

CPLOP

Cal Poly's Library of Pyroprints

Senior Project Guide

Kevin Webb , CPE Fall 2011

About this Guide:

This guide will be broken into two parts, one geared for a biology student end-user; and another for a php web programmer to add additional functionality to this project in the near future. It will be important for the programmer to read both parts, as the database will be described abstractly in the beginning.

Thanks and Credits:

Before we begin, I want to thank Alexander Dekhtyar for starting this project in CPE 366 and advising me throughout its progress. I want also thank Christopher Kitts and Michael Black (Biology Department) for their time answering questions, using laymen's terms, that brought this project from a concept into a production reality.

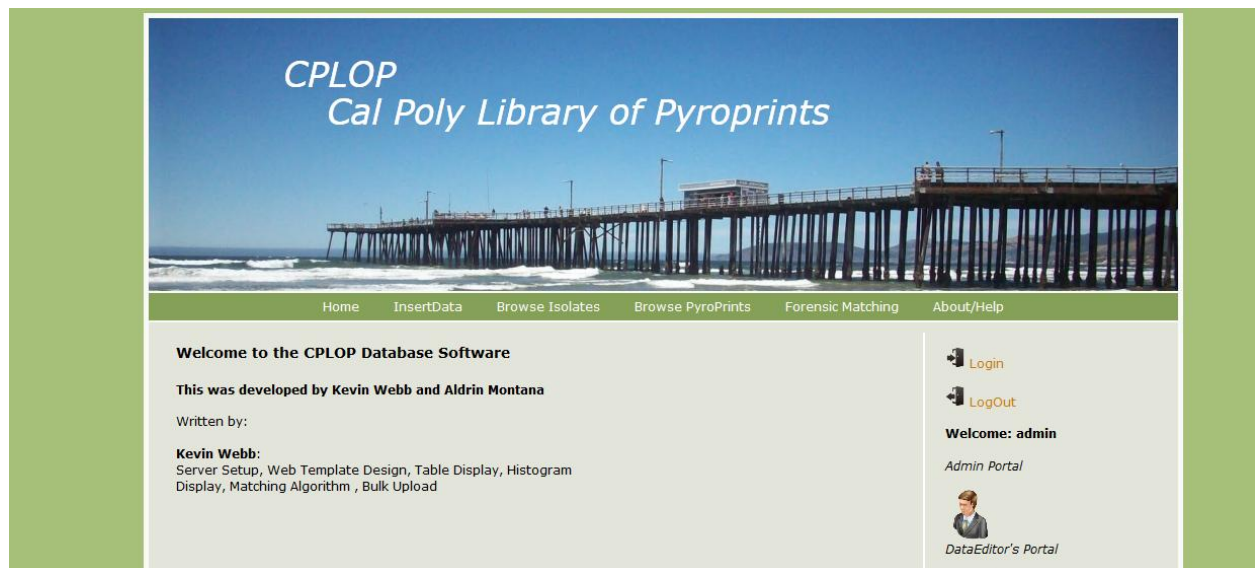


Figure 1 Main Page of CPLOP

Part 1: An End-User's Guide

California Polytechnic Library of Pyroprints, CPLOP, is a web driven data-base application that stores data from the biology's departments Pyro sequencing machines. This data was typically stored in excel datasheets, with no real way to perform complicated searches, organize, or store massive amounts of additional data from multiple file sources in one location. CPLOP's primarily goal is to store such data in many organized tables that relate to one another. It was designed to hold data ranging from Host Species, to Samples, to Isolates, all the way to Pyro prints. The details of each table will be documented below.

WebSite Address: <http://cplop.cosam.calpoly.edu/>

The Actual Data:

Before we go into all of the web site's features, it is very important to understand how this database is organized. For example, one such table is Isolates. It links data between the Samples and Pyroprint tables. Therefore the end user will understand how a Sample ID in the Samples table relate to the same Sample ID found in the Isolate's table.

Table 1: Host Species Table

This table holds the common name of a sample, its latin name, and letter code. This allows other tables, such of samples, to pull data about an Animal from this table.

Common Name	Latin Name	Letter Code
Cat	Felis	Ct

Table 2: Host Table

This table holds relational data between itself and the Samples Table. It holds the common name of a sample, and a host ID number – which identifies unique Hosts.

Common Name	Host ID
Cat	10

Table 3: Samples Table

This table contains unique entries for each Host that was sampled. A host, for example host ID 10, could have multiple samples. Also the table host the relating data common Name, host Id Sample ID, and date/location data about the sample.

Sample ID	Common Name	Host ID	Location	Date Collected
3	Cat	10	SLO	10/10/2010

Table 4: Isolate Table

This table is also known as the freezer stock table. It contains additional sample information separated into unique Isolates. The last three columns relate to the previous tables.

Isolate ID	Freezer Location	TANotes	Pyro Printed Date	User who Pyroprinted	Sample ID	Common Name	Host ID
Ct-001	Fridge 1	"Went Well"	10/15/2011	Kevin Webb	3	Cat	10

How Are Tables 1-4 Uploaded? (Part 1 of 3 of the Uploader)

The file uploader is a feature of the website that allows lab data (typically excel files) to be uploaded into the database. Tables 1-4 are populated by one file, the pyroprint master record. This is also known as the isolate file. This section is part 1 of 3 of the upload section.

Table 5: Pyroprint Table

The previous tables, in essence, divide animal samples into unique isolates or samples. Each isolate can be pyro sequenced many times thru the lab's pyrosequencer based on the well the pyroprint came from. This pyrosequencer returns an xml file of peak/ compensated heights per nucleotide sequence. That data is stored in the histogram table. This table, the pyroprint table, uniquely identifies a sequence. Hence, a isolate comes from a sample, which comes from a Host. A Pyroprint comes from a Isolate. Each isolate can have many Pyroprints because it is sequenced in many wells.

PyroID (Computer Generated)	fileName	pcrDate	Well ID	pcrMachine	Seq Machine	Dispensation Name	Forward Primer
1	GatcMOD	10/15/2011	A1	1	1	Disp 1	For 1

(Table Continued)

Rev Primer	Sequence Primer	PyroPrinted Tech	Isolate ID
Rev 1	Seq 1	Kevin Webb	Ct-001

How is the Pyroprint Table Uploaded (Part 2 of 3 of the Uploader)?

The pyroprint table data comes the master record pyroprint file. This will be detailed in part 2 of the bulk uploader.

Histogram 6: Histogram //Compensated Slope Table

The xml file data for each pyroprint is placed in Histogram table. The data is collected by the pyrosequencer. This table relates to the pyroprint table by the field “pyroID”. Each pyroprint entry will contain between 100 – 200 nucleotides. Each nucleotide sequence will have data associated to it. Recall that part 3 of the upload sequence adds data to this table.

PyroID	Position	Nucleotide	Peak Height	Compensated Peak Height	Peak Height	Peak Width	Baseline Offset	Signal Value
1	1	A	.67	1.0	2.4	4.5	5.3	1.1
1	2	A
1	...	C						
1	100	G		Data Differs	By	Each	Nucleotide	
1	200	A						

Other Tables

The following tables are for reference by the lab student. They are individually added one by one using the online software. They are not “bulk uploaded” in any sense. An example includes the dispensation sequence table. Only three sequences have been added and therefore do not need this added functionality.

Dispensation Table

Dispensation Name	<u>Sequence</u>
ModGATC-2c	AACACGCGAGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCG...

Primer Table

Primer Name	Sequence
23-5ITS-F	ATGAACCGTGAGGCTTAACCTT

Users Table

For the sake of the end user, this table contains the user names and security information to log someone into CPLOP’s system. For an administrator, he/she could add or delete users that can use CPLOP’s system.

Using CPLOP's System

How to Log on

There are three types of users. A guest, a data editor, and an administrator. A guest can only view the isolate and pyroprint table. A data editor can view, edit, and add to all tables except the users table. The administrator can view, add, edit, and delete all tables. Both a data editor and administrator can use the extra features, such as viewing the Histogram of a pyroprint.

A user can log in by accessing the main page using internet explorer or firefox at

<http://cplop.cosam.calpoly.edu/>

The user will hit the “Log in Icon” and follow the steps shown in the pictures below.

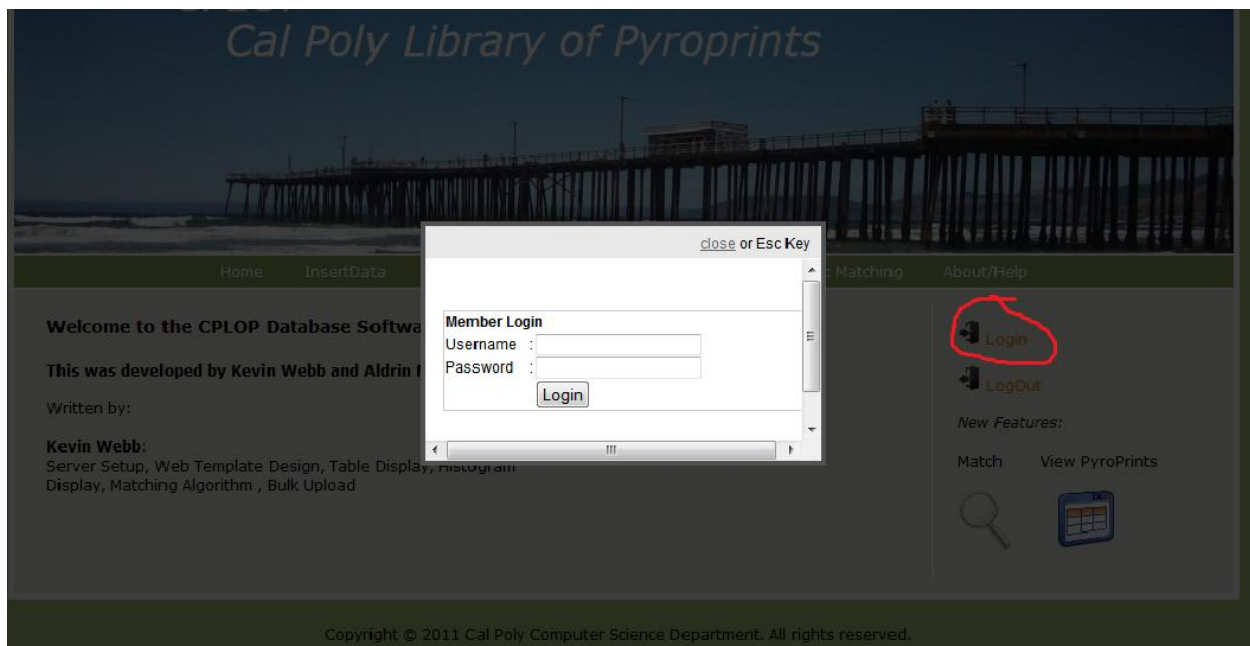


Figure 2 Step 1 of Logging In

Once logged in, a dataEntry user will see the data Entry portal button. An administrator will see both the dataEntry portal and administrator portal shown in the below picture.

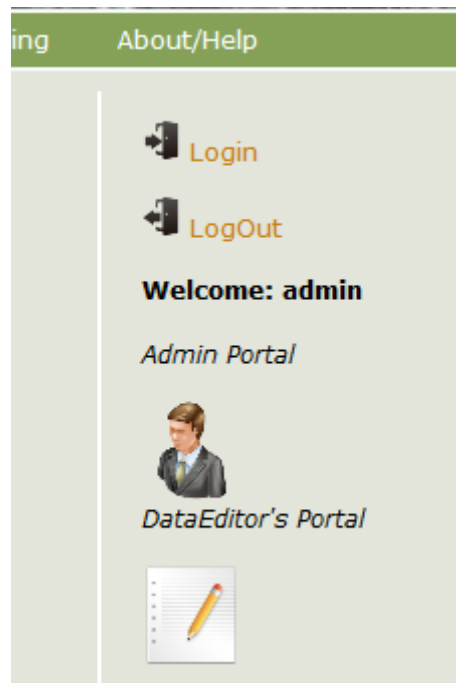


Figure 3 Levels of Login Access

Data Editor Portal

The DataEditor's portal provides access to the following:

- *Insert Data – Inserting Data for tables 1 – 6 thru the Bulk Upload Process.
- * Manage Host Species Data –View/ Add/Edit Data to the Specified Table
- * Manage Sample Data – View/Add/Edit Data to the Specified Table
- * Manage Primer Data –View/ Add/Edit Data to the Specified Table
- * Manage Sample Data – View/ Add/Edit Data to the Specified Table
- * Manage Dispensation Sequence Data – View/ Add/Edit Data to the Specified Table

Note that Administrators can also delete from the above tables.

- *Change My Password -- Changes the Password of the current user

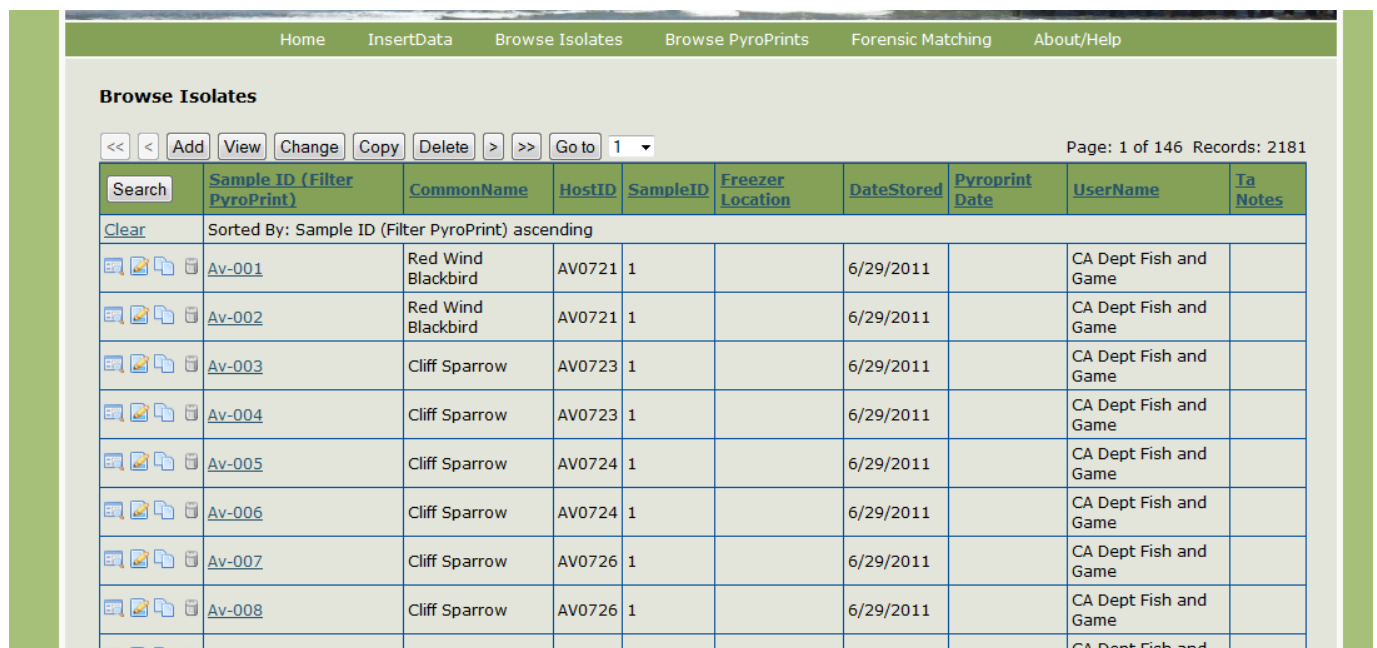
* Check Histogram State -- Tells the user how many entries are missing from the Histogram table.

Administrator Portal

Administrator users have access to modify (add/delete/view) from the user's table. They also can gain access to the backend database tools (phpMyAdmin).

Viewing CPLOP Data

Take for example the Isolate Table. A user may want to know what is in the freezer stock. They log in as a dataEntry user, hit the dataEditor portal button, and click on "manage Isolate table". The user will see the following interface.



The screenshot displays the 'Browse Isolates' interface. At the top, there is a navigation bar with links: Home, InsertData, Browse Isolates, Browse PyroPrints, Forensic Matching, and About/Help. Below the navigation bar, the title 'Browse Isolates' is followed by a set of action buttons: <<, <, Add, View, Change, Copy, Delete, >, >>, and a 'Go to' dropdown menu set to '1'. To the right of these buttons, it says 'Page: 1 of 146 Records: 2181'. The main data is presented in a table with the following columns: Search, Sample ID (Filter PyroPrint), CommonName, HostID, SampleID, Freezer Location, DateStored, Pyroprint Date, UserName, and Ta Notes. The table is sorted by 'Sample ID (Filter PyroPrint) ascending'. The first eight rows of data are visible, showing various bird species like Red Wind Blackbird and Cliff Sparrow, all with a date stored on 6/29/2011 and associated with 'CA Dept Fish and Game'.

Search	Sample ID (Filter PyroPrint)	CommonName	HostID	SampleID	Freezer Location	DateStored	Pyroprint Date	UserName	Ta Notes
Clear	Sorted By: Sample ID (Filter PyroPrint) ascending								
	Av-001	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	
	Av-002	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	
	Av-003	Cliff Sparrow	AV0723	1		6/29/2011		CA Dept Fish and Game	
	Av-004	Cliff Sparrow	AV0723	1		6/29/2011		CA Dept Fish and Game	
	Av-005	Cliff Sparrow	AV0724	1		6/29/2011		CA Dept Fish and Game	
	Av-006	Cliff Sparrow	AV0724	1		6/29/2011		CA Dept Fish and Game	
	Av-007	Cliff Sparrow	AV0726	1		6/29/2011		CA Dept Fish and Game	
	Av-008	Cliff Sparrow	AV0726	1		6/29/2011		CA Dept Fish and Game	









Figure 4 Example of Viewing CPLOP's Table Data

A user can hit the search button, and enter into the text fields that discriminates entries based on that input. The following picture shows isolates being discriminated by the "common Name" and "date" columns.

Home InsertData Browse Isolates Browse PyroPrints Forensic Matching About/Help

Browse Isolates

<< < Add View Change Copy Delete > >> Go to 1 Page: 1 of 1 Records: 2

Hide	Sample ID (Filter PyroPrint)	CommonName	HostID	SampleID	Freezer Location	DateStored	Pyroprint Date	UserName	Ta Notes
Query		Blackbird				6/29/2011			
Clear	Sorted By: Sample ID (Filter PyroPrint) ascending								
Clear	Current Query: (' PMETable0`.`commonName` LIKE '*Blackbird*') AND (' PMETable0`.`dateStored` LIKE '*6/29/2011*')								
   	Av-001	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	
   	Av-002	Red Wind Blackbird	AV0721	1		6/29/2011		CA Dept Fish and Game	

6.742 milliseconds

Copyright © 2011 Cal Poly Computer Science Department. All rights reserved.

Figure 5 A Search Example

The user can also click on the buttons on the lefts that edit/changes/modifies the individual entry.

View - Looks at a single entry horizontally.

Change - Allows user to edit any individual field of the data.

Delete - Allows user (admin) to delete an individual entry.

Home InsertData Browse Isolates Browse PyroPrints Forensic Matching About/Help

Browse Isolates

Save Apply Cancel

Sample ID (Filter PyroPrint)	Av-001
CommonName	Red Wind Blackbird
HostID	AV0721
SampleID	1
Freezer Location	
DateStored	6/29/2011
Pyroprint Date	
UserName	CA Dept Fish and Game
Ta Notes	

1.995 milliseconds

Copyright © 2011 Cal Poly Computer Science Department. All rights reserved.

Figure 6 Editing an Example Isolate Entry

Special Table Features

All tables can use the same interface shown above to modify CPLOP's data. The isolate table has each Isolate ID as a hyper link to search the pyroprint table for all pyrprints that are linked with that particular Isolate ID. The pyroprint table's "pyroID" are hyperlinks to a gathering page that allows the user to perform special features (like displaying graph data) for that particular pyroID.

The following pictures show how the user can use the Isolate table to display all "Sg-027" isoaltes in the pyroprint table. We can also display Histogram data for all pyroprints linked to that isolate. This is shown in the pictures below.

Browse Isolates

Page: 1 of 1 Records: 1

Hide	Sample ID (Filter PyroPrint)	CommonName	HostID	SampleID	Freezer Location	DateStored	Pyroprint Date	UserName	Ta Notes
Clear	Sg-027								
Clear	Sorted By: Sample ID (Filter PyroPrint) ascending								
Clear	Current Query: ('PMETable0'.isoID LIKE "Sg-027")								
	Sg-027	Seagull	6	1		8/12/2010		Vanessa Castillo	12/13/2010

4.211 milliseconds

Figure 7 Searching Isolates for SG-027 and Using Link Button

Browse Pyroprints

This table shows all Pyroprints linked to Isolate ID 'Sg-027'

Page: 1 of 1 Records: 1

Search	PyroID (Options)	IsoID	AppliedRegion	DsName	ForPrimer	RevPrimer	SeqPrimer	FileName	WellID	PyroPrintedTech	PyroPrintedDate
Clear	240	Sg-027	23-5	ModGATC-2 contr	23-5ITS-F	23-5ITS-R-biotin	23-5ITS-S	12/13/10 ModGATC-2 controls	B5	VanderKelen	12/13/2010

3.462 milliseconds

Copyright © 2011 Cal Poly Computer Science Department. All rights reserved.

Figure 8 Viewing All Pyroprints Linked by Isolate Sg-027 (Found PyroID 240)

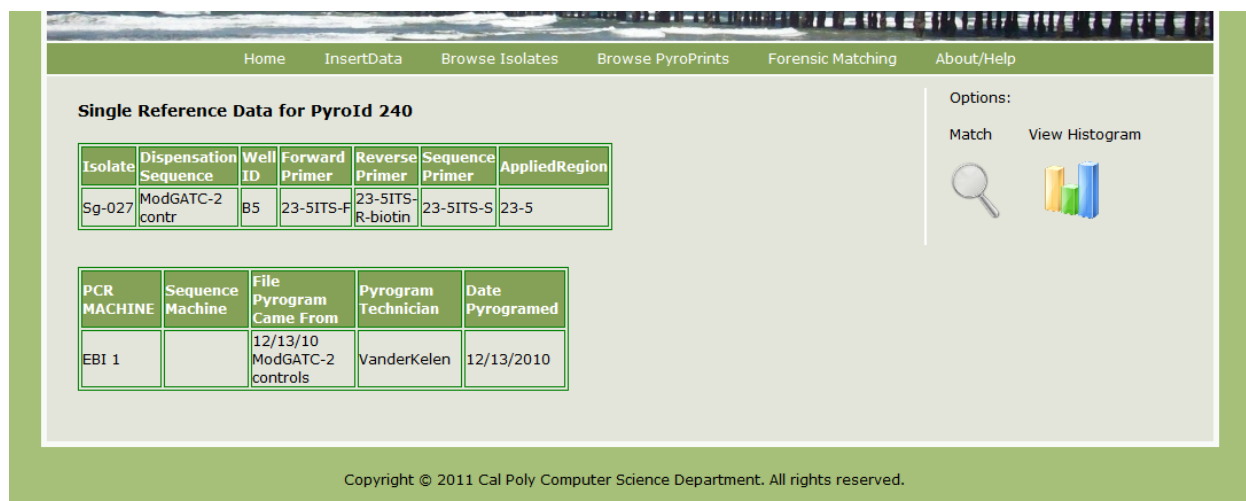


Figure 9 Pyroprint Features of PyroID 240

**Please note that the matching functionality will be completed by another student in the near future. My senior project focused on creating the database, upload functionality, and providing a base for future features. **

Uploading Data to CPLOP

Method 1) Manual Entry – Tables such as Primers or Users are not reguuarly added to. Therefore the user can use the datagrid viewing interface, shown above, to inividually add an entry. This feature is shown below for the isolate table by clicking the “Add button” on the top of the table. By clicking the save button, the data entered will be saved to the database as another entry. If an entry is left blank, the table will show nothing in that field.

The screenshot shows the 'Browse Isolates' form, which includes a table with input fields for various isolate attributes. At the top of the form are buttons for 'Save', 'More', and 'Cancel'. The table has the following fields:

Sample ID (Filter PyroPrint)	<input type="text"/>
CommonName	<input type="text"/>
HostID	<input type="text"/>
SampleID	<input type="text"/>
Freezer Location	<input type="text"/>
DateStored	<input type="text"/>
Pyroprint Date	<input type="text"/>
UserName	<input type="text"/>
Ta Notes	<input type="text"/>

1.575 milliseconds

Method 2) Bulk Upload—The majority of the data will be held in the isolate, pyroprint, and histogram tables. Both the isolate and pyroprint tables are added via master excel sheets found on the lab computer. The histogram table's data are added via a .pyrorun xml file produced from the pyrosequencer machine.

The whole process takes three steps. Step One reads a excel file (in the .csv format) that adds Isolate data. Step two reads the data from another excel file (in the .csv format) which adds pyrolds to the pyroprint table linked by an Isolate ID. Step Three takes the .pyrorun file and adds histogram data to the histogram table. This file contains information about the wellID and filename. A pyroID is linked to a histogram file by the unique wellID and filename of the uploaded .xml file. Additional data about file requirements can be found at

<http://cplop.cosam.calpoly.edu/insertdata.php> (Login Required)

The following pictures serve as examples of the three step bulk uploading process:

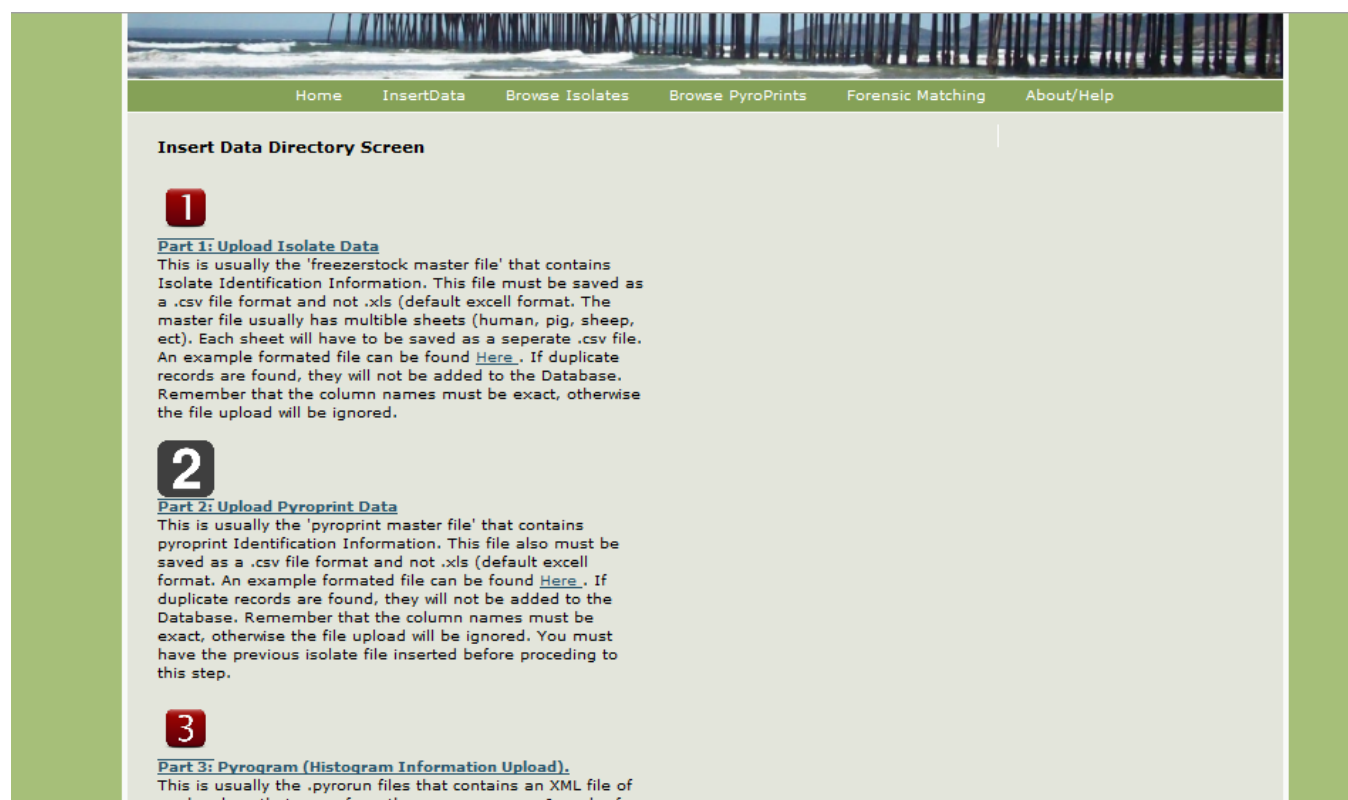


Figure 10 Main Screen of 3 Step Bulk Uploader Process

Upload Isolate Information (.csv)

Name

Your email address

Choose a file to upload

Browse...

Submit

Figure 11 Example Step One's File Uploader

Part 2: A Programmer's Guide

This part is geared toward a programmer that already has php, html, and mysql experience. This guide is to give a head start about the structure of the CPLOP system so a programmer could add the additional functionality. This guide also assumes the programmer read part one of this guide so he/she understands the database structure and how the tables are linked.

Overview of System

CPLOP is sitting on a virtual machine on the College of Math and Science Server. This is a Ubuntu server (Version 10) running php 5 and mysql. The php.ini file is located at `/etc/apache2/` and the entire website is sitting at `/var/cplop/`. A person can "ssh" into the machine at `cplop.cosam.calpoly.edu`.

The server's administrator is Jon Sehmer (`jsehmer@calpoly.edu`)

Each .php file represents a certain website. Index.php is the page's home address. The upload folder is a temporary storage location for the uploaded file of the bulk uploading process. The file "functions.php" provide the programmer with functions that automatically connect to the server's mysql database. (The mysql username//pass are in this file). In addition, it provides tools such as "isAdmin() or isDataEditor()", when combined with the session_start() can tell if the user has logged in or not. A session lasts for 30 minutes. After that, a person is automatically logged out. All of the tables above are in the CPLOP database. The user table is in the User database. A user can be of three levels. Level 1 is admin. Level 2 is dataEditor (can view/edit/add data). Level 3 is guest, anybody who is not logged in.

Example of a Blank Page (Where to Start with Adding Functionality)

The following code segment represents a blank typical page of CPLOP, using the default html template. It also shows how the programmer can check the user's credentials and parse data from the mysql database. If I were to begin adding the match functionality, I would start editing the match.php file with the following code.

//Example Match.php (Blank Template)

<?session_start(); //Start the Session ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href="style.css" rel="stylesheet" type="text/css" />

<title>Cal Poly Library of Pyroprints</title>

</head>

<script type="text/javascript" src="/cplop/ThickBox/jquery-latest.js"></script>

<script type="text/javascript" src="/cplop/ThickBox/thickbox.js"></script>

<link rel="stylesheet" href="/cplop/ThickBox/thickbox.css" type="text/css" media="screen" />

<body>

<div id="container">

 <div id="header">

 <div id="logo_w1">CPLOP</div>

 <div id="logo_w2">Cal Poly Library of Pyroprints </div>

<?php include("functions.php"); webSiteLinks(); //Prints out theLinks ?>

 </div>

 <div id="content">

 <div id="left">


```

<?php

//SECTION TO DO YOUR WORK

$isAdmin = isAdmin(); //Checks if User is Admin (Called from Functions.php above)

if(!$isAdmin)

{

    echo("<H1>You do have a high enough access level to modify the CPLOP system or have
not logged in. <br> Access Error 02 <H1><br>Log in: <a
href=\"http://cplop.cosam.calpoly.edu/#TB_inline?height=155&width=300&inlineId=hidden
Content\" class=\"thickbox\"><p><img src=\"images/login.png\" width=\"20\"
height=\"20\" alt=\"Log In !\" /></div>");

}

else

{

    //Left Section of the Page

    echo("<H1> Write on the Left Side of the Page</div>");

    //Right Section of the Page

    echo("<div id=\"right\"> Write on Right Side of Page</div>");

}

?>

<div id="footerline"></div>

</div> <!-- end content -->

<div id="footer">Copyright &copy; 2011 Cal Poly Computer Science Department. All rights
reserved.</div>

</div> <!-- end container -->

</body>

```

</html>

In the php work section, a user could use the following code to call a query to the MySQL DataBase.

//Start Code

<?php

//Connect to CPLOP DB

mysqlConnect("CPLOP");

\$result = mysql_query("Select count(*) as COUNT from Pyroprints");

while(\$info = mysql_fetch_array(\$result))

{

 \$returnedNum = \$info['COUNT'];

}

?>

=====

Included Functions in the Functions.php (Common API)

function mysqlConnect(\$tableName)

Connects to User or CPLOP depending on \$tablename. This will allow all calls to mysql_query to work.

isDataEditor()

Checks User Table if user has level 2 or greater access. True if so, false if not so.

isAdmin()

Checks User Table if user has level 1 or greater access. True if so, false if not so.

Void function webSiteLinks()

Lists (in format) the header links of the website. If a change is needed, for example adding another button next to InsertData, a field could be added. This function is included in any page that uses CPLOP's standard template.

function connectToDataBaseTableSoftware()

CPLOP uses a php script to generate the table software used on all of its pages. This script requires an array to connect with a user Name and password. This script fills these array with the correct mysql login information.

Int function pearson(\$sequenceOne, \$sequenceTwo)

CPLOP has a rudimentary match system. This feature needs to calculate the pearson's coefficient between two sets of similarly sized arrays. The return is between -1 and 1.

PME Table Package

A table script was used <http://www.phpmyedit.org/>. You can download its .php package, run it in a browser and it will generate all php code needed to display the tables live. The code it generates is the values the script needs to display one table, its permissions, ect. Examples can be found on browseIsolates.php or browsePyroprints.php.

Bulk Uploader

Each step of the bulk uploader executes a page (upload_1.php, upload_2.php, or upload_3.php) respectfully.

Upload_1 Parses .CSV File and uploads data into Host, Samples, and Isolate Table

Upload_2 Parses .CSV File and uploads data into Pyroprint Table. This creates a unique pyroID for each entry.

Upload_3 Parses XML File (.PYRORUN) and dumps data into compensated_slope and Histogram table. Each entry is linked to a pyroprint by the well and filename. This function calls Aldrin Montana's xml parser package, xmlparser.php .

PHPMyAdmin (DataBase view of Database//Create Statements)

PHPMyAdmin has been installed for the convenience of the programmer. The standard mysql username and password will grant you access to cplp.cosam.calpoly.edu/phpMyAdmin. A link can also be found in the admin portal section. Php My Admin allows the user to back up the databases, perform complicated sql queries, and check on the state of the database.

The following includes the CREATE_TABLE statements of SQL needed to Create the CPLOP DataBase.

```
--
-- Database: `CPLOP`
--
-- -----
--
-- Table structure for table `compensation_slope`
--

CREATE TABLE IF NOT EXISTS `compensation_slope` (
  `pyrogram_num` decimal(50,0) NOT NULL,
  `position` decimal(50,0) NOT NULL,
  `level` decimal(10,0) NOT NULL,
  `drop_off_value` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----
--
-- Table structure for table `Dispensation`
--

CREATE TABLE IF NOT EXISTS `Dispensation` (
  `dsName` varchar(15) NOT NULL DEFAULT '',
  `dispSeq` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`dsName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----
--
-- Table structure for table `Histograms`
--

CREATE TABLE IF NOT EXISTS `Histograms` (
  `pyroID` int(11) NOT NULL DEFAULT '0',
  `position` int(11) NOT NULL DEFAULT '0',
  `pHeight` decimal(8,4) DEFAULT NULL,
  `cPeakHeight` decimal(8,4) DEFAULT NULL,
  `nucleotide` char(1) DEFAULT NULL,
  `PeakArea` float NOT NULL,
  `PeakWidth` float NOT NULL,
```

```

    `BaselineOffset` float NOT NULL,
    `SignalValue` float NOT NULL,
    PRIMARY KEY (`pyroID`, `position`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `Host`
--

CREATE TABLE IF NOT EXISTS `Host` (
  `commonName` varchar(50) NOT NULL,
  `hostID` varchar(50) NOT NULL DEFAULT '',
  PRIMARY KEY (`commonName`, `hostID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `HostSpecies`
--

CREATE TABLE IF NOT EXISTS `HostSpecies` (
  `latinName` varchar(50) DEFAULT NULL,
  `commonName` varchar(50) NOT NULL,
  `letterCode` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`commonName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `Isolates`
--

CREATE TABLE IF NOT EXISTS `Isolates` (
  `isoID` varchar(15) NOT NULL DEFAULT '',
  `FreezerLocation` varchar(50) DEFAULT NULL,
  `TANOTES` varchar(300) DEFAULT NULL,
  `dateStored` varchar(15) DEFAULT NULL,
  `PyroPrintDate` varchar(50) DEFAULT NULL,
  `userName` varchar(50) DEFAULT NULL,
  `sampleID` int(11) DEFAULT NULL,
  `commonName` varchar(50) NOT NULL,
  `hostID` varchar(50) NOT NULL,
  PRIMARY KEY (`isoID`),
  KEY `commonName` (`commonName`),
  KEY `hostID` (`hostID`, `commonName`),
  KEY `sampleID` (`sampleID`, `commonName`, `hostID`),
  KEY `userName` (`userName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

-- -----

--
-- Table structure for table `Primer`
--

CREATE TABLE IF NOT EXISTS `Primer` (
  `primerName` varchar(40) NOT NULL DEFAULT '',
  `sequence` varchar(40) DEFAULT NULL,
  PRIMARY KEY (`primerName`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- -----

--
-- Table structure for table `Pyroprints`
--

CREATE TABLE IF NOT EXISTS `Pyroprints` (
  `pyroID` int(11) NOT NULL DEFAULT '0',
  `fileName` varchar(100) DEFAULT NULL,
  `appliedRegion` varchar(20) DEFAULT NULL,
  `pcrDate` varchar(15) DEFAULT NULL,
  `wellID` varchar(5) DEFAULT NULL,
  `pcrMachine` varchar(20) DEFAULT NULL,
  `sequencerMachine` varchar(20) DEFAULT NULL,
  `dsName` varchar(15) DEFAULT NULL,
  `forPrimer` varchar(40) DEFAULT NULL,
  `revPrimer` varchar(40) DEFAULT NULL,
  `seqPrimer` varchar(40) DEFAULT NULL,
  `pyroPrintedTech` varchar(100) DEFAULT NULL,
  `pyroPrintedDate` varchar(15) DEFAULT NULL,
  `isoID` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`pyroID`),
  KEY `dsName` (`dsName`),
  KEY `forPrimer` (`forPrimer`),
  KEY `revPrimer` (`revPrimer`),
  KEY `seqPrimer` (`seqPrimer`),
  KEY `pyroPrintedTech` (`pyroPrintedTech`),
  KEY `isoID` (`isoID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```
-- -----  
  
--  
-- Table structure for table `Samples`  
--  
  
CREATE TABLE IF NOT EXISTS `Samples` (  
  `sampleID` int(11) NOT NULL DEFAULT '0',  
  `commonName` varchar(50) NOT NULL,  
  `hostID` varchar(50) NOT NULL,  
  `location` varchar(50) DEFAULT NULL,  
  `dateCollected` varchar(15) DEFAULT NULL,  
  PRIMARY KEY (`sampleID`,`commonName`,`hostID`),  
  KEY `hostID` (`hostID`,`commonName`),  
  KEY `commonName` (`commonName`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;  
  
-- -----
```

The following includes the CREATE_TABLE statements of SQL needed to Create the “Users” DataBase.

```
--  
-- Database: `Users`  
--  
-- -----  
  
--  
-- Table structure for table `members`  
--  
  
CREATE TABLE IF NOT EXISTS `members` (  
  `id` int(4) NOT NULL AUTO_INCREMENT,  
  `username` varchar(65) NOT NULL DEFAULT '',  
  `password` varchar(65) NOT NULL DEFAULT '',  
  `type` int(4) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

The SQL Code has to be executed in the order above to perverse foreign and primary key constraints. Two DB’s are needed, CPLOP and Users as shown above.

Summary:

CPLOP is now at a functional state to upload, store, and display Pyroprint data to Cal Poly's Biology Department. Instead of using multiple excel files stored across many computers and folders, any student can access the data from any computer anywhere. My senior project's goals were to start the CPLOP system and provide a common base for future development. All of these goals have been met, and the result is a usable database content delivery system.