

ComboScreen User Guide

D. Curtis Jamison

January 11, 2000

NHGRI/NIH

Bethesda, MD 20892 USA

PUBLIC DOMAIN NOTICE

This software/database is "United States Government Work" under the terms of the United States Copyright Act. It was written as part of the author's official duties for the United States Government and thus cannot be copyrighted. This software/database is freely available to the public for use without a copyright notice. Restrictions cannot be placed on its present or future use.

Although all reasonable efforts have been taken to ensure the accuracy and reliability of the software and data, the National Human Genome Research Institute (NHGRI) and the U.S. Government does not and cannot warrant the performance or results that may be obtained by using this software or data. NHGRI and the U.S. Government disclaims all warranties as to performance, merchantability or fitness for any particular purpose.

In any work or product derived from this material, proper attribution of the authors as the source of the software or data should be made.

1 Introduction

Creation of large genome sequence ready maps requires screening a large number of markers. While STS and overgo technologies make the assays easier, doing an individual hybridization for each probe can quickly become onerous. Multiplex probe analysis was introduced for cosmid mapping [Evans and Lewis, 1989] and adapted to medium size BAC contig mapping [Cai et al., 1998]. Multiplex (or combinatorial) screening greatly reduces the number of experiments needed to create a sequence ready map, however resolving the combinatorial pools quickly becomes a nightmare of accounting paperwork. ComboScreen was created to alleviate and automate some of this processing.

The ComboScreen program allows the user to enter and track the substrates needed for combinatorial screening. Oligonucleotide probes can be entered either from tab-delineated files or directly parsed from the output of OvergoMaker 4.0 [MacPherson, 1998]. Clone plates are generated automatically from library screen results, and probe pools and filters are constructed using tools which graphically reflect the object being input. The probes, plates, and filters are stored in a database back-end. Screen results are input graphically, and resolved probes are output in a format suitable for input into map building software like SegMap.

2 Getting started

2.1 Installation

2.1.1 Requirements.

ComboScreen is supplied as C++ code. You need a good compiler, as well as the support libraries. The primary library you need to acquire is the wxWindows GUI toolkit [Evans and Lewis, 1989]. This library is available for a variety of operating systems and compilers, including UNIX, Linux, and MS Windows. Additionally, you will need a database system and associated coding libraries. Currently, the database and libraries expected are the Sybase 10.* series.

2.1.2 Compilation for UNIX

(For the absolute latest instructions, see the file README)

1. Install wxWindows and Sybase, if needed.
2. Unpack the source tree. In the CS2 directory, you should have three directories (graphics, src, and include) and one file (makefile).
3. Follow the instructions to set up the database (see below).
4. Set up the CB_LIBRARY table.
5. Edit src/makefile.unx to point at the proper places.

6. Edit `src/LoginBox.C` to set the names for Sybase database.
7. Do a "make all" at the top of the source tree.

Other compilers will vary. In general, ComboScreen follows the compilation procedures of wxWindows, so solutions to specific problems might be generalized from the problems solved during installation of those libraries.

2.2 Database Setup

ComboScreen relies upon a backend database to store information. You can define multiple databases on your server and access them using ComboScreen. You must edit `LoginBox.C` to tell the program how to connect to your server and the databases. There are three variables that need to be changed.

`LoginBox.C` line 17: `#define DBNUM 4`

`LoginBox.C` line 18: `#define DBSERVER "quark"`

`LoginBox.C` line 32: `wxString c1[DBNUM] = {wxString("dev_jamison"), wxString("mh7synteny"), wxString("rh7synteny"), wxString("hRPC11")};`

First change `DBNUM` to reflect the actual number of databases you're setting up. Then alter `DBSERVER` to be the name of the database server you have defined in the system (for Sybase, it is the alias specified in the interfaces file). Finally, edit the `wxString` array `c1` to contain the name(s) of your database(s). It is very important that the number of `wxStrings` defined in the array matches `DBNUM`, or your compiler will barf when you do the make.

The table structures are given in the text file `DB_Setup.txt`. Another possible method is to use the (thus far untested) SQL script in `DB_Setup.sql`. Make sure to edit the script to replace `<database_name>` with the database name you are using, to bring it in line with the database names in `Login.C`.

However you create the database, it is important to **PRESERVE THE TABLE NAMES EXACTLY!** There are SQL statements embedded in the program which will fail if the tables are not right. Bad things may happen to your data if you change the table names or structures without fixing the sourcecode.

2.2.1 Using other Databases

ComboScreen is capable of utilizing any SQL database. Simply subclass the `dbConnector` class (`connector.h`) and provide the proper methods. Classes inheriting from `dbConnector` act as adaptors to the database.

In theory, you could even write an adaptor that took the SQL queries and translates them into a non-SQL query language (like ACEDB). You could also play games with what the tables look like through your adaptor.

2.3 Library Setup

You must initialize the `CB_LIBRARY` table with your library. The best way to do that is to use the bulk-copy routine provided with your database. The three fields you need are `filter_panel`, `pattern`, and `location`. `Filter_panel` is the filter number and the panel number concatenated with a period. `Pattern` is the pattern number, derived from the stamping patterns found in the `pattern#.xpm` files. `location` is the library location which the pattern resolves to.

For example:

1.1,1,1

1.1,2,7

1.1,3,13

1.1,4,19

1.1,5,25

1.1,6,31

1.1,7,37

1.1,8,43

defines the 8 wells stamped on filter 1, panel 1 in the `RPCI23` library. The program will automatically pad the plate with leading zeros if there are less than three digits. If your library has more than 999 plate locations, or if you want to have a custom prefix, you need to fully qualify the location name you are importing (*eg.*, “1.1,1,NH0001” is how we defined the first well location in the 1152 well `RPCI11` library). You will also want to make sure the definition of the `CB_LIBRARY` table reflects the change (*eg.*, `char(6)` for the `RPCI11` library designation above).

The program is written for the `RPCI23` mouse BAC library, a 60 panel library with 6 panels per filter, stamped in duplicate. This is unfortunately hard-coded. If you need to alter this, check out `LibraryObject.C` and make the appropriate changes there. You will also probably want to check out the stamping patterns. If you alter or add patterns, make sure you include the `.xpm` files needed to draw the patterns. Also, make sure the `#defines` in `def.h` reflect your library as well.

2.4 Other Platforms

One of the attractive features of `wxWindows` is the multi-platform support. Even though the current `ComboScreen` release supports only Motif UNIX systems, `wxWindows` supports `MSWindows`, Linux, GTK, and hopefully soon Macintosh. In theory, the only limiting factor should be the availability of the database libraries.

In order to migrate to another platform, create a `makefile.XXX` in the `src` directory, paralleling the structure of `makefile.unx`. Then create a dependency in the makefile, and change the "all" dependency to point at your dependency. For example, to create a Visual C version for Windows, create a `src/makefile.vc`, then edit the make file to look like this:

```
all:
  make vc; cp src/ComboScreen_vc ComboScreen
motif:
  cd src; make -f makefile.unx motif
vc:
  cd src; make -f makefile.vc
clean:
  rm ComboScreen; cd src; make -f makefile.vc clean
```

Note the three places there is a change from "motif" to "vc": Twice in the all dependency and once in the clean dependency. I expect the makefile will get a little more sophisticated in future releases so this will get easier.

3 Running ComboScreen

When you start ComboScreen, you are presented with the control panel, as shown in Figure 1.



This screen is the main navigation center for ComboScreen. The menu bar gives you the options you need to input data, view data, or perform data analyses. Of special note are the two fields at the bottom: The status line and the user name. Since everything in ComboScreen relies upon a connection to the database, you must be logged in before you can do anything. These two fields keep you informed of your log in status, and your login name.

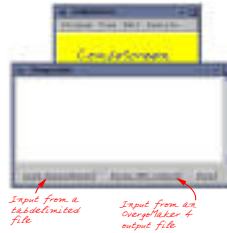
3.1 Database Connection.

Under the Program menu, there are three options: Log In, Log Out, and Quit. These are fairly self explanatory. You must be logged in as a user to access any menu options other than Program:Log In or Program:Quit. If you are already logged in, choosing either Log In or Quit will automatically log you out before performing the function.

3.2 Entering Data

3.2.1 Oligonucleotide Probes

Edit:Import Overgo will bring you to the overgo editor, as shown in Figure 2.



In this screen, you really cannot edit the overgos, but you can import new ones from either a spreadsheet or from OvergoMaker4 output.

To import from a spreadsheet, use your favorite spreadsheet to create a file with the data following columns:

1. name
2. forward
3. reverse

and export it as a tab-delimited text file. Transfer the file to a place accessible to ComboScreen, making sure that the text file is in the proper flavor (*eg.*, Unix, PC, or Mac). Then when you press “Load Spreadsheet” you’ll get a file chooser menu where you can either navigate to your file or type in the name. The rest is automatic, and you’ll be able to monitor the progress in the Editor screen.

To extract primers from an OvergoMaker4 output file, press the “Parse OM4 Output” button and choose the output file. The advantage of using an OM4 file is that you have the opportunity to add comments the record. Figure 3



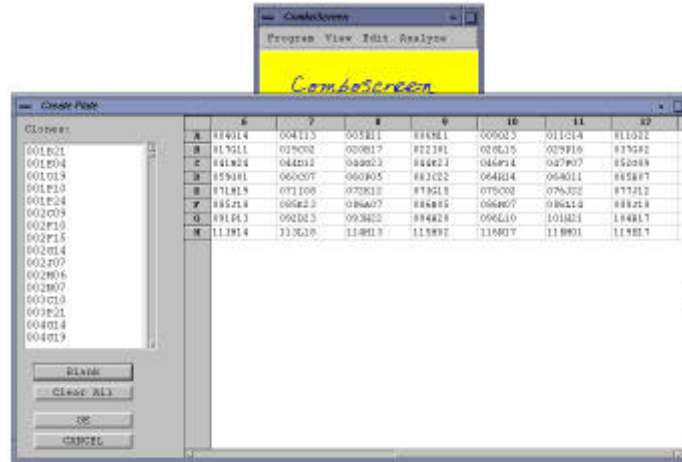
shows an overgo import in progress, with the comment field.

3.2.2 Plates

96 well plates are generated automatically for newly identified clones from a library screen analysis. If *in vitro* were as clean as *in silico*, that would be all that was necessary. However, as we all know, nothing is that simple. Occasionally

clones don't grow, or clones are picked into the wrong well. For cases like that, the Edit>Edit Plates menu choice exists.

The plate editor is shown in Figure 4.



The plate is represented on the right in a spreadsheet format. On the left is a list of all possible clones, and a set of buttons. Corrections in the plate are made by selecting a clone name, then clicking on the well to change. To delete a clone from a well, press the “Blank” button before clicking on the empty well. When you’ve finished, press “Ok.” If you really want to, you can clear the entire plate using the “Clear All” button. But then you’d have to go through and rebuild the plate by hand, which would be a fairly tedious exercise.

3.2.3 Pools

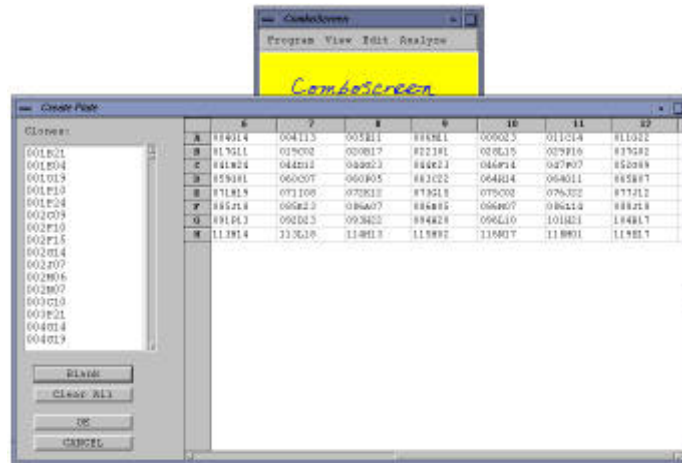


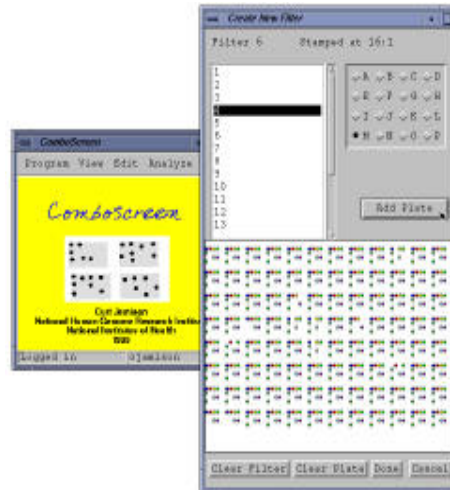
Figure 5 shows the pool creator interface, which is accessed by choosing

Edit>Create Pools from the menu. The interface works exactly like the interface for editing plates described above. Currently the 15 X 15 grid size is static.

Put an overgo into a cell by selecting it and clicking on a cell. When you've added all the probes you wish to screen in this experiment, press "Ok" and the program will create the pools for you. It will the print out the pool grid and a pools list.

3.2.4 Filters

Filters for secondary screens are created using the interface in Figure 6.



A list of plates is at the top left, while a set of radio buttons is on the top right. The radio buttons represent the 4X4 grid created using a robotic filter stamping machine. Select a plate from the list, turn on a radio button, then press the "Add Plate" button. In the large filter field, you will see a set of dots appear: this represents the plate that you just stamped onto your filter. Continue selecting plates and grid locations. Each location is a different color to make it easier to see.

If you make a mistake, use the buttons at the bottom to correct them. "Clear Plate" wipes out the currently selected grid location. Use the "Clear Filter" to wipe away all the plates if you've really screwed up. "Cancel" will quit the filter editor without saving anything, while "Done" writes the information on the filter to the database. Note there isn't any provision for re-editing the filter: What you have when you click "Done" is what you'll store. If it is wrong, you'll just have to create a new filter.

3.3 View

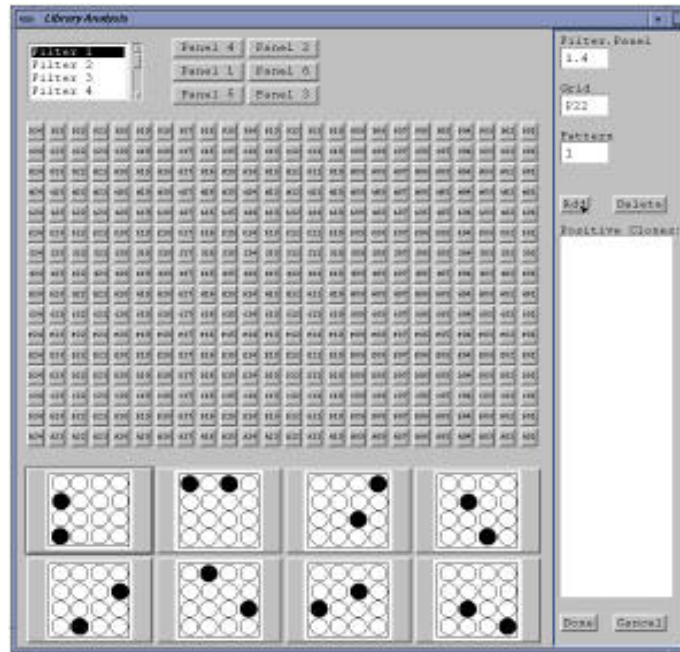
Database entries can be viewed from the View menu. All the data is presented in a table format. There currently is no way to restrict the view, edit the data,

or to print the table. Large scale data manipulation like that is currently only possible through the SQL interface of the RDMS (*eg.* Sybase's isql program). Note: The database browser has been only partially implemented, since it was deemed to be the least important viewer. Portions of it do not work.

3.4 Analysis

3.4.1 Library Screens

The primary library screen input interface is shown in Figure 7.



The three panes on the left are the primary input method. The top pane allows you to select the filter and panel you are currently working on. The middle pane represents the grid within the filter. The bottom pane shows the possible patterns. The unique address of each clone is specified by clicking the appropriate boxes.

On the right is the clone list controls. As you click in the panes on the left, the three text fields at the top right change value to reflect what you just clicked. Errant clicks can be corrected simply by clicking over again. When the three text boxes hold the proper information, press the “Add” button to resolve the coordinates into a clone name, which appears in the positive clones list in the middle right. The grid and pattern fields are reset to empty, while the filter and panel remain the same until you click on a new one.

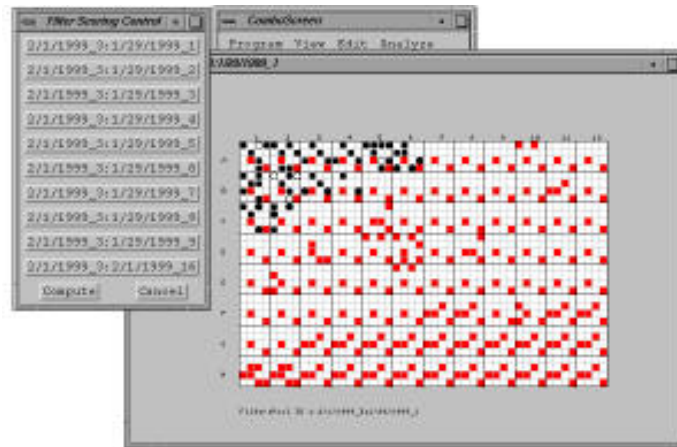
Three buttons at the bottom control the positives list. Selecting a clone name and pressing the “Delete” button will remove that clone from the list. Pressing

“Cancel” will cancel the analysis without writing anything to the database, in case you’ve been working from the wrong filters. The “Ok” button will take the clone list and compare it to the clones already stored in the database. Clones already present will be moved to the “common” list, while new ones will be placed in the “unique” list. The unique clones will be assigned to wells in new plates, and finally you will be asked to choose a file for the final report.

The final report contains the list of common clones, as well as a pick report, which gives you the plate and well assignments for each clone. When you’re picking the clones, any deviations from the pick report will need to be logged, so that you can edit the

3.4.2 Secondary Screens

To analyze a multiplex screen, you’ll choose **Analyze:Secondary Screen** from the menu. First you’ll be presented with a dialog box asking you to select a filter and pools to test. Select a filter and the associated pools. After a few moments, you’ll see several windows appear. There is a window for each pool, and a control panel with buttons that allow you to choose which filter-pool combination you’re currently entering, as shown in figure 8.



Screen results are entered by clicking in available (non-red) squares where the filter shows a positive. You’ll see a solid black dot appear. If you have the same clone in another spot, that one will appear as a hollow dot (a “shadow” positive). Note that this is to help identify clones that have been stamped in duplicate and only the first clone with the same name will be shadowed, so if you have a clone in three or more spots you’ll only see the first two.

When you have entered all the positive spots for one pool, choose another from the control panel. When you have finished all the windows, click the “Compute” button in the control panel. The filter results will be compared pairwise, and reports will be written to a couple files in the current directory. The “log” file contains a verbose report about the experiment. The “segmap”

file contains the same information in a form that can easily be translated to a SegMap input file.

3.4.3 Overgo Test Screens

This interface works identically to the secondary screens interface, except that instead of choosing pools, you choose the overgo(s) you wish to test. This is good for confirming the results of a pool resolution.

3.4.4 Primary Screens

Primary screens are halfway in between library screens and secondary screens. The 96 well plates are rearranged into 384 well plates and stamped out like a library filter. The master filters created are then probed using pooled overgos, and resolved in a manner similar to the secondary screens. This is specifically for use in end-walking from established contigs.

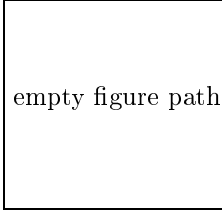
After selecting the Analyze:Primary Screen choice, you'll be presented with a dialog box that lets you choose your probe pools. After clicking the "Ok" button, you'll get a control panel that lets you switch between the filter-poll combinations. Select one, fill in the positives like a library screen, then choose another from the control panel (you'll note that you no longer have a done or abort button on the library screen). When you're finished entering your filters, click on the "Compute" button in the control panel.

4 Database Technicalities

The database setup is pretty much straight forward. The tables are defined in the file DB.Setup.txt and, if you're brave, the file DB.Setup.sql contains an untested and unsupported SQL command file which might set up your database automatically. Most of the tables can be filled in via the interface. However, there are three tables which must be populated using methods outside the ComboScreen program. The CB_LIBRARY table must be filled before the ComboScreen program is used. The CB_MASTER and CB_REARRAY tables must be filled if you're going to use the primary library screens. Use of a utility program like bcp to load these tables from a spreadsheet is highly recommended.

The structures of the CB_LIBRARY and CB_MASTER tables are analogous. The filter_panel field is a decimal representing the filter number and the panel number. For example, the middle left panel panel on the third filter of the RPCI11 library would be 1.1 (due to the layout pattern of the library). The pattern is the stamping pattern, numbered 1-8, and the plate is the corresponding plate number. The illustration of the LibraryAnalysis screen should make this a little more understandable.

The CB_REARRAY table data is made clear in figure 9. Four 96-well secondary plates are rearranged into a 384 well plate according to the scheme shown. The CB_REARRAY table supports the translation of the master filters into original library clone names.



empty figure path

5 Support and Open Source

This is a program which I've written for use in our lab. Since the release of the software, I've gone on to other things. While I'm perfectly willing to answer an occasional question, I'm not able to provide custom programming changes and alterations. That is why I'm distributing source code so you can make the changes you need to make.

If you do make changes which you think are sufficiently general enough to warrant inclusion in the release, please contact me and we'll talk about incorporating them. Examples of the kind of thing I'm talking about here are new database connector classes, or makefiles for other platforms (or even better ones for UNIX).

6 Acknowledgements

ComboScreen owes a great deal to the suggestions and patient beta testing of Jim Thomas. Others in the lab who provided guinea pig services were Johanna Doyle, Val Braden, and Jackie Idol. Steve Beckstrom-Sternberg provided help in tweaking the SQL queries and proof-reading the documentation.

7 Appendix

7.1 Experimental design

ComboScreen was written to support the work-flow in our lab. Specifically, it was written to support the construction of a human-mouse comparative map for human chromosome 7. Thus, the program reflects strongly the experimental design of the project, and it might be illustrative to describe roughly the process we use in order to better understand what the ComboScreen program is doing.

To begin, a collection of mouse ESTs or gene sequences are used to create overgo probes: A pair of oligonucleotides of 36 bases in length which overlap by 8 bases are used to generate radio-labeled probes by PCR. The probes are pooled into batches of 40 or more and hybridized to the library filter. The positive clones are then picked into 96 well plates, grown up, and then spotted in duplicate on filters for a secondary screen.

Meanwhile, the probes are subdivided into secondary pools such that any two pools will have only one probe in common. Each of the secondary pools is hybridized to the secondary filter, and then compared pairwise. The common overgo probe is deemed to have hybridized to common positive clones. This information is then used to create the sequence ready BAC map.

More information about creating and analyzing multiplex screens can be found in reference [Evans and Lewis, 1989] and [Cai et al., 1998].

References

- [Smart, 1999] Smart, J. wxWindows 2.0.
<http://www.wxwindows.org>
- [Evans and Lewis, 1989] Evens, G. A. and Lewis, K. A. Physical Mapping of Complex Genomes by Cosmid Multiplex Analysis. *Proc. Natl. Acad. Sci. USA* **86**, 5030-5034.
- [MacPherson, 1998] McPherson, J. D. OvergoMaker 4.0 overgo designer program. Available for use or ftp at <http://genome.wustl.edu/gsc/overgo/overgo.html>
- [Cai et al., 1998] Cai, W., Reneker, J., Chow, C., Vaishnav, M, and Bradley, A. An Anchored Framework BAC Map of Mouse Chromosome 11 Assembled Using Multiplex Oligonucleotide Hybridization. *Genomics* **54**, 387-397