as follows:

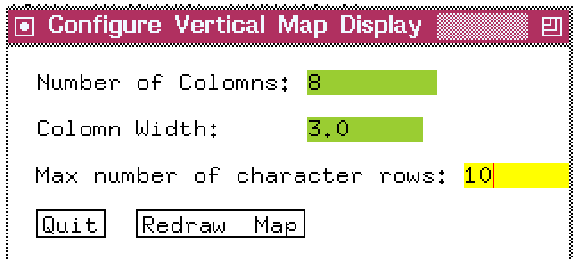
* canonical with exact or approx
\+ canonical with pseudo only
= buried exact
~ buried approximate
\# buried pseudo

Clones have a tag that indicate their sequenced state; by default, this tag is NONE. When picked for sequencing, the user changes it to TILE (see §H). This tag is changed to SENT when submitted for sequencing (see §D.3). For the Sanger Centre, when the dump of Pace is read, the state of the clones picked for sequencing is updated. The state can be shown on the Project window By Sequence (see §E). The Contig Analysis window Show sequenced command colours the clone according to sequenced state (see §J.1)

Singletons (unattached clones) are in contig 0. This is not shown on the clone text window; i.e. it is not obvious. The easiest way to remove a clone from a contig is through the Contig Analysis window function Remove from ctg (see §J.1) but it can also be removed through the clone editor by giving it a contig number of 0 (see §H). To remove all the clones from a contig or a region of clones, use the Edit Contig window; select all or some of the clones, then select Move/Remove and enter 0 for the contig (see §F.6).

F.2. Vertical display





Vertical Map and Configure window

The clones are shown vertically with the names to the right; a bullet (tiny round circle) is positioned on the clone lines vertical from the clone name. The bullet is red for canonical clones and blue for buried clones. If there is no bullet, then the clone name is not displayed.

The clone intervals are drawn as follows: they are sorted by left end, and then laid out in sets of *n* columns with a spacing of *w* between columns. If the *nth* set does not fit below the *n-1* set (i.e. its intervals overlap), it gets shifted to the left and gets placed between the *n-1* columns. If the coverage is very deep, these intervals may get crowded together.

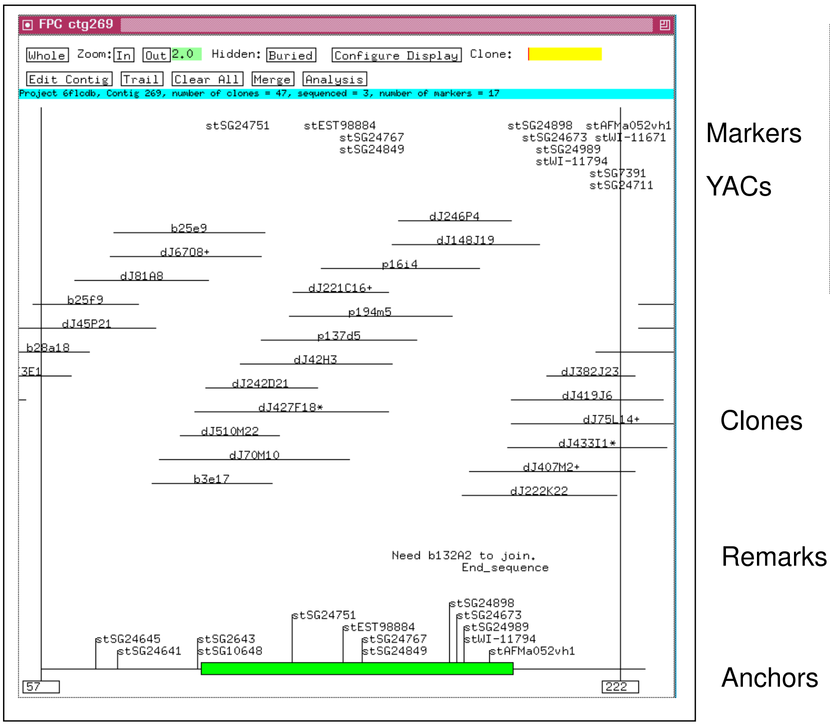
To adjust the display, select Configure Display, adjust values and then select Redraw Map. The values to be adjusted are:

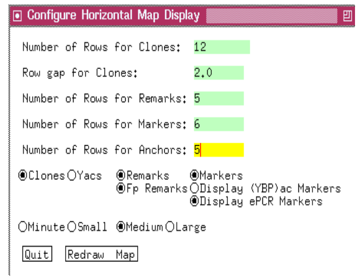
Number of Interval Columns: the *n* value as described above; default is 8.

Column Width: the *w* value as described above; the default is 5.0.

Number of Clone Name Columns: the default is 10 (if you are not seeing all your clone names, this value needs to be increased).

F.3. Horizontal display





Horizontal map and Configure window.

The display is arranged from top to bottom: Markers, YACs[2], Clones, Remarks and Anchors. Referring to the Configure window: The text boxes are similar to the ones described for vertical.

❍ There are four circles that toggle between having a bullet in the middle and not having one. When the bullet is not present, the corresponding entities are not drawn, and the corresponding text boxes are not shown.

❍ The display of YAC, PAC, and BAC markers can be turned off with the (YBP)ac circle.

❍ The display of ePCR markers can be turned on and off. They are displayed with a yellow highlighted when on.

❍ There are four circles regarding size, only one will have a bullet in it. The text on the map is drawn according to the size specified.

<u>Anchors:</u> the anchor line shows the anchored markers for the entire map; the green bar indicates the region that is being shown. You can click on an anchored marker to move to that part of the map. The markers that are anchors are decided by the user and can be set via the clone editor or are set from the framework file. The maximum depth is 20.

Beware - if your display looks weird (e.g. no clones or not all your anchors are shown, etc.), check your configurations! YACs and buried are included in defining the width of the contig regardless if they are displayed or not.

2. There are no YACs in this map. The *C.elegans* fingerprinted contigs have YACs define as intervals though they do not have fingerprints; these are correctly displayed in the YAC tier. There is no way to add YACs without fingerprints except to create a *.bands file that has the YAC names with 0 bands.

F.4. Contig display functions

There are two regions bars, each has a box on the end showing the bar coordinate. These bars can be moved by clicking on the box with the left button and dragging. When a region bar is clicked, it turns yellow and becomes the current bar; the current bar can be moved with the arrow keys. Anything that specifies ‘region’ indicates all clones contained partially and totally within the region bars.

Zoom: Select In to make the display larger; select Out to make the display smaller. The text box is the scale; by default it is 2.0 indicating that In doubles the size and Out halves the size. The value 1.1 is the lowest allowed.

Scrolling: When the display covers more than one page, use the green scroll bar to scroll. To move a small distant, click with the middle button in the contig window and the map will be centered where you clicked.

Hidden: Buried/Pseudo/None Using the left button, toggle between hiding all buried, hiding the pseudo, and showing all. Alternatively, pull down with the right button and pick the appropriate display option.

Clone: Enter a clone name followed by a <CR>, and the display will be centred on that clone. If the clone is in a different contig, the text window for the clone will be displayed.

Print: The display can be printed by: (1) place the cursor in the white space of the contig window, and press the right button; a menu will appear. While holding down the right button, drag the arrow down to the Print Screen and release the mouse button. (2) Select Execute.

Quit: Pull-down the menu found in the white space of the window, and select Quit

F.5. Current entity, friends, and Trail

Click a clone or marker, it will be coloured as follows:

- Cyan - current clone or marker.
- Dark Green - friends. The definition of a ‘friend’ is all the other entities associated with the entity; e.g. buried clones are friends of the parent canonical clone and vice versa, markers are friends of the clones they are attached to and vice-versus.
- Light Green - friend of a friend. If a marker is clicked, and one of its friends is a buried clone, the parent will be coloured light green unless it is also a friend of the marker.
- Light Red - weak friend. A clone that has a weak hybridization signal to the current marker.

Double click an entity and a window will be displayed with information about the entity. Clicking a new entity will remove the colour from the previous current entity before colouring the new entity and friend -- unless the Trail button is on.

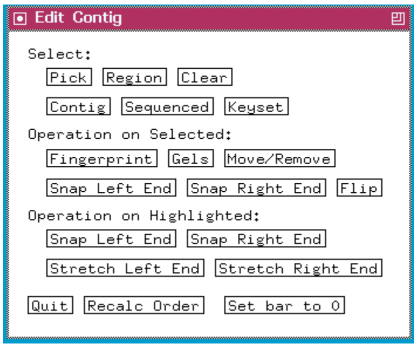
The Trail is a stop/go button. When on, the colouring of entities is not cleared when a new entity is clicked. Turn Trail off and the colours are removed.

There is *crosstalk* between the contig display, the fingerprint (fp) display (see §G.1) and/or a consensus band (CB) display (see §K).

- A clone clicked in any of these displays will be highlighted in all of them.
- If Trail is on, clones are trailed in all of the displays.
- The Contig Analysis (see §J.1) functions highlight clones in response to various evaluations; this highlighting is not removed if trail is on; if trail is off, the highlighting will be removed before new highlighting results are shown.

F.6. Edit Map and Selected clones

A *selected* clone is blue. One or more clones can be selected. Operations are performed on the set of selected clones. *NB A clone can be both selected and highlighted which means it has two colours; the most recent colour request will be the one visible.*



The Selected menu allows the user to select (1) Contig: the whole contig; Buried are included regardless if they are not displayed (2) Region: the region between the region bars; if Buried are not displayed, they are not included in the selected, (3) Sequenced: the clone selected for sequencing, (4) Pick: this is a stop/go button, and every clone selected while the button is on has its selected state toggled. (5) Keyset: selects the clones that are in the clone keyset.

There are two more ways to select clones: (6) Clones can be selected from the Analysis menu (see §J). And (7) the select state of a clone can be toggled using the Select option on the clone's pull-down menu.

Recalc Order reorders the clones; when clones are moved, added or deleted within a map, the positions of all clones except the affected ones stay the same. This makes it easier to keep track of changes, but the map can get messy, so you can request the clones to repositioned (as if the contig was newly drawn) by selecting this function.

Set bar to 0 readjusts all clone coordinates so that the current region bar is at location zero. This change is absolute; i.e. the coordinates of the clones really change but their relation to one another does not.

✥ Operations on Selected

After selecting a set, the user can perform one or more of the following:

Fingerprint displays the fingerprints for all the selected clones (see §G.1).

Gels display the gel images for the selected clones (see §G.2)

Move/Remove prompts the user for a contig number, and moves the selected clones to that contig. A contig can be removed by selecting Contig and moving it to contig 0. A contig can be moved to a different contig number by selecting Contig and moving it to an unused contig number. (See Rules in §H).

Snap left will move all selected clones so that the left most selected clone will abut the current region bar. Snap right moves them so that the right most clone abuts the bar.

Flip flips the selected clones.

❖ Operations on Highlighted

The following work on the current highlighted clone (not including trailed clones).

Snap Left moves the highlighted clone so that its left end abuts the current region bar.

Snap Right moves the highlighted clone so that its right end abuts the current region bar.

When moving a canonical clone, the children get moved by the same amount.

Stretch Left stretches the left end of the highlighted clone so that it abuts the current region bar; the right end is not moved.

Stretch Right stretches the right end of the highlighted clone so that it abuts the current region bar; the left end is not moved.

❖ Clearing colours

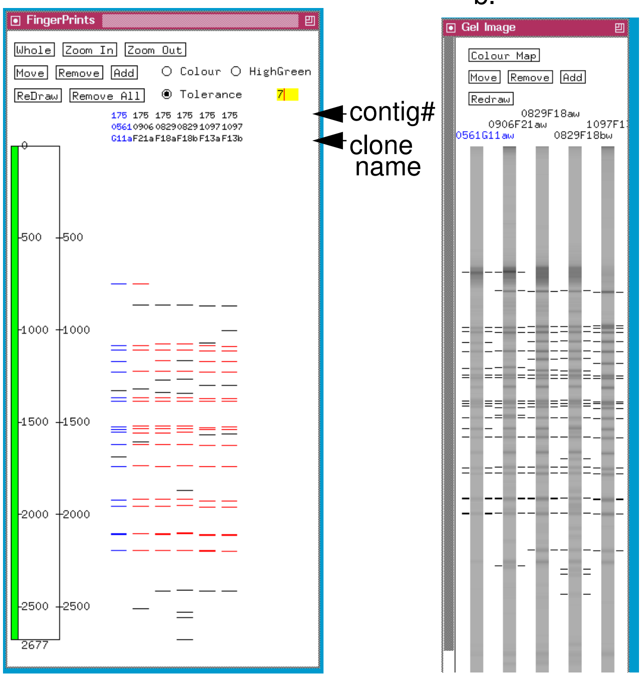
(1) The Clear option on the select menu clears the selected state of all clones. (2) Turning off the Trail clears the highlighting colours. (3) Clear All on the map window un-selects all selected clones and removes all highlighting from the map; this function is duplicated on the contig analysis menu. Clear All also clears the highlighting on a Fp display or CBmap if one or both are active.

F.7. Moving clones and markers

- When a clone is moved from a contig: If it has buried clones, the buried clones also get moved. Any markers get moved with it; if the marker is not attached to any other clone in the contig, it is removed from the contig.
- If a change is made to any of the clones attached to a marker, the markers position is recalculated. A marker's position is centered over the largest number of overlapping clones that score positive for the marker.
- A singleton can have buried clones and markers.
- A child can be selected and moved without its parent if the move is within a contig, but it cannot be moved without its parent to another contig.

G. Displaying bands

a. b.



a. Fingerprint - bands in ascending order.
b. Gel image

NB If the Gel Image (b) is reversed; i.e. it should be black on white: This is because the colour map does not work when another application is running (such as framemaker). So, if you get an incorrect image and cannot access the Colour Map, you need to remove other applications. Alternatively, you can start fpc with a "- install" option; this installs its own colour map so the colours are correct but your entire screen becomes funky when moving the mouse around.

G.1. Fingerprint (Fp) display

Fingerprints can be displayed by any mix of the following:

(1) On a contig window, with the right mouse button, click on a clone and a menu with the item Fingerprint will appear; continue to keep the mouse button depressed and pull the cursor down until it is over the word Fingerprint; let go of the mouse and the fingerprint will be drawn for the clone. The fingerprint window will be initiated if it does not already exist.

(2) From a clone text window, double click the Gel hotspot, or pull down on the Gel hotspot and select Fingerprint from the menu.

(3) As described in the section on Selected, select a set then request Fingerprint.

(4) Once the fingerprint window is active, select the Add button and as long as it is active (red), any clone picked in the contig will be added.

Fingerprints from different contigs may be displayed; the contig number is displayed over the clone name. A fingerprint for a clone may be displayed more than once.

Generally, when a clone is added to the Fingerprint display, it is added on the right end, unless a clone is highlighted on the fp display, then the fingerprint is added to the left of the highlighted fingerprint.

Move: this is a stop/go button that when on, you may move fingerprints as follows: Put the mouse pointer in the clone column (i.e. over the clone name, fingerprint or the general area) and press the left button; while keeping the button depressed, move the clone to where you want it.

Remove: this is a stop/go button that when on, selecting a clone column will remove it.

Add: this is a stop/go button that when on, any clone selected from the contig will be added to the Fingerprint window as specified above.

With these three stop/go buttons, if one button is active and another one is selected, the active one is automatically deactivated.

Zooming and scrolling: work as described for the contig window.

ReDraw: redraws the fingerprints from the current contig by order of left end. This automatically runs ReCalc Order!

Remove All: removes all graphics from the window. Use this when you want to see a new set of fingerprints, but are too lazy to quit the window and start a new one.

Quit and print screen: work as described for the contig window.

❖ **Tolerance/Colour/None**

Tolerance: When tolerance is on, the yellow editable box contains the tolerance. When a clone is highlighted (picked from the contig, Fp, Gel or CB windows), the following is performed: for all fingerprint clones, it compares the bands with the selected one and highlights all the bands that are within the tolerance specified. For the highlighted clone, all its non-matched bands are black; if the HighGreen is on, the matched bands are green, else they are blue (I cannot distinguish the blue from black bands, so I prefer green, others prefer blue, hence, the option).

Colour: When this is on, any highlighting to the contig will be reflected on this window; i.e. the clone name will take on the same colour as the clone in the contig (cyan is replaced by blue).

G.2. Gel images

Fred Wobus added an option to Image such that a trace file can be created for a gel. The file will be named <proj>.<gelno>.gel. and should be placed in your project /Gel directory. On the pull-down menu for clones, there is an item called Gel Image. Select this item and FPC will use the gel number for the clone to find the appropriate file. It will then display a window with the trace, as follows: the display will appear like the fingerprint display, except it will show this data as a grey scale with the actual bands along each side.

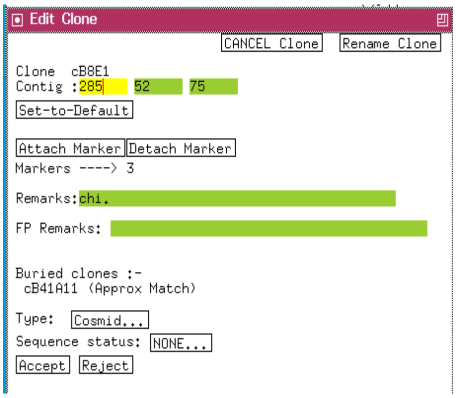
Select the "Colour map" button and a window comes up to with the grey scale to allow you to adjust the intensity.

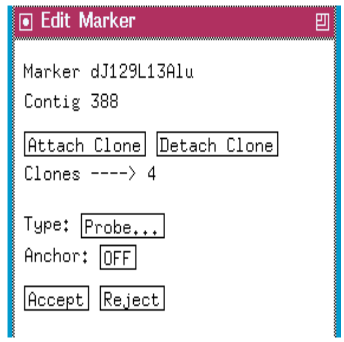
The ReDraw, Move, Remove, and Add function all work as described for the fingerprint display.

H. Editing Clones and Markers

Markers can be added to the database through the clone editor, and clones can be attached to markers through the marker editor; clones can only be added to the database via the Update .cor function. Markers are deleted from the database by detaching all its clones. Clones cannot be deleted but they can be cancelled; i.e. ignored. Moving clones and markers obey the rules in §F.7.

Marker and Clone text windows have an Edit button, which brings up the information that can be edited. Both clones and markers have Created and Modified Date fields.





✥ Edit clone.

The following describes the edits that can be made to a clone; after making one or more edits, pick Accept to instantiate them; if a contig is displayed, the changes will become visible when accepted. Select Reject to reject the changes.

Contig: there are three text boxes for contig number, left coordinate, and right coordinate. If the user changes the contig number, FPC makes the appropriate action: (1) If it is an unused contig number, FPC creates a new contig. Alternatively, enter the letter ‘n’; FPC will find the lowest unused contig number and create a new contig using the calculated number. (2) The user can move the clone to a different contig by changing the contig number. If the user moves the clone to a contig and does not give it coordinates, it will have a default position at the far left with a length equal to the number of bands. (3) The user can set the contig to 0 to make the clone a singleton.

Buried: To bury a clone, put the parent name in this text box. A clone can be unburied by clearing this box (just backspace over the name). NB FPC does not check the bands of the two clones for similarity. Clones can also be buried and unburied via the Contig Analysis window.

Remark and FP Remark: Click on a remark box to add or edit it. Hitting <CR> creates a new box to continue a remark. The remark can be deleted by backspacing over it.

Attach Marker: Attach marker brings up a window that allows the user to type in marker name (by default, the name of the last marker added is in the box); hit <CR> in the text box after typing in the marker name. If the marker is not new, it will immediately be added to the clone’s list of markers. If the marker is new, a window for the marker will be created; after setting the type and Anchor flags, enter Accept. FPC will not let you accept the clone changes while the new marker window is displayed.

Detach Marker: Opens up the marker list with the markers as hotspots; pick the markers you want deleted; they are removed from the list immediately.

Sequence status: The status can be {NONE, TILE, SENT, READY, SHOTGUN, FINISHED}.

Type: The type of clone can be {Cosmid, YAC, BAC, PAC, Fosmid, Clone}.

Set-to-Default If there are multiple gels for a clone, a stop/go button Set-to-default will be part of the edit window; select it and the gels will be listed as hotspots; select the gel to be the default; the list will be rearranged with the new gel at the top; select Set-to-default to terminate the function. Changing the default Gel is also available via the pull-down menu on the Gel hotspot of the clone text window.

Rename allows you to change the name. The existing name will become the Fp name for the one or more gels currently belonging to the clone which do not already have an Fp name. This is so the clone can still be found in the gel file without changing the gel file; i.e. when FPC looks for a clone in a gel file, it uses the Fp name r if one exists, else it uses the clone name.

Cancel "virtually" removes it; i.e. the FPC file will still contain the record, but a "!" will be in front of it. From the Search Commands on the main menu, you can access a cancelled clone. You can bring up a partial clone editor window for a cancelled clone which will allow you to add remarks or undelete it; you cannot change the contig number or coordinates since this is not valid for a cancelled record.

❖ **Edit markers.**

The following describe the edits that can be made to a marker; after making one or more edits, pick Accept to instantiate them; if a map is displayed, the changes will become visible when accepted. Select Reject to reject the changes.

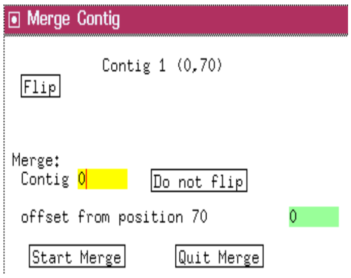
Anchor: Setting the anchor to Yes will cause it to be displayed on the bottom of the contig.

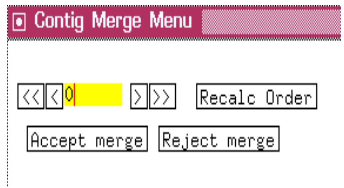
Type: This can be set to {STS, Probe, ePCR, End, Cosmid, Fosmid, cDNA, Locus, Clone, YAC, BAC, PAC}.

Attach Clone: When this stop/go button is on, the clone list is opened up; pick clones from the contig display and/or the clone keyset; the picked clones are added to the list.

Detach Clone: Opens up the clone list and makes the clones hotspots. Pick the ones to detach. Detaching all clones from a marker deletes the marker from the database.

I. Merge Operations





On the horizontal display of a contig, select the Contig button. The menu shown on the left will be displayed. Flip flips the current contig. This flip is permanent; i.e. the coordinates of all clones in the contig are changed.

❖ **Merge contigs options:**

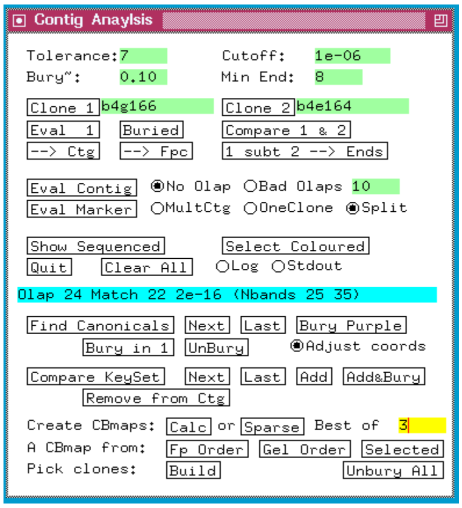
Enter the contig number to be merged; this will be referred to as the visiting contig as it will not be actually merged until you give the command. By selecting the Do not flip toggle, it will toggle to Flipped, which will cause the visiting contig to be displayed flipped. An offset from the right end of the existing contig must be specified; if this is 0, the left end of the visiting contig will be placed abutting the right end of the existing contig. (NB the coordinates of the visiting clones are not changed, they are just offset by this amount when displayed).

Select Start Merge; the visiting contig will be placed on the map with the names of its entities in a smaller font and the Contig Merge Menu (shown above) will be displayed. You can move the offset of the visiting contig to the left or right with the “<<“ and “>>” buttons. The offset of all clones will change, but their order will not change unless you select Recalc Order which will run the standard ordering routine on both contigs together. When the position of the second contig is satisfactory, select Accept Merge and the clones from the visiting contig will have their actual coordinates changed appropriately and they will become part of the current contig.

NB (1) While in contig merge mode, clones cannot be edited either by the clone editor or by stretching or snapping. (2) They can be selected such that clones from the intersection of the two contigs can be analysed and fingerprinted together.

J. Analysis

All parameters on the Main Analysis and Contig Analysis windows are saved in the FPC file. The parameters on the Rule Menu are not.



J.1. Contig Analysis

The fpc_faq.shtml document mentioned on page 3 describes rules for setting tolerance and cut-off and on how to use the automatic contig building routines.

a. Tolerance: This can have three different meanings depending on the configure values from the Main Configure menu: (1) Fixed: if the difference two bands from two different clones is less than or equal to the tolerance, they will be called 'matching bands'. (2) If the Variable flag is set and there is no Tolerance file; if two bands from two difference clones are within (tolerance * 0.01) * (the average of the two bands), they will be called matching bands. (3) If a Tolerance file is given, linear interpolation between the values is used. (See §D.2 and §A.2).

b. Cutoff: the Sulston cutoff[3] is in scientific numbers; e.g. 1e-05. It is the probability of coincidence; i.e. low numbers (high negative exponents) are good. This cutoff is also used if Equation 2 (the alternative probability) is being used (see §D.2).

c. Bury~: A clone can be buried in another clone with the state of 'approximate' if the following is true: Given M matches and N bands for the potential child,

$$\text{N-M} \leq \text{N} * \text{Bury\textasciitilde} \qquad \text{Eq 2}$$

e.g. if Bury~ is 0.1, then 90% of the bands must match the parent clone.

d. Min End: this is used when comparing ends of contigs; this is the minimal number of bands that must be shared in order to consider two contigs for a join.

Clone1 and Clone2: Either highlight a clone in the map, then pick the Clone1 button or type the clone name in. Same for Clone2.

---

3. This is the same score used in ContgiC (Sulston et al. 88). It takes into account the size of the two clones. For example:
If clones with 52 bands and 38 bands have 12 matching bands, the probability of coincidence is 3e-02.
If clones with 52 bands and 16 bands have 12 matching bands, the probability of coincidence is 3e-06.

When the Log circle is black (i.e. switched on), all results are concatenated to file proj.<userid>.log; this can be turned on/off to only record desired results. When the Stdout circle is black, all results are written to the screen. Results can be written to both the log and stdout.

NB Most functions result in highlighting clones in various colours to show various information; these colours are temporary, i.e. they are not saved with the clone, so if a coloured clone is picked and then another clone is picked, the previously coloured clone will go from cyan to clear. The highlighting obeys the rules for trail; i.e. they are not automatically cleared when trail is on. ClearAll clears all highlighted and selected clones (same as on the contig display)

❖ Functions

In the following function descriptions, the letters in parenthesis beside a function are the parameters (as described above) used by the function. The word “Log” implies that output is written to Log and/or Stdout or None, depending on associated switches. The word “HideBuried” implies that the buried clones are ignored if they are not displayed. The word “NoMerge” indicates that the function will not work when there is a visiting contig (i.e. displaying a second contig before actually merging it). The word ‘Edit’ implies that the function changes the contig. See Rule Menu in J.3 for some more functions for contigs.

Eval 1 (a, b, Clone1, Log, HideBuried): Compares Clone1 with all clones it overlap or has a good overlap score. Clone1 is highlighted grey. All “bad” matches, as defined in the two ‘Olap’ tests (see Eval Contig), are highlighted light-red. The following results are written:

```
>> Eval b1d83 16b       ctg1 (Tol 7, Cutoff 1e-08, Bad Olap  1.00, NoPseudo)
     Clone  Coordinates Bands Olap Match Diff Prob
      b1e84    0,  47    48b    15  12   3    1e-06* Spans
      b1j89    7,  48    42b    15  14   1    8e-10  Spans        canon
      b1i88   12,  49    38b    15  13   2    6e-09  Spans
      b1b81   16,  67    52b    15  12   3    3e-06* Spans        canon
      b1d83   21,  36    16b -----------------------------
      b1c82   22,  65    44b    14  13   1    3e-08               canon
      b1j79   26,  77    52b    10   9   1    1e-03               canon
```

The clones are ordered by left end. The ----- indicates Clone1. For the rest of the clones: “Olap” is the amount of overlap with Clone1 and Match is the number of matched bands. The star beside the probability is when it fails one of the Olap tests. The words {Spans, Inside, Equal} describe the relation of the coordinates of the clone to Clone1; these apply to the coordinates of the clones, not necessarily to reality!!

Buried (a, c, Clone1, Log): Finds bad buried and possible buried, as described in Eq 2. Writes to the log and colours clones, as follows: LightRed - If a clone is buried in Clone1 or Clone1 is buried in a clone, and the bury is bad. Purple - If the clone could be buried in Clone1 (this does not check if clone is already buried). LightPurple- If Clone1 could be buried in the clone (this does not check if clone is already buried). (This function is probably no longer useful).

--> Ctg (a, b, Clone1, Log, HideBuried): Compares Clone1 with all clones in the contig. Clone1 will be coloured grey. All clones that have a probability (when compared with clone1) less than the cutoff will be coloured purple.

--> Fpc (a, b, Clone1, Log): Compares the clone with all clones in the database that are not in the current contig and writes the results to the Log or Stdout, as specified.

Compare 1 & 2 (a, Clone1, Clone2, Log): The output is, as follows:

```
>> Compare cV465E5 cV468D6  (Tol 7, Shift 0)
   cV465E5   -7,   6 13b                        total  24817
   cV468D6   -6,   6 12b                        total  21373
Clone1 Clone2     Diff
   422    424    -2
          603        119
   722               -20
   742    742     0
  1479   1478     1
         1525        187
  1712               -18
         1730         74
  1804              -109
  1913   1913     0
  2234   2234     0
  2450   2449     1
         2515        156
  2671   2670     1
  2690              -400
  2887              -203
  3091   3090     1
Olap 12 Match 8 3e-08 (Nbands 13 12)
```

The first few lines tell the number of bands and accumulative length. The four columns show what bands are within the tolerance.

1 subt 2 --> Ends (a, d, Clone1, Clone2, Log): The bands of Clone2 that are shared with Clone1 are subtracted from the set of bands for Clone1, and the residual set is compared against the ends of all contigs. The results are printed if (1) the number of matched bands is greater or equal to MinEnd, or (2) the probability of overlap when comparing the residue bands with all the bands of the end clone is greater than the cutoff. If Clone2 is left blank, all the bands from Clone1 will be used. NB *This can end up matching bands that are internal because it does not calculate the residue (i.e. end) bands of the end clone; the positive aspect is that, if some or all of the residue bands are bad, they are appropriately ignored.*

Eval Contig (a, b, Log, HideBuried): All the clones in the contig are evaluated by one or both of the following tests (depending on whether the adjacent circle is filled in).

(1) No Olap: If two clones overlap based on the cutoff and yet their coordinates do not overlap, both are highlighted. All pairs of clones are checked. I feel this is a good test, but it can be misleading, i.e. if you use a cutoff that is not stringent enough, clones will overlap that should not, but this test will score perfect.

(2) Bad Overlaps: The yellow box following this button is a parameter referred to as 'Diff'. If the abs(overlap - shared bands) > Diff, a clone is marked as bad.

Eval marker (Log): All markers are evaluated by one of the following tests. (1) MultCtg: if a marker is in more than one contig. (2) OneClone: if a marker is in only one clone within the current contig. If one test fails, the marker is coloured light-red, if both tests fail, it is coloured red. (3) Split: A marker hybridizes to two or more clones that do not overlap.

Show Sequenced (Log): This colours the clones that have a sequence status and records the number of matched bands between overlapping sequenced clones. The clones are coloured {TILE green, SENT blue, READY grey, SHOTGUN dark blue, FINISHED red}.

Select coloured: Adds all the coloured clones (no matter what colour) to the selected set.

Find Canonicals (a, d, Log, NoMerge): this routine finds possible clones to bury using Eq 2. The algorithm used is as follows:

```
(1) Sort the clones by number of bands. (2) Start with the longest clone
and find all clones that fit within it based on Eq. 2. (3) Continue on
through the list until the end.
```

When the Find Canonical button is grey, the list of possible canonicals can be stepped through. Selecting Next or Last will highlight the parent in lightgray, and all the children in purple; a clone that is already buried are also shown in purple.

Bury Purple (Edit): will bury all the purple [4] clones within the highlighted one.

UnBury (Edit): click this and it will turn light red; the next clone picked will be unburied.

Bury in 1 (a, c, Clone1, Edit): click this and it will turn light red; the next clone picked will be buried in Clone1. The bury type is calculated as follows:

Code 1

```
If all the bands match  bury type = Exact
else if Eq 2 succeeds   bury type = Approx
else                    bury type = Pseudo
```

Adjust coords: If the black circle is filled in and a clone is being buried, FPC will check to see if the coordinates are within the parent, and if not:

Code 2

```
If it is to the left of the parent, it is positioned:
   left = left coordinate of C_B
   right = left + number-of-bands in child
If it is to the right of the parent, it is positioned;
   right = right coordinate of C_B
   left = right - number-of-bands in child
```

This is used for Bury Purple, Bury in 1, and the bury routines on the Rule Menu (see §J.3).

Compare KeySet[5] (a, b, NoMerge): compares all the keyset clones against the current contig and creates an internal list of clones that match one or more clones in the contig. When the Compare KeySet button is grey, you can step through the internal list by repeatedly selecting Next; all clones that match the target clone will be highlighted purple and the target clone name will be printed in the blue message box. Step backwards through the list with Last. If there is no keyset, it will use all the singletons as the default (the button will say Compare Singles when singletons have been used).

Add: When a singleton is listed in the blue message box and its matched clones are purple, you can select this function to have the singleton added to the map as follows:

Code 3

```
For all the clones that share a low coincidence score with the target
clone (i.e. purple clones), find the far left and far right coordinate,
and determine the midpoint. The left end of the target clone will be the
midpoint, and the right end will be midpoint+length (of clone in bands).
```

---

4. The reference to purple refers to the clones highlighted purple on a Next or Last; if you pick one causing it to lose its purple colour, it still gets buried; i.e. FPC references the internal list and not the actual colour.
5. Cancelling a clone causes the internal list for this function to be incorrect; so after cancelling a clone, it will insist that you run this function again. The same is true for Find Canonicals.

You should view the fp display and/or CBmap in order to refine the coordinates. By executing Select Coloured, the clones will become the selected set and hence, can easily be viewed in the Fp, Gel or CB windows.

Add&Bury: When a singleton is listed in the blue message box and its matched clones are purple, you can select this function. It will add the clone as follows: Find the clone that has the lowest coincidence score with the target clone and add it according to Code 1 and Code 2.

✥ CBmap routines

The following routines build CBmaps, as described in §L. All output goes to stdout (i.e. it ignores the Log and Stdout circles).. See §K.2 for instantiating the CB coordinates in the contig.

Calc (a, b, c, HideBuried) build one or more CBmaps for the contig. Each CBmap is a transitively overlapping set, i.e. each clone in the CBmap has a good probability (i.e. below the cutoff) with at least one other clone in the CBmap. Some clones may be 'Ignored' if they do not overlap with any other clone in the contig based on the cutoff. The right-most clones may have poor alignments as they are the clones with the poorest accumulative scores are added last. Clones that are already buried will be treated as buried by the CB routine and, if there are additional clones that can be buried, these will be determined.

Sparse: same as Calc, but it removes redundant clones, which are buried as pseudo's on OK.

Best of: given the value is N, the algorithm will generate N solutions and display the best one.

Selected (a, b, c) runs the CB routine on the selected set and forces all clones to be in one set.

Build (a) allows you to pick clones to add to the CBmap. If you pick them in an non-optimal order, they may create a very poor CBmap; at any time, you may select the Again button on the CBmap window to re-compute the consensus bands.

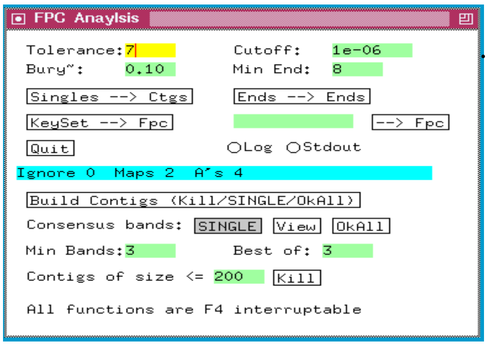
Gel Order (a) runs the CB algorithm with the constraint that the clones must be greedily added in the order according to that found in the Gel image window and any bands to be added must be added on the bottom (right) of the map. Given a good gel order, this provides an excellent map.

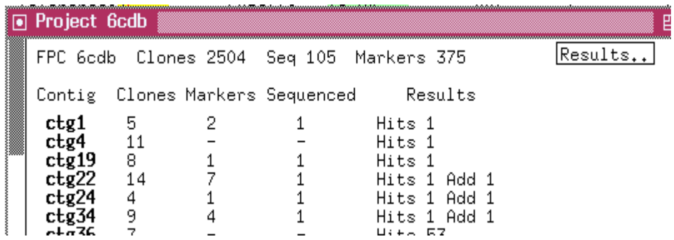
Fp Order (a) same as Gel Order but the order of clones is taken from the fingerprint window. If no fingerprint window is active, the order of the clones in the contig is used.

NB For all the function except Calc and Sparse, if you input clones that are not connected, i.e. there are two or more transitively overlapping sets of clones, it will tell you that you have N disconnected set. If you have disconnected sets, you are liable to get nonsense results and some possible 'CB error' messages. It does not work to input a set of clones with the following layout:

As a consensus band will only be made if there are two bands at least matching it. So the center bands will end up as extras or incorrect matches to otheres. In other words, it expects are least a 2-fold coverage.

J.2. Main Analysis



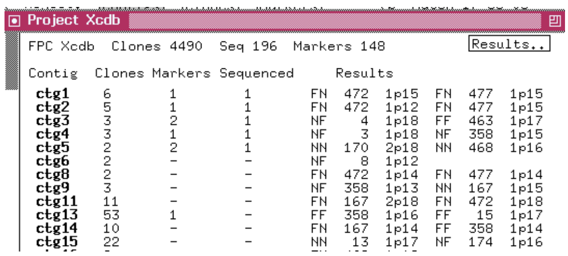


The main menu analysis window has text boxes for (a) Tolerance, (b) Cutoff, (c) Bury~ and (d) MinEnds as described above. The following three functions write to the Project window.

Single -> Ctgs (a, b, Log): Compares all new clones against existing contigs, and writes on the project window the number of hits per contig. If log or stdout are on, all matches are written to stdout or the log file ‘sorted by contig’.

KeySet -> Fpc (a, b, Log): Compares all clones in the keyset against all clones in the FPC database (except cancelled clones), and writes to the project window the number of hits per contig. The output above used the auto-add option (see §J.3). If log or stdout are on, it writes all matches ‘sorted by coincidence score’.

Ends -> Ends (a, b, d, Log): Compares all pairs of contig ends for (1) MinEnd or more matches and (2) a probability below the cutoff; if either test succeeds, the corresponding contigs are candidates for joins. It lists at most two possible joins for each contig; if there are more than two, it lists the ones with the most matched pairs. *This considers all the bands for each end clone which may result in matching internal ones.*



As shown in the Project window above, the triple is as follows: a two letter flip code, the contig to be joined, and the number of pairs matched followed by a *p* then the maximum number of bands matched for the pairs of clones (e.g. ctg1 should be flipped, then joined with ctg472, they share one high scoring pair which matched 15 bands). The following relates the flip code to operations in the contig merge menu.

Code	*Contig*	*Visiting Contig*	*Orientation*
FN	Flip	-	LL
FF	Flip	Flip	LR
NN	-	-	RL
NF	-	Flip	RR

--> Fpc (a, b, Log) This command is proceeded by a text box. Enter a clone name in the text box and execute -->Fpc. The clone will be compared against all clones in the database.

❖ **CBmap routines**

Build Contig runs Kill, SINGLE and OkAll, respectively. The results are written to standard out and to the Project window. See fpc_faq.shtml for more information on this command. See §M for batch processing.

The following three steps are seldom used as running them together with Build Contig is generally preferred. They just remain for completeness.

SINGLE (a, b, c) compares all singletons and computes one or more CBmaps. Any clones with less than or equal to Min Bands will be ignored. As will be shown, there is an OK button on the CBmap in which you can create a contig from an existing map.

View: displays the CBmaps creates. This can be used as an alternative to OKAll. That is, you can step through the maps and individually OK each one.

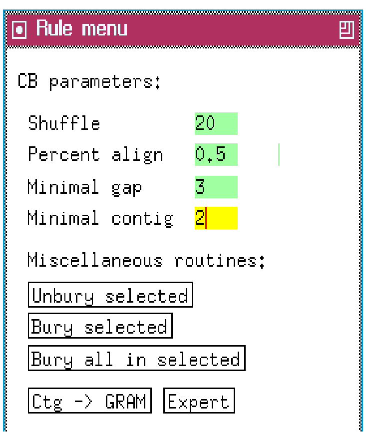
OK ALL creates a contig for each CBmaps created by SINGLE. It uses the same position as calculated but extends or reduces the length such that it is equal to the number of bands present in the fingerprint. Clones than can be buried as exact or approximate will be.

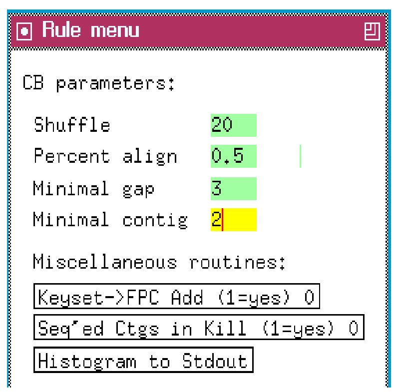
Min Bands: given the value is N, the algorithm will ignore clones with less or equal N bands.

Best of: given the value is N, the algorithm will generate N solutions and display the best one

Kill is preceded by a parameter *x*. Executing Kill will remove all contigs of size *x* or less. Contigs with sequenced clones will not be killed. If a '-1' or 'n' or 'max' is entered, the maximum size contig will be used, i.e. all contigs without sequenced clones will be killed.

J.3. Rule Menu





The pull down menu on both the Main and Contig Analysis windows have an option called Rule Menu, which pop-up a window as shown above.

The first four items are parameters used by the CB routine, as follows: (1) The first determines how many consensus bands to create before doing a shuffle. (2) The Gap parameter is used in the CB heuristics. (3) The percentage of bands that must align to the consensus or extend the CBmap; the clone is marked with an 'A' if this percentage is not matched and the clone is centered in respect to the clones it shares a high overlap. (4) This determines the minimal contig that will be created. I have found these values to work fine and never change them.

For the Contig Analysis, the following four items are on the Rule Menu:

Unbury selected: this function unburys any clone that is selected and buried.

Bury selected: this function buries all selected clones as follows: Find the unselected clone with the lowest coincidence score and bury the clone according to Code 1 and Code 2.

Bury all in selected: this function buries all un-selected clones as follows: Find the selected clone with the lowest coincidence score and bury the clone according to Code 1 and Code 2.

Ctg -> GRAM: writes a GRAM fragment file for the contig.

Expert (a, b, d, HideBuried) tries to find the set of 'good' overlapping clones.

For the Main Analysis, the following three items are on the Rule Menu:.

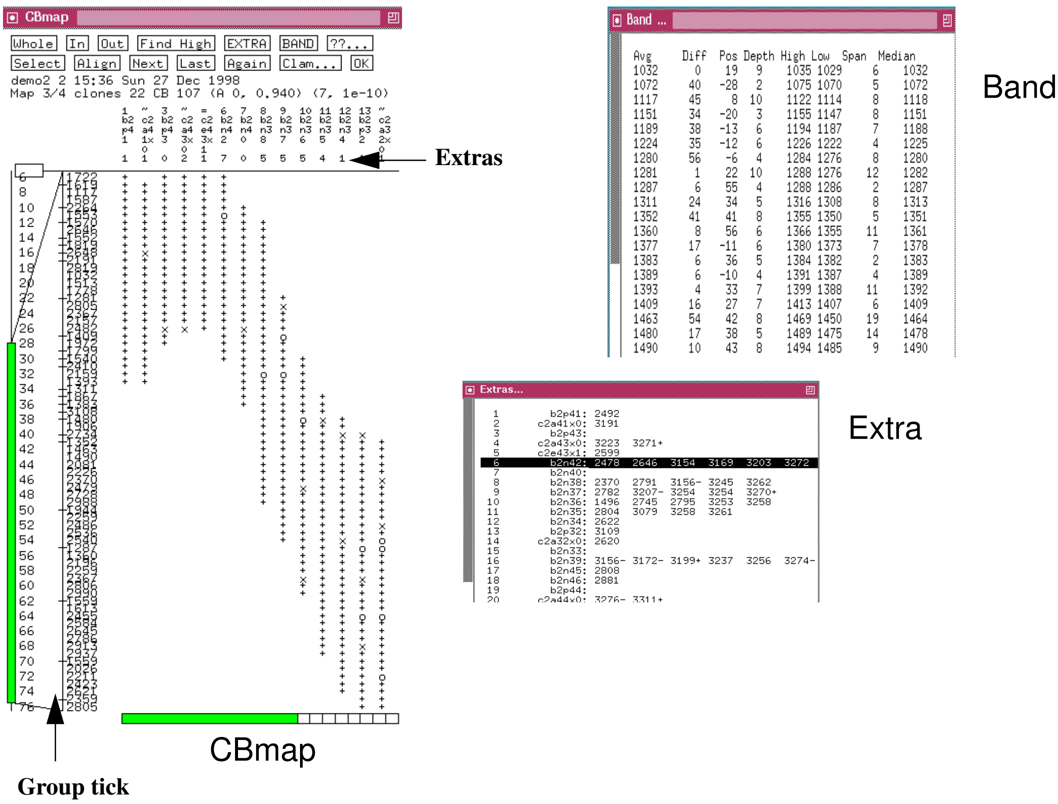
Keyset->FPC Add: selecting this toggles the 0 to 1 which turns on the option. When this is 'on' and Keyset->FPC is executed, clones are automatically added to contigs as follows: Find the clone with the lowest coincidence score that is in a contig. Add it to that contig as described in Code 3. *Beware if you build a contig using only this automatic add feature, the fidelity of the positioning of the clones will digress as its only a rough approximation. You should run the CB routine for accurate positions once a contig has good coverage.*

Seq'ed Ctgs in Kill: selecting this toggles the 0 to 1. When 0, the Kill function will not destroy a contig that has clones marked for sequencing. When 1, it destroys any contig below the limit.

Histogram: writes a histogram of the band sizes to stdout; The second column is the exact number of the bands of the size given in the first column; the third column is the number of +/- tolerance within the band.

K. Consensus band (CB) map

The Consensus Band (CB) function creates one or more CBmaps; each map has a consensus band map and an alignment; each map includes all clones linked below the cutoff.



From the main analysis window, the CB routines can be run on all singletons. From the contig analysis window, the CB routines run on the current contig or a selected set.

Referring to the CBmap in the figure above; the consensus bands are the second column on the left; the ticks beside some of the bands show the start of a new group. A group is a set of bands that can be in any order within the group. The clone names are listed across the top.

If an ‘=’ or “~” is shown above a clone name, the CB algorithm has determined that the clone could be buried in the first non-’=/~’ to the left, if an ‘A’ is shown, no decent position could be found so the clone was just aligned.

The number below each clone name is the number of bands that are not included; i.e. for which there is no “+” or “x” in the column below the clone name. A “o” is a gap (i.e. a false negative if the CBmap is correct). An “x” means the band is not within the tolerance but is within twice the tolerance.

BAND lists the bands in sorted order: `Avg` is the average of all the bands (where there is an ‘x’ or ‘+’). `Diff` is the difference with the next lowest CB band. `Depth` is the number of clones with this band. The `High` and `Low` give the highest band and the lowest band contributing to the `Avg`; i.e. for all the ‘x’s and ‘+’s beside the band in the CBmap, these are the highest and lowest values. `Span` is the difference between the highest and the lowest values.

EXTRA shows, for each clone, the bands that were not used. When ?? is on, selecting a clone on the CBmap causes the following: all extra bands for all clones in the set are compared with the current clone, and a + or - is placed beside the band if it is within the tolerance with one of the highlighted clone's extra bands.

?? tries to align any extra bands within the map. It checks to the left and right of the clone, and if an extra band fits within the tolerance of consensus band, it puts a "?", and if there is an extra band that are within twice the tolerance, it puts an "X". The ?? can be selected with the left button to do as just stated. Or, the right button will pull down a menu with the following:

- Show Friends: Will highlight the clones with a good overlap score with the current clone, i.e. the more recently picked clone, and any clone picked while this function is on.
- Clear Friends: turns off the above function and unhighlights all clones.
- Show Sparse: highlights redundant clones. When a Calc solution is displayed, these are the clones that will be removed with Sparse.

Next shows the next set. Last shows the last set.

Again re-computes the CBmap for the current set. It starts with a different clone and difference random number and hence, gives a different map. On the first run of the CB algorithm, the clones are displayed in the order in which they are added; on subsequent Again's the order is not changes as this makes it easier to determine what clones consistently do not place well and should be removed. Even though multiple solution are tried and the best displayed, it can still be a sub-optimal solution. If there are many A's, try Again a few times to see if a better solution exists.

Clam/Left/Real is a toggle that changes the order of the clones, as follows: Clam[6] is the order in which they were added to the map. Left is the order of their left CB coordinate. Real only applies to the map if computed for a contig; the clones are ordered by the real left coordinate.

OK is described in the next section.

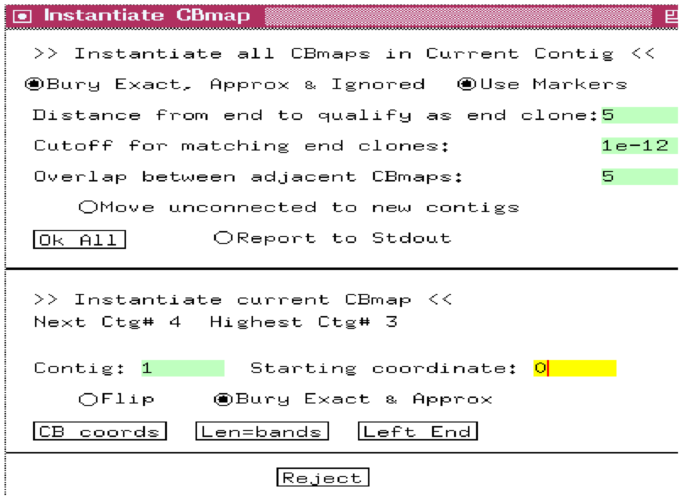
When run from the Contig Analysis, the following two functions are on the CBmap window.

Align is a stop/go button, when on, picking clone from the current contig will align the clone with the CBmap; this will not produce any new bands, it will only align to the current set.

Select will make the current CBmapthe selected clones.

6. Clam is the name we use for the analysis software and comes from "Cari's lean-mean assembly machine".

K.1. Ordering CBmaps



OK brings up a menu in which the user can instantiate all CBmaps at once, that is, assign position to the clones based on the CBmap, or each CBmap individually.

OkAll instantiates all CBmaps by comparing the end clones and arranging the maps based on best scores, where the score includes both cutoff and marker data.

- If Bury .. is set, exact and approximate are buried, and Ignored clones that are not used for bridging CBmaps are buried as pseudo.
- If Use Markers is set, they are used in the score to determine if two CBmaps are adjacent.
- The end clones are those that have a '+' on either of the end of the CBmap or within X of the end where X is the first parameter in the menu.
- Two end clones match if their score is below the cutoff N where N is the second parameter in the menu. By defaults, 2 is subtracted from the exponent.
- Adjacent CBmaps will overlap by D units where D is the third parameter in the menu.
- If Move.. is not set: disconnected contigs are placed with 10 spaces and a yellow bar between them. If it is set, each disconnected contig is moved to a new contig. The first set of merged CBmaps stay in the current contig (arbitrary).

These parameters can be changed and OkAll executed again without having to regenerate the map. The clone coordinates are computed as Len=Bands, as described below. *NB Small CBmaps are sometimes flipped the wrong way due to insufficients information.*

To OK a individual CBmap, enters a contig number and an offset. Depending on the contig number, (i) the set is made into a new contig, (ii) added to an existing contig, or (iii) the coordinates of the clones in the current contig are changed. It the Flip circle is selected, the CBmap will be flipped. Then execute one of the following.

CB Coords This uses coordinates that exactly align to the CBmap.

Len=Bands It uses the same position as calculated but extends or reduces the length such that it is equal to the number of bands present in the fingerprint.

Left End Is the same as Len=Bands but will insure the clones left ends are ordered the same as the clones, e.g. the first clone must have a left end to the left of the second clone, etc. This is generally used with the Gel Order and Fp Order functions.

K.2. The CB algorithm

The algorithm is described in detail in Soderlund *et al.* [1997]. This algorithm uses the Tolerance, Cutoff, and Bury~ values. There is another variable: Gap, which by default is set to 3; this is used in some of the heuristics and is located on the Rule Menu. And finally, the Shuffle parameter tells how many consensus bands to accumulate before doing a stochastic shuffle (as used in SAM and GRAM) which gives a initial arrangement of the bands; this is done because there are not enough constraints when first starting the map to correctly order groups of bands, but after the shuffle, the remaining consensus bands are added greedily.

The Calc and Single algorithm is as follows:

(0) For all clones to be analysed, compare each clone with all other clones to be analysed, as follows: Compute the number of matched bands and the Sulston score. Calculate the buried clones. Make a list of all the clones that have scores below the cutoff value. A grand score is computed for each clone based on the number of good scores and length of clone.

(1) Start with the clone with the best grand score, call it clone1, and start a CBmap, as follows: Put all the clones in the clone1 list on a queue where the score for each queued clone is equal to the number of matches; buried clones are given highest precedence.

(2) Take the best clone off the queue and (i) initialize the CBmap with the bands that are shared between the two, and (ii) put all the clones from the clone2 list on the queue, and if the clone is already on the queue, make its score the greater of the two matched band values.

(3) Until all clones are gone from the queue, take the best clone off the queue, and (i) align it with the current CBmap, (ii) extend the map to the left or right if possible, and (iii) add the clones in its list to the queue (as in ii above). If a good position cannot be found (i.e. a decent number of bands matched with a low number of gaps), the clone is just aligned to the consensus map and step ii is skipped; this was done so that all linked clones would, come hell or high water, be shown in the map (this can still result in totally screwing up any additional clones; the user should cancel such bad clones and re-assemble).

(4) If there are any clones left with an acceptable grand score, go to 1 and try to make the next set.

L. File formats

❖ **bands file (input)**

```
                c217       4          9
390
926
1144
3310
-1
                c218       3          9
885
912
1520
-1
```

A file has one or more clone entries. The first line of a clone entry is the clone name, number of bands, and the gel number. This is followed by the list of bands for the clone, one per line. The list is terminated with a -1 (The number of bands and list terminator are redundant information but both are necessary). The above example has two clones from gel 9.

❖ **Merge ace file (option 2)**

Markers can be added from an aceDB formatted file. A sample input is as follows:

```
BAC : "bK3D3"
Positive_STS      "stdJ412A9A"
Positive_STS      "stdJ515N1B"
Positive_probe    "pryA78E6R"
Gridded  "BAC3"
Gridded  "SC22BAC_grid_1"

BAC : "bK30G2"
Positive_STS      "stbK592D9.TV"
Positive_STS      "stbK592E10.TV"
Positive_STS_weak         "stcN128A12A"
Positive_probe    "prpH19"
```

The type of clone can be {BAC, PAC, YAC, Fosmid, Cosmid, Clone}. All Positive_x where x can be {STS, Probe, End, Cosmid, Fosmid, cDNA, Locus} are added. If there is a _weak, this information is also added and such clones are highlighted in a different colour. The word in quotes after 'Gridded' gets added as a comment. All other lines keyword lines are ignored. The fpc file is updated to be exactly like the input file (NB the one exception is if a clone is not included in the input file, its marker data is not deleted).

Framework file - our framework markers are from radiation hybrid maps and from YAC/STS maps. These are used to orient contigs (see §E). They are also automatically made anchors when read in (see §F.3). The file must be called 'framework' and exist in the same directory as the marker file. Its format is a table dump from aceDB. Example

```
"bK7D10";"Chr_22";14534.6;
"bK10D8";"Chr_22";28030.6;
"bK24D6";"Chr_22";32418.1
```

The 2nd term is internally made the project name and is used on subsequent dumps of data. The last term is the location.

❖ Merge ace format (option 1)

The first line must contain '// SAM' or '// Framework'. Then there can be zero or more sets of clones and their markers, for example:

```
// SAM
Clone : "dJ145C9"
Positive_STS "stSG22782"
Positive_End "stSG61655"

Clone : "dJ163A11"
Positive_End "stSG61655"
```

Clones must be in FPC, markers are added. Any of the marker types allowed by FPC can be used, if the marker already exists in FPC, the type is ignored.

The set of clones is followed by a line with 'Markerdata' and the framework marker, e.g.

```
Markerdata

Marker_End "stSG22782"
Global_position 5.0

Marker_End "stSG61655"
Global_position 275.0

Marker_End "dJ1107A15SP6"
Global_position 285.0
```

Markers must be in the FPC to be added to the framework (if works fine if it was just added via one of the clones in the file). A marker can be removed from the framework by using 'No_Global_position'. The file can contain only clones and markers or only Markerdata.

M. Batch processing

>fpc demo2 -batch -help

```
Usage: fpc project -batch [args]
   Parameters for clone name analysis
       -n clonename
       -o output file [default stdout]
   Parameters for Build Contigs
       -k maximun sized contig to kill [use -1 for maximum size]
       -m minimum sized contig to build [default 2]
   Parameters for either option
       -c cutoff [default from project fpc file]
       -t tolerance [default from project fpc file]

   Parameter to dump sequenced clones
       -s digit where digit is
          1 pace
          2 oracle
          3 ace all contigs
          4 ace seq contigs
          5 clones with bands
       -o output file [default stdout]

   Parameter to load ace file which must be in fpc directory
       -u ace file [e.g. bacterial.ace]
       -d complete directroy path
           [default fpc direcotry, sub-dir ‘update’ ]
       NB if files ‘framework’ and/or ‘sequencenew.ace’ exist,
           they are also loaded
   Errors go to stderr and exit(0)
   Other Fpc output and batch args go to stdout

One of the following arguments must be present: n, k, s or u.
```

Acknowledgments

We would like to thank Alan Coulson, Simon Gregory, Gareth Howell, Lisa O'Neil, Joanne Fay, Sean Humphray and the rest of the fingerprinting team for their suggestions and bug reports. We would also like to thank Ian Dunham for his suggestions on the FPC <-> ACeDB interface.

References

Collins, J. *et al.* (1995). A high density YAC contig map of human chromosome 22. *Nature* 377:367.

Coulson, A. and J. Sulston (1988). Genome mapping by restriction fingerprinting. In *Genome Analysis: A Practical Approach*, ed. K.E.Davies, p 19, IRL, Press Oxford.

Coulson, A., J. Sulston, S. Brenner, and J. Karn (1986). Toward a physical map of the genome of the nematode *C. elegans. Proc. Natl. Sci. USA.* 83: 7821-7825.

Dunham, I., R. Durbin, J. Thierry-Mieg, and D. Bentley (1994). Physical mapping projects and ACEDB In *Guide to Human Genome Computing,* ed. Martin J. Bishop, Academic Press Limited.

Gregory, S., Soderlund, C, and A. Coulson (1996). Contig assembly by fingerprinting. In Dear, P.H. (ed), *Genome Mapping: A Practical Approach*, O.U.P. (book in preparation).

Soderlund, C. and L. McGavran (1993). GRAM User's Manual. Los Alamos National Laboratory, Technical Report LA-UR-93-3855.

Soderlund, C. and C. Burks (1994). GRAM and genfragII: solving and testing the single-digest partially-ordered restriction map problem. *CABIOS* 10: 349-358.

Soderlund, C. (1995). SAM V2.5 User's Manual. Sanger Centre, Technical Report, SC-01-95.

Soderlund, C. and I. Dunham (1995). SAM: A system for iteratively building marker maps. *CABIOS,* 11: 645-655.

Soderlund, C., I. Longden, and R. Mott (1997a). FPC: a system for building contigs from restriction finerprinted clones. *CABIOS*. Vol 13 No 5.

Soderlund, C., S. Gregory and I. Dunham (1997b). Sequence ready clones. In Bishop, M.J. (ed) *Guide to Human Genom Computing*, Academic Press (paper & book in preparation).

Sulston, J., F. Mallet, R. Staden, R. Durbin, T. Horsnell and A. Coulson (1988). Software for genome mapping by fingerprinting techniques. *CABIOS* 4: 125-132.

Sulston, J., Mallett, F., Durbin, R. and T. Horsnell (1989) Image-analysis of restriction enzyme fingerprint autoradiograms. *CABIOS*, 5:101-105.