John Clayton England, III

A Robust User Interface to the Stanford Microarray Database (SDM)

M.S. Pilot Adviser: M. W. Berry

April 10, 2003

This project is an add-on to the Stanford Microarrary Database (SMD) to allow easier file transfers from a user location to the database. The data stored in the database consists of scanned images of slides containing genomic data. This genomic data is collected from living organisms treated with color coded dyes and separated and placed on a slide. This slide is scanned and an image is created with the spots on the graphic varying from green to red. These images are overlaid showing which spots show similar traits with a yellow color. This combined image is uploaded to the database. With the standard SMD install this is done through a UNIX account interface. The goal of this project is to add a web based upload system and remove the need to have a UNIX account for each user.

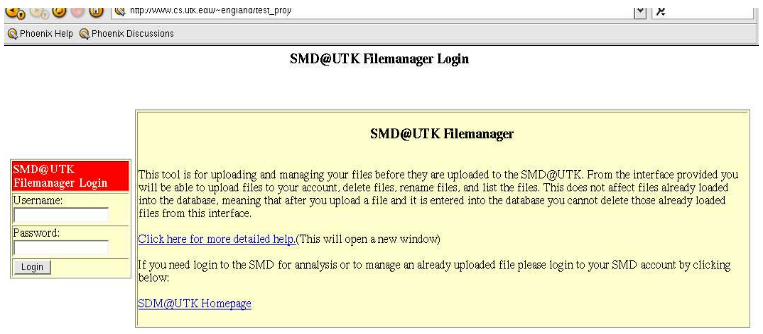
The SMD is a collection of perl scripts which are run using CGI (Common Gateway Interface) on a web server. The scripts provide a web-based interface to the SMD so that users can, from a web browser, analyze, catalog, and compare experiments. The install of the software is rudimentary as compared to modern UNIX software installs. Writing scripts to aid in the installation was helpful in speeding up the install, since many steps had to performed by hand previously. The install of the SMD did not function correctly upon installation due to differences in the software deployment at Web Services. Namely, the Oracle database the system required was of a different version than the one SMD was known to work with at the time. After changing the Oracle install it was possible to connect the database from the SMD interface. Other problems encountered were errors in the SMD code; most of these were corrected by finding solutions on the web, others were worked out by Colton Smith, who works at Web Services.

Once the install was complete and in operation, it was possible to determine how users were able to get data in to the database. The users would scan their genome chips or order premade chips and scan those chips. Next the users would analyze the chip scans with a software tool, usually with ScanAlyze (http://rana.lbl.gov), the images obtained from the scanned chips are analyzed with these tools to mark errors on the chips, and set a grid that matches the spots on the scanned image. Then the users would copy their files onto a disk and give the disk to the administrators at Web Services. The administrators would then place the scanned images and the grid files in the proper locations in the file system. The data can then be loaded into the database. After the data has been loaded, the users can use the SMD web interface to analyze their data.

This is not how Stanford suggests data insertion should be done. At Stanford the machine that holds the filesystem where the data is loaded also contains the database. Users are given a UNIX account on this machine and must transfer their files to the UNIX account. This requires the user to understand, and be proficient with, UNIX file transfers. While this is not unreasonable, it seems one could have a much wider audience if file transfers were done through a web interface like all other functions in the SMD. At the University of Tennessee we have two machines to provide these functions. One machine holds the filesystem where the data is uploaded and the other holds the Oracle database. This setup requires each user to have an account for the SMD which includes an Oracle account and an account on the machine holding the data filesystem. The extra UNIX accounts can cause overhead for the administrators along with adding security concerns on the machine running the SMD web-server.

The goal was to remove the need for an extra UNIX account and make a much easier user interface for file transfers. After observing how the Oracle database interacted with the SMD database client it was clear that the Oracle database did not depend on having local UNIX users. As long as the data files are in the proper place it does not matter whether the files are owned by the user who uploaded the files or another user. This is because the scripts that gather the data for inclusion in the database being as root, so the local owner does not matter. The UNIX username is not used to obtain the username for the Oracle database, so it is clear the UNIX account is only there to provide an interface to the filesystem. It was clear that the UNIX accounts could be removed if files could be placed on the filesystem in the proper place by another means. A web interface would require additional software to be installed on the web server to ease the file transfers. PHP was chosen as the scripting language to facilitate these file transfers.

PHP is a scripting language that runs on the web server, allowing scripts to be written inside of the web page while not allowing the scripts to be seen on the client web browser. The scripts that make up the file manager for the SMD consists of a login page, file upload page, file delete page, rename file page, and a file listing page. The login page (see Figure 1.) contains a username and password form that allow a user to access the other functions of the file manager. To use the login page you need to have a username and password; this will be assigned by the SMD administrators. A script has been written that will make a password file for the SMD filemanager login. The script takes a username and password as input, and as output it produces a password file containing a username and an MD5 password hash. The hash is used so that the password will not be stored in clear text on the filesystem. The password is sent in clear text once then the user logs in, then it is converted to a MD5 hash and checked against the stored hash. This could be overcome if the web server was recompiled with SSL support, adding encryption to all access of the SMD web pages. When the login is complete the username is carried through all other loaded pages to keep track of the user.


http://www.cs.utk.edu/~england/test_proj/

SMD@UTK Filemanager Login

SMD@UTK Filemanager Login
Username:
Password:
Login

SMD@UTK Filemanager

This tool is for uploading and managing your files before they are uploaded to the SMD@UTK. From the interface provided you will be able to upload files to your account, delete files, rename files, and list the files. This does not affect files already loaded into the database, meaning that after you upload a file and it is entered into the database you cannot delete those already loaded files from this interface.

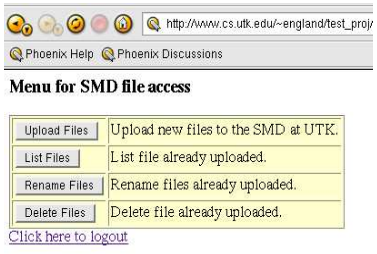
Click here for more detailed help.(This will open a new window)

If you need login to the SMD for annalysis or to manage an already uploaded file please login to your SMD account by clicking below:

SDM@UTK Homepage

Figure 1 Login Page

After login, the user is taken to the main menu page (see Figure 2). This page consists of the choices available to make use of the file manager. The upload option (see Figure 3) takes the user to a page where local files on the user's machine can be selected and uploaded to the filesystem. After a file is selected and the upload button has been selected the script checks the filesystem for the proper directory structure. If the directory structure exists the file is uploaded. If the directory structure does not exist it is created as the script runs and then the file is uploaded. This allows the filesystem to be completely cleared, and the file manager will still function since the directories wil be recreated when the next upload is attempted. The file is uploaded to a temporary directory and the file name is checked to see if it already exists on the filesystem; if it does exist, an error is returned. The user can either do nothing since the file is already uploaded, or can then delete or rename the file on the filesystem and upload the file again. After the file has been checked for existence, the temporary file is checked to make sure it was uploaded from the web interface and not placed there by other means by way of a built in PHP function. Then the file is moved to its final place.

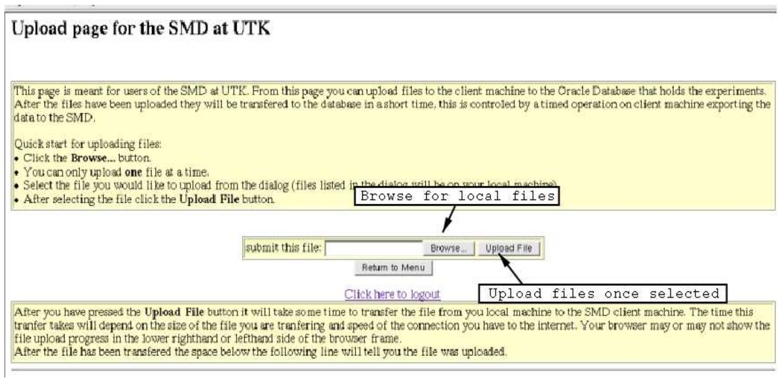

http://www.cs.utk.edu/~england/test_proj/

Menu for SMD file access

Upload Files	Upload new files to the SMD at UTK.
List Files	List file already uploaded.
Rename Files	Rename files already uploaded.
Delete Files	Delete file already uploaded.

Click here to logout

Figure 2 Main Menu



Upload page for the SMD at UTK

This page is meant for users of the SMD at UTK. From this page you can upload files to the client machine to the Oracle Database that holds the experiments. After the files have been uploaded they will be transfered to the database in a short time, this is controled by a timed operation on client machine exporting the data to the SMD.

Quick start for uploading files:
- Click the **Browse...** button.
- You can only upload **one** file at a time.
- Select the file you would like to upload from the dialog (files listed in the dialog will be on your local machine).
- After selecting the file click the **Upload File** button.

After you have pressed the **Upload File** button it will take some time to transfer the file from you local machine to the SMD client machine. The time this tranfer takes will depend on the size of the file you are tranfering and speed of the connection you have to the internet. Your browser may or may not show the file upload progress in the lower righthand or lefthand side of the browser frame.
After the file has been transfered the space below the following line will tell you the file was uploaded.

Figure 3Upload Page

Once a file has been uploaded, the user may choose to list the files they have uploaded. This listing shows the file name, size of the file, and the date of the last change to the file. This date will reflect the day the file was uploaded or the date the file was moved. From the file list page you can only logout or return to the main menu. By returning the main menu you can choose any of the functions listed above. Deleting a file (see Figure 4) is done through a simple interface: a drop down menu will allow the user to pick only files that exist on the filesystem. Once the file is selected, the delete button can be clicked to remove the file. After the button is pressed the file name is checked just to make sure the file does exist then it is removed. A page is then displayed showing a message about the completion of the deletion or an error if the file does not exist. This page will also allow the user to logout or return to main menu. The rename page is very similar with a drop down list to select the file to rename and an input box to choose a new name. When the rename button is pressed, the existence of the file name in the drop down is checked and the non-existence of the filename in the input box is checked. If both tests pass the file is renamed. There is one other check in place for renaming: renaming is the only time a user is allowed to type in a filename to change the remote filesystem.

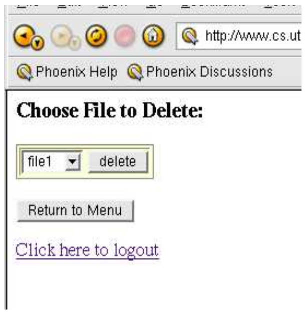


Figure 4 Delete Page

Allowing a user to write a file name could cause an issue with what is typed in the input box (see Figure 5). A user could try to write a file to another part of the filesystem by using input like **../../../filename**, wanting to go up three directories and write a file to a place they were not meant to use. This would probably fail since the PHP scripts run as a certain user (not root) who would most likely not have permission to write to that directory. A user could try to write to another user's space by giving a file name like the following: **../username/file**. This would succeed since the PHP script has the ability to write to any user's area by the design of the file manager. This is taken care of by using the basename function which will strip off everything before the last slash, leaving just the file name. If this were tried, the file would be written with the filename given at the end of the line but it would be written to the logged-in user's area. These rudimentary checks should keep a user's files safe on the filesystem.

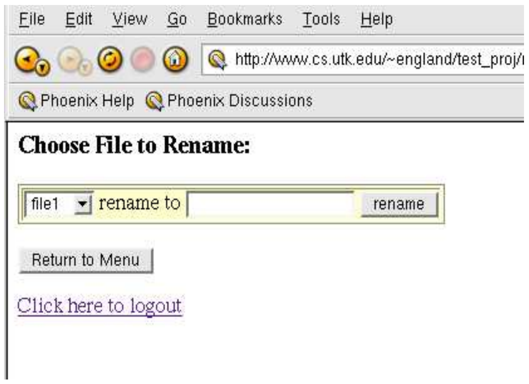


Figure 5 Rename Page

This file manager should allow a user to use only a web browser to upload files, and use the SMD for analysis of those files while taking the need for a UNIX account away from the SMD system. Other upgrades that could be made to the system are multiple uploads, meaning more than one file at a time for upload. If it is found that users need to upload many files at one time, this should be considered. Also enabling SSL on the web server would add some security to the system as a whole. Lastly, it might be possible to integrate the login of the file manager with the SMD login to give a single sign-on to the system.

References:

G. Sherlock, T. Hernandex-Boussand, A. Kasarskis, G. Binkley, etal.
The Stanford Microarray Database, *Nucleic Acids Research* 29:1:162-155, 2001

S. H. Friend and R. B. Stoughton.
The Magic of Microarrays, Scientific American, February 2002, pp44-49

Web References:

The Stanford Microarray Database:
http://genome-www5.stanford.edu/MicroArray/SMD/

The UTK install of SMD
http://genome.ws.utk.edu/

ScanAlyze
http://rana.lbl.gov

PHP
http://www.php.net

Perl
http://www.perl.com

Software from the project
http://www.cs.utk.edu/~england/SMDFM/