AvidaLab Software Toolkit

Requirements Analysis Document

Carlos A. Tijerina Ibarra
Luis A. Tijerina Ibarra

The University of Texas – Pan American
CSCI 6390 – Masters Project
Instructor: Dr. Laura Grabowski
Fall 2014

Table of Contents

1. Introduction

1.1 Purpose of the System

Avida (Ofria & Wilke, 2004) is a software platform for digital evolution, used for studying evolutionary processes relevant to both natural and artificial systems. Avida places a population of self-replicating computer programs, called *digital organisms* or *Avidians*, in a user-defined computational environment. Avida is not a simulation of natural evolution, but rather a separate instance of evolution in its own right: the digital organisms compete, replicate, and mutate, thus satisfying the fundamental requirements of evolution (Dennett, 2002). Avida is the most widely used digital evolution software in the world and its users encompass a wide range of backgrounds and expertise, including biology, computer science, engineering, and philosophy.

Because of the extensibility of the platform and the flexibility of system configurations, Avida experiments produce vast amounts of data that need to be processed. Avida contains some limited analysis tools, but individual researchers have to write most of their own analysis tools outside of Avida itself. This situation can be a problem for researchers who are not trained or experienced programmers, such as the biologists who use Avida. Writing tools for basic analysis and/or learning how to use existing tools can be time consuming for both programmers and non-programmers.

The AvidaLab Software Toolkit intends to address these issues by providing an intuitive interface that guides non-technical users through every step of the process of loading Avida data files into the program, manipulating this data, and providing a meaningful representation of experiment results, through the use of graphs and statistical operations.

1.2 Scope of the System

The AvidaLab Software Toolkit will provide the user with an easy to use, directly manipulable interface. This interface will allow the user to import a set of Avida experiment output files. Once the files are imported, the user will be able to perform common statistical analyses on the data and display a number of results in the form of graphs.

Graphs will be displayed and edited on AvidaLab's graphical user interface, and will be easily exported to PDF or image formats.

1.3 Objectives and success criteria of the project

The objectives of the AvidaLab project are to provide:

- Tools for analysis of Avida data, including importing data sets, calculating common statistics.
- Data visualization tools for analyzed data, such as line plots and box and whisker plots.
- A user-friendly environment that is accessible for any kind of researchers, programmers or non-programmers.
- An integrated software toolkit that is easy to install, configure, and manage.

The AvidaLab Software Toolkit seeks to produce results of analysis done on some initial set of Avida data that the user finds important. The software will provide such a solution for those who are running experiments and retrieving data from the Avida platform.

Success of the project will be judged on four different criteria:

- Data Import: Correct and manageable organization of the imported data. This includes parsing, object design, and on-screen representation of data fields.
- Statistical Analysis: Presence of useful and correct statistical operations to be applied on the experiment data.
- Graphing: Meaningful and consistent graphical representation of graphs. There should be a way to display all the requested kinds of graphs on the data the user selects.
- File management: Successful export of graphs and statistical data. Single install package for all the AvidaLab software tools.

1.4 Definitions, acronyms, and abbreviations

- **Mean:** The arithmetic average of all values.
- **Median:** The middle number in a given sequence of numbers.
- **Standard Deviation:** A measure of how spread out numbers are, the square root of the variance.
- **Variance:** A measure of how spread out numbers are, the square root of the variance.
- **Mann-Whitney U Test:** Also called Wilcoxon-Mann-Whitney test. It uses the ranks of data to test the hypothesis that two samples of sizes m and n might come from the same population.
- **Kruskal-Wallis Test:** Kruskal-Wallis one way analysis of variance, a non-parametric method for testing equality of population medians among groups.
- **Line Plot:** Shows data on a number line with x or other marks to show frequency.
- **Histogram:** A histogram is a graph of a dataset, composed of a series of rectangles.
- **Box/Whisker Plot:** A bar or diagram using a number line to show the distribution of data.
- **Scatter Plot:** Also called scatter diagrams, used to investigate the possible relationship between two variables that both relate to the same "event."
- **IDE:** (Integrated Development Environment)Includes a source code editor, compiler, and usually a debugger that all work together to build a software program.
- **SVN:** (Subversion) A version control system for code files and document files.
- **TortoiseSVN:** Version control software for Microsoft Windows. It is implemented as a Windows shell extension which makes it integrate into Windows explorer.
- **PDF:** (Portable Document Format) A multi-platform file format developed by Adobe Systems.
- **URL:** (Uniform Resource Locator) The address of a specific website or file on the Internet.
- **TGZ:** Unix .TAR file archive compressed with GNU zip compression.
- **TAR:** (Tape Archive) A Unix based utility used to package files together.
- **GNU:** A free Unix-like operating system distributed by the Free Software Foundation.
- **GPL:** General Public License.
- **JPG:** Also JPEG (Joint Photographic Experts Group) A compressed image file format.
- **EPS:** (Encapsulated PostScript) A PostScript file format that is compatible with PostScript printers and is often used for transferring files between various graphics applications.
- **PNG:** A compressed raster graphic format commonly used on the web and is a popular choice for application graphics.

- **MB:** (Megabyte) Roughly 1 million bytes or 1,024 kilobytes. A byte is a unit of measurement used to measure data.

1.5 References

- Dennett, D. (2002). The new replicators. In M. Pagel (Ed.), *Encyclopedia of Evolution*. New York, NY: Oxford University Press, E83-E92.
- Ofria, C., and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial Life , 10*, 191-229.

2. Current System

As of today, there is no system similar to the one proposed in this project, the AvidaLab Software Toolkit. Avida itself contains some limited analysis tools, but individual researchers have to write most of their own analysis tools outside of the Avida system. The computer scientists and engineers who use Avida do not generally find this situation particularly difficult, except that writing such custom tools is time-consuming. The situation is a larger problem for researchers who are not trained or experienced programmers, such as the biologists who use Avida. Learning the tools and programming skills required to write software for basic analysis is time consuming, and presents a significant obstacle for non-programmer users of Avida. Making this issue more challenging is the fact that most of the software tools used to produce analysis programs are heavily command-line based. Most non-programmer users are not familiar to working with command line interfaces, and the added complexity of learning a new interface scheme along with programming skills makes the task even more difficult to complete.

Another difficulty in Avida experiment analysis is that, in the past, most existing Avida data analysis software was written using Matlab, an expensive proprietary software product. More recently, Avida researchers have turned to Python and other freely available software to write analysis tools. Although this solution eliminates the need for high cost tools such as Matlab, it still fails to address the lack of tools available to non-programmers, or the potentially complex tool chains and configurations that may be required in order to use non-proprietary software.

3. Proposed System

3.1 Overview

The proposed software toolkit main feature will be a useful analysis tool for data files created by the Avida software platform after running a set of experiments. The software system will allow users to import data files generated from Avida and calculate common statistics through a user-friendly environment. In addition, the AvidaLab toolkit will offer visualization tools to analyze the data graphically, as well as numerically, such as line plots and box and whisker plots.

Other capabilities include a user-friendly environment that is accessible for users who are non-programmers and an integrated software toolkit that is easy to install, configure, and manage.

3.2 Functional Requirements

- Users must be able to import archived/zipped files, as well as regular files, with file sizes ranging from 50 MB to 250 MB (typically in tgz format).
- Archived files must be prepared for use.
- Users must be able to select files for processing, from a single file to a whole set of files in a treatment (up to 100 files).
- The software will import, organize, and display the data from the Avida data files.
- Users will be able to select columns of data from files to process.
- Users will be able to select preprocessing features, such as computing log values, before statistics or visualization.
- The software should not modify the user's data files.
- The user will be able to compute statistics and statistical tests on the aggregated data, including:
    - Mean
    - Median
    - Standard deviation
    - Mann-Whitney U test
    - Kruskal-Wallis one-way analysis of variance
- The user will be able create visualizations of data, including:
    - Line plots
    - Histograms
    - Box and whisker plots
    - Scatterplots
- The user will be able to export plots in a variety of file formats, including .eps, .pdf, .jpg, and .png.

3.3 Non-Functional Requirements

3.3.1 USABILITY

- AvidaLab will provide an intuitive graphical user interface.
- Since the target audience are non-technical users, AvidaLab will require minimal or no training before using the system, aside from knowledge on the Avida system and related research.
- A top-down/left-right design approach will be used.
- Actions and menus will be appropriately named and easily accessed.
- User interaction will be handled mainly through widgets (buttons, checkboxes, dialogs) with less emphasis on menus.
- A user manual will be provided with easy access through the system, and troubleshooting information will be available through the system.

3.3.2 RELIABILITY

- The system will provide mechanisms for the user to correct input errors without failure of the system.
- The system will provide validation for inputs from the user, and will provide meaningful error messages in the event of user input error.
- The system will provide backtracking on user actions to correct any accidental or erroneous changes.

- AvidaLab will produce correct output results.

3.3.3 PERFORMANCE

- The system should respond quickly to initial selections and provide feedback ahead of time for long running processes.
- The system will only load data when needed by user, and will delete any data stored in memory that is not being currently used.
- Long tasks should provide visual feedback of the progress.
- All user actions will provide immediate visual feedback on their results.

3.3.4 SUPPORTABILITY

- Troubleshooting will be provided through the AvidaLab user manual.
- The code and object design will be as readable and organized as possible, and will be made available to Dr. Grabowski for future projects.

3.3.5 IMPLEMENTATION

- AvidaLab will be implemented with non-proprietary software tools.
- AvidaLab will be implemented with an object-oriented, platform-independent programming language (such as Java or Python).
- AvidaLab will use freely available code libraries. The developer will reuse code for certain functions within the system, such as interfaces with certain kinds of file formats.
- Project documents will be produced in PDF format, and image formats such as JPG.

3.3.6 INTERFACE

- The AvidaLab toolkit will use a directly manipulable graphical user interface.
- The file import feature of AvidaLab will be wizard-based, guiding the user through each step helping to reduce user based error.
- The user will be provided a series of buttons, menus, and checkboxes. This will allow the user to identify the dependent and independent variables and select which operations and what kind of graphing to perform on them.
- Within this interface the user will have all abilities to bring in a set of data which has been compressed in Avida, extract all the data within that compressed file for analysis, and output those results to different formats.

3.3.7 PACKAGING

- AvidaLab will be made available for download.
- The system will be installed on the user's computer and will not require use of remote resources.
- Installation and configuration of the system will be simple and require a minimum of software needed outside of the system itself.
- The code and design documents will be made available to Dr. Grabowski for future reference and student projects.

3.3.8 LEGAL

- AvidaLab will be free software.
- The GNU General Public License (GPL) will govern AvidaLab, unless superseded by the license of one or more of the tools used to build the system.

3.3.9 SECURITY

- The system will be accessed by only one user on computer per copy of the software.
- AvidaLab, like Avida itself, does not handle sensitive information, nor does it have any network functionality, and does not require any special security measures.

3.4 Constraints

- The system will be freely available and platform independent.

3.5 Target Environment

- AvidaLab must be cross-platform compatible, so that it runs on a variety of operating systems, including Windows, Mac OS X, and Linux.

3.6 System Models

3.6.1 SCENARIOS

Creating a new project

Neil is a biologist who is currently researching evolution with Avida. He ran an experiment at work and has output data compiled from a number of runs. He has all the data in a zipped folder, since he had to move this data from his work PC. Neil runs AvidaLab on his home computer to run some analysis on this data set. The program displays the start screen which gives the options to “Create a new project”, “Select an existing project”, which shows recently used projects and the option to browse for a project file, and “Exit”. Neil chooses to create a new project; a dialog pops up prompting Neil to enter a name for the project and select which directory to save it in. After the program confirms that the file was successfully created, another dialog prompts Neil to select a directory, set of files, or zipped folder to input as a data set. Neil selects his zipped folder. The program confirms that the data set is valid and imports the data, displaying a progress bar that fills as the files get loaded. Once the files are loaded, Neil receives a confirmation showing all the loaded files and is taken to the next page that allows him to select graphing options and statistical operations.

Graphing and Performing Statistical Operations on Data

Neil needs to create a graph to best represent the results of his last Avida experiment. Since he previously loaded this data into a project, he runs AvidaLab and select this project from the list of recently used files. The program displays the main menu, where Neil is shown a series of tools for statistical analysis. One tab displays tools for graphing, and another allows the user to perform single statistical operations. Neil wishes to calculate the mean value of merit across all runs in the data set. He goes to the tab for statistical analysis and selects the Mean operation and the Merit column. He then selects all runs in the data set to perform the analysis on. He presses a button to calculate the value and receives a dialog with the result. Now he wishes to create a line plot graph to compare the median merit value of two different runs across iterations. He clicks on the tab for graphing, selects “line plot” as the graph type, “iterations” as the x-axis, “merit” as the y-axis, and the two runs to be used on the graph. Then he chooses to compare the mean merit values on all runs, by selecting a “scatterplot”, “merit” with the “median” operation as the y-axis, and run number as the x-axis.

Exporting Generated Graphs as Images

Neil is asked to be one of the featured researchers during the HESTEC event at The University of Texas - Pan American, presenting a poster on the evolutionary research he has done with the Avida platform. Neil is aware that the graphs generated by the AvidaLab Analysis Toolkit would be an exceptional visual aide to catch the attention of students and faculty that are not familiar with his research field. To get the graphs on his presentation, Neil will open his project on AvidaLab, select the option to create a graph, and create a scatter plot graph on some of the previously imported data. After creating the graph, toolkit displays a menu with a series of image export options. Of the ones listed, he selects the PNG image to export. The system converts the displayed graph into PNG format and presents a dialog to save the image to the computer. Neil names the file "HESTEC_POSTER_1.png" and selects "Save" to finish the operation. The system writes the image to the file system, displays a dialog to let the user know that the export was successful, and returns the user to the screen where the graphs and statistical values are displayed from the project. Neil exits the software toolkit and proceeds to insert the image into the presentation.

3.6.2 USE-CASE MODEL

Use Case name	**Create Project**
Participating Actors	AvidaLab User
Preconditions	1. The user is at the Main screen after starting the software. 2. The user has previously compressed Avida data files that will be the subject of analysis.
Flow of Events	1. The user selects "Create Project" on the main AvidaLab screen. 2. The system presents a dialog window allowing the user to enter a name for the new project and browse for compressed project files from Avida. 3. The user selects "Next" to confirm request. 4. The system creates a directory on the file system with the name of the project assigned by the user. 5. The software decompresses the Avida data files into the directory previously created. 6. The user is returned to the main screen with an updated project list containing the newly created project.
Exit Condition(s)	User is shown an updated list of projects.
Exceptions	1. Avida data files were not compressed correctly. 2. Compressed file does not contain the proper data files' format.

Use Case name	**Generate Statistics**
Participating Actors	AvidaLab User
Preconditions	1. The user is at the main screen after starting the software. 2. The user selects a project from the available project list window.
Flow of Events	1. The user clicks the "Open" button after selecting the desired project.

	2. The software reads through all the files located in the project folder. 3. The software toolkit presents a dialog window allowing the user to select one or multiple data files stored in the project folder. 4. The user clicks the “Next” button after selecting the desired file(s). 5. The system reads through the selected files and generates a list of columns. 6. The software presents a form with a list of columns available for the user to select for pre-processing, statistical analysis, and visualization features. 7. (Optional) The user selects pre-processing features to be performed in one or more columns (i.e., Logarithmic function). 8. The user selects the desired column(s), and then selects the statistical analysis operations such as mean, median, or standard deviation that will be performed. 9. The user clicks the “Next” button and the software presents a confirmation screen summarizing the selected preferences of the user. 10. The user confirms by clicking the “Next” button. 11. The system performs column initial data validation then any selected pre-processing on specific columns. 12. The software generates the selected statistical values on the columns. 13. AvidaLab presents a Results screen showing the calculated statistical values.
Exit Condition(s)	User is successfully shown the results of the selected analysis features and operations.
Exceptions	1. Project folder was empty (No files available for selection). 2. No column-formatted data was found in the data files. 3. Columns contained invalid data types/format/values. 4. No statistical analysis tools were selected by the user. 5. User cancels at confirmation screen.

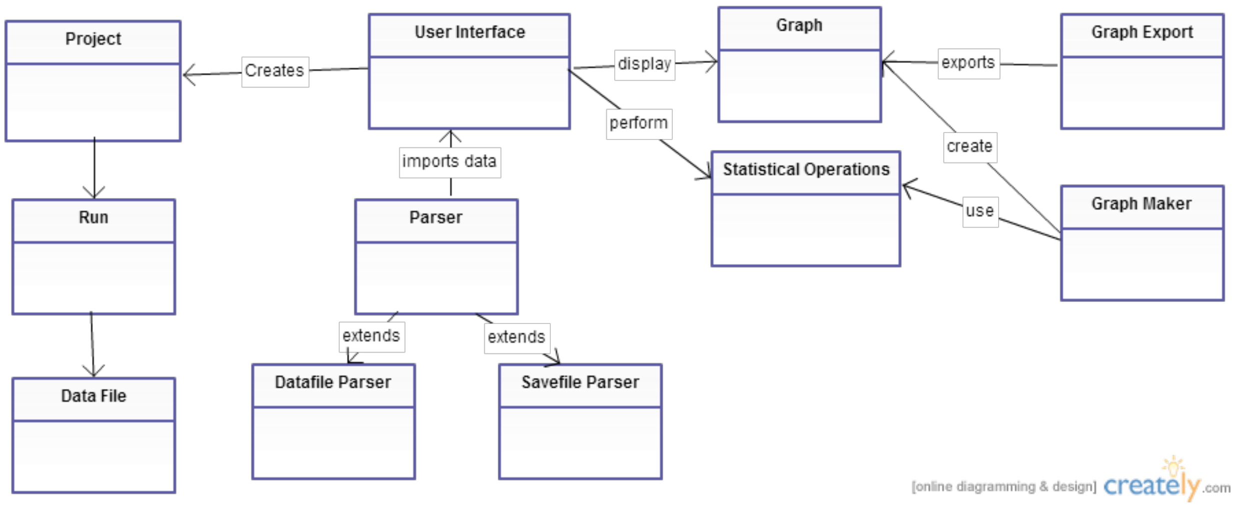
Use Case name	**Generate Visualizations**
Participating Actors	AvidaLab User
Preconditions	1. The user is at the main screen after starting the software. 2. The user selects a project from the available project list window.
Flow of Events	1. The user clicks the “Open” button after selecting the desired project. 2. The software reads through all the files located in the project folder. 3. The software toolkit presents a dialog window allowing the user to select one or multiple data files stored in the project folder. 4. The user clicks the “Next” button after selecting the desired file(s). 5. The system reads through the selected files and generates a list of columns. 6. The software presents a form with a list of columns available for the user to select for pre-processing, statistical analysis, and visualization features. 7. (Optional) The user selects pre-processing features to be performed in one or more columns (i.e., Logarithmic function). 8. The user selects the desired column(s), and then selects the visualization features that will be available to be shown such as linear plot, histogram, box and whiskers plot, and scatter plot. 9. The user specifies the correct columns and axis for each plot type selected. 10. The user clicks the “Next” button and the software presents a confirmation screen summarizing the selected preferences of the user. 11. The user confirms by clicking the “Next” button.

	12. The system performs column initial data validation then any selected pre-processing on specific columns. 13. The software generates the selected visualizations on the columns. 14. AvidaLab presents a Results screen showing the plotted graphs.
Exit Condition(s)	The user is successfully shown the results of the selected visualization features and operations.
Exceptions	1. Project folder was empty (No files available for selection). 2. No column-formatted data found in data files. 3. Columns contained invalid data types/format/values. 4. No graph types were selected by the user. 5. Invalid columns were selected for a certain axis. 6. User cancels at confirmation screen.

Use Case name	**Exporting Image Files**
Participating Actors	AvidaLab User
Preconditions	1. The user is at the 'Results Summary' screen after performing analysis on data files within an existing project. 2. The user has selected one or more visualization features to be generated.
Flow of Events	1. The user selects one of the graphs plotted and shown on the screen. 2. The software provides a menu of image formats to export the graph including PDF, JPG, EPS, and PNG. 3. The user selects only one of the options. 4. The software toolkit processes and generates the graph as an image in the selected format and provides the user with a "Save As" dialog window. 5. The user navigates and selects a folder location and a name to save the image. 6. The user confirms the action and the software saves the image file in the format specified, with the assigned name, and in the selected folder. 7. The software returns the user to the 'Results Summary' screen.
Exit Condition(s)	The graph selected by the user is exported as an image in the specified format to the file system.
Exceptions	1. Invalid path or name of file is chosen. 2. User cancels at "Save As" dialog. 3. Access is denied to save file on the desired folder. 4. Duplicate name of file already exists in specified folder.

3.6.3 OBJECT MODEL

3.6.3.1 Class diagram

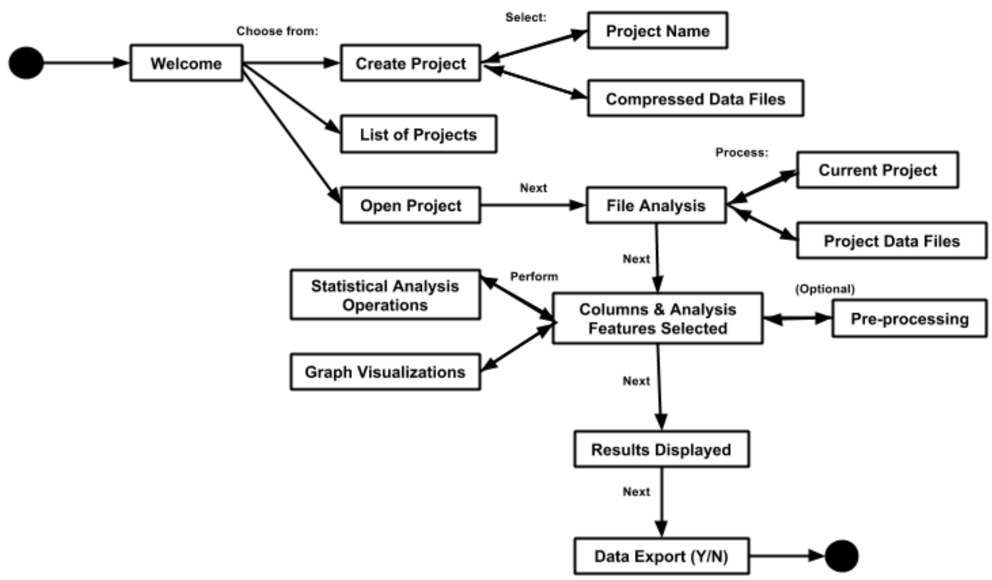


3.6.3.2 Data Dictionary

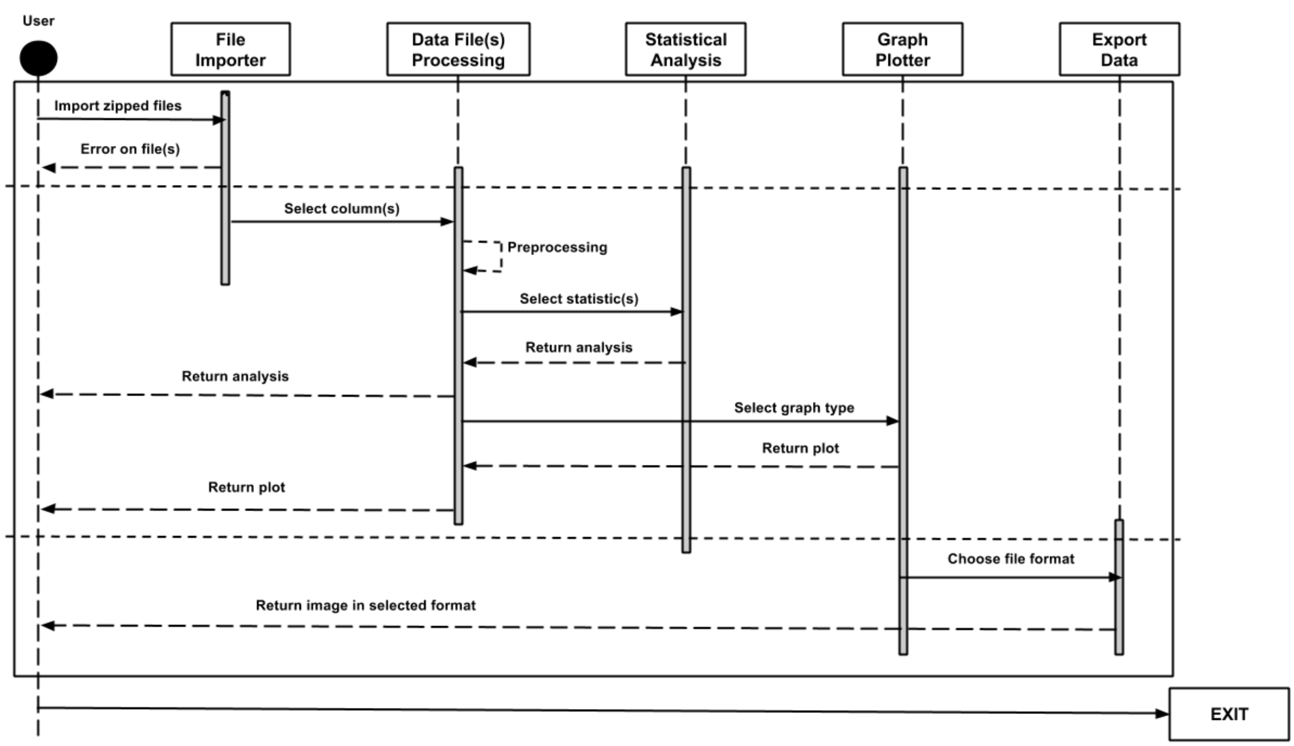
- The logical organization of experiment data will be as follows:
  - A **Project** will contain AvidaLab saved and imported data for that particular data set, divided by runs.
  - A **Run** will consist of all the data files for a single execution of the experiment.
  - The **Data File** class will internally organize all the fields read from a file in a "dictionary" type array with the name of the field and its value.
- The **User Interface** will provide a way for the user to interact with the data and read/write files.
- The user will perform statistical analysis through:
  - A **Graph Maker**, which creates **Graphs** from experiment data.
  - The **Statistical Operations** class which performs a set of operations on this data.
- File input and output will be done through:
  - **Parsers** extended to a **Datafile Parser** for importing datasets, and a **Savefile Parser** for loading project save files.
  - The program will also use a tool for **Graph Export**, in order to convert Graphs to different image file formats.

3.6.4 DYNAMIC MODEL

State Diagram

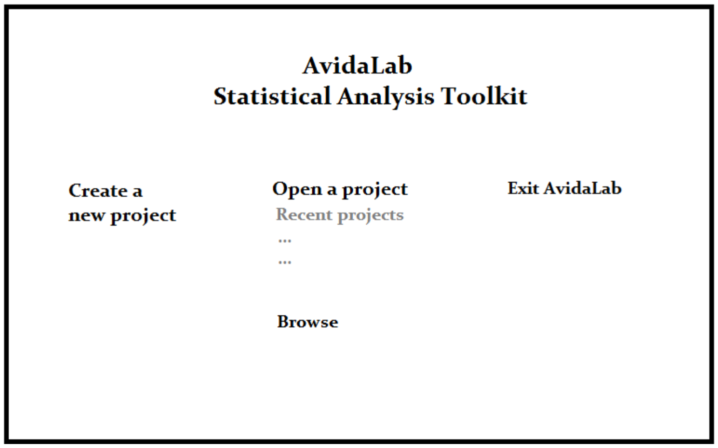


Sequence Diagram

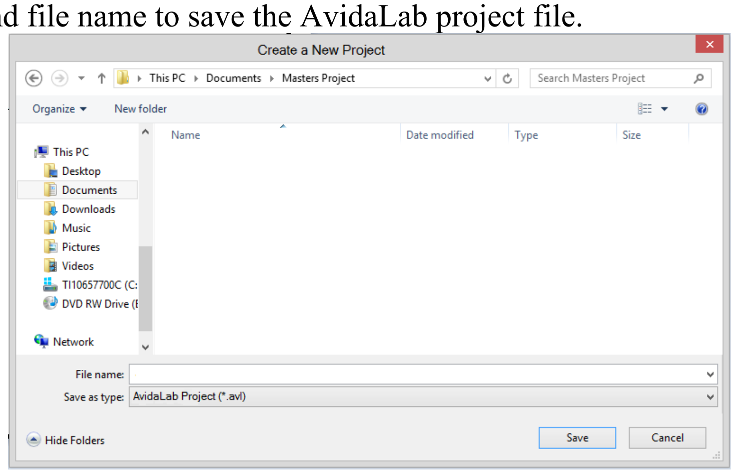


3.6.5 USER INTERFACE - NAVIGATIONAL PATHS AND SCREEN MOCK-UPS

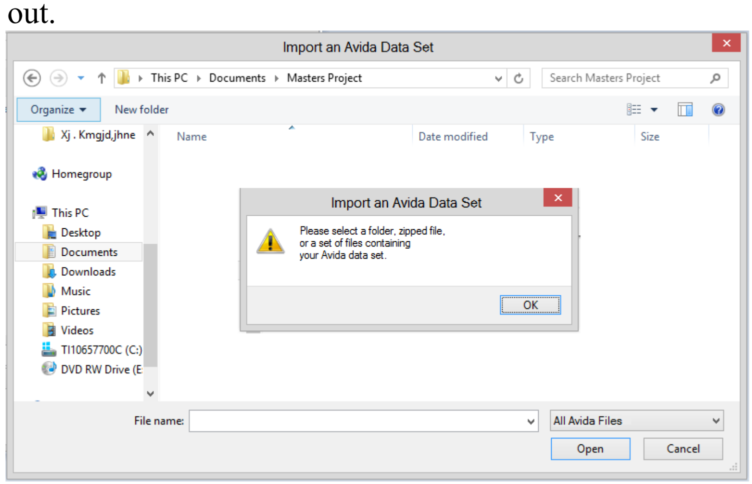
When the user runs AvidaLab, the program displays the Welcome Screen. This screen displays a button-driven menu that allows the user to create a new project, select an older project from a list of recent ones or by browsing the file system to find a project file, or exit the program.



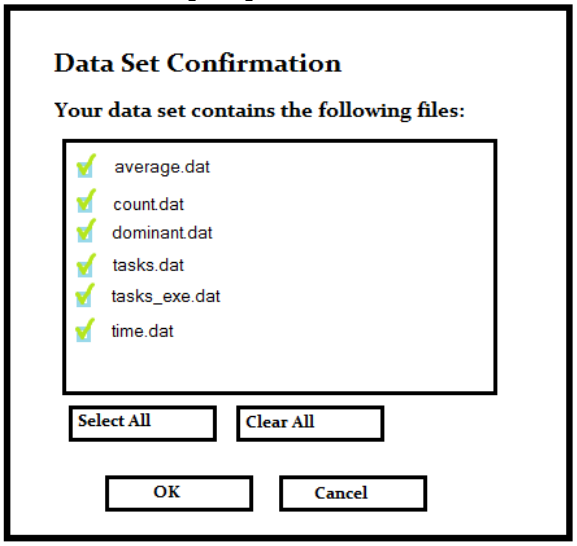
When the user selects "Create a new project", a save file dialog pops out prompting the user to select a directory and file name to save the AvidaLab project file.



After saving the file, the program prompts the user to select an Avida data set to import. A similar dialog pops out.

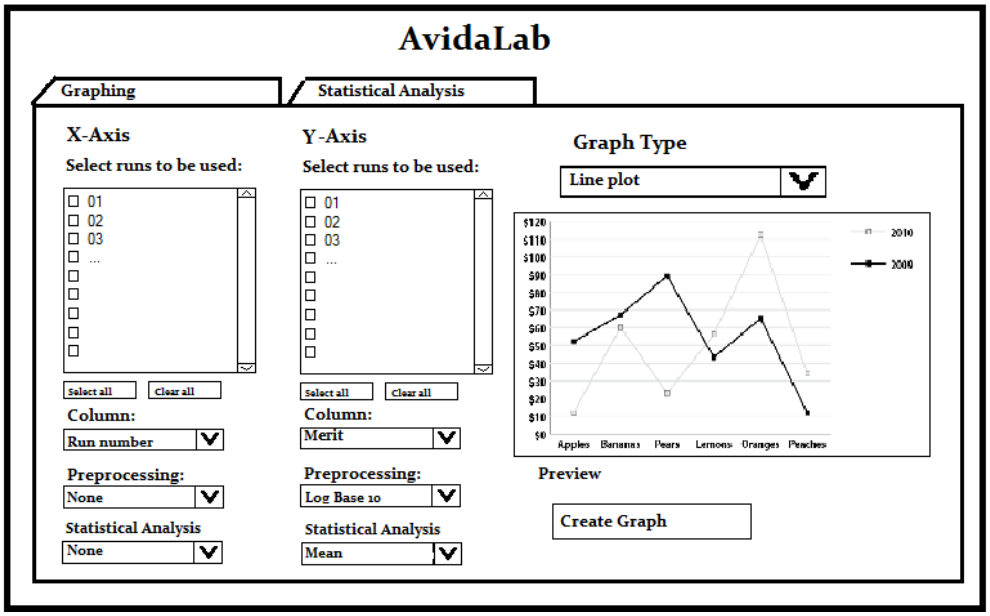


Having selected a proper set of files, a confirmation dialog is displayed. This dialog allows the user to review and select which files are going to be used.

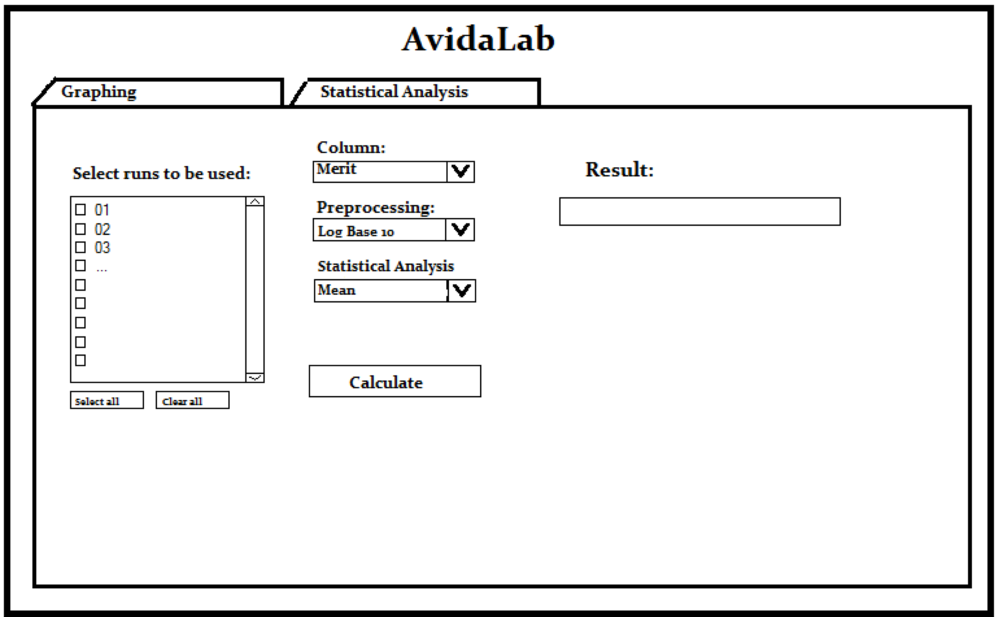


After selecting a data set to analyze, the user is taken to the main screen. This screen contains two tabs: one used for graphing, and other used for statistical analysis.

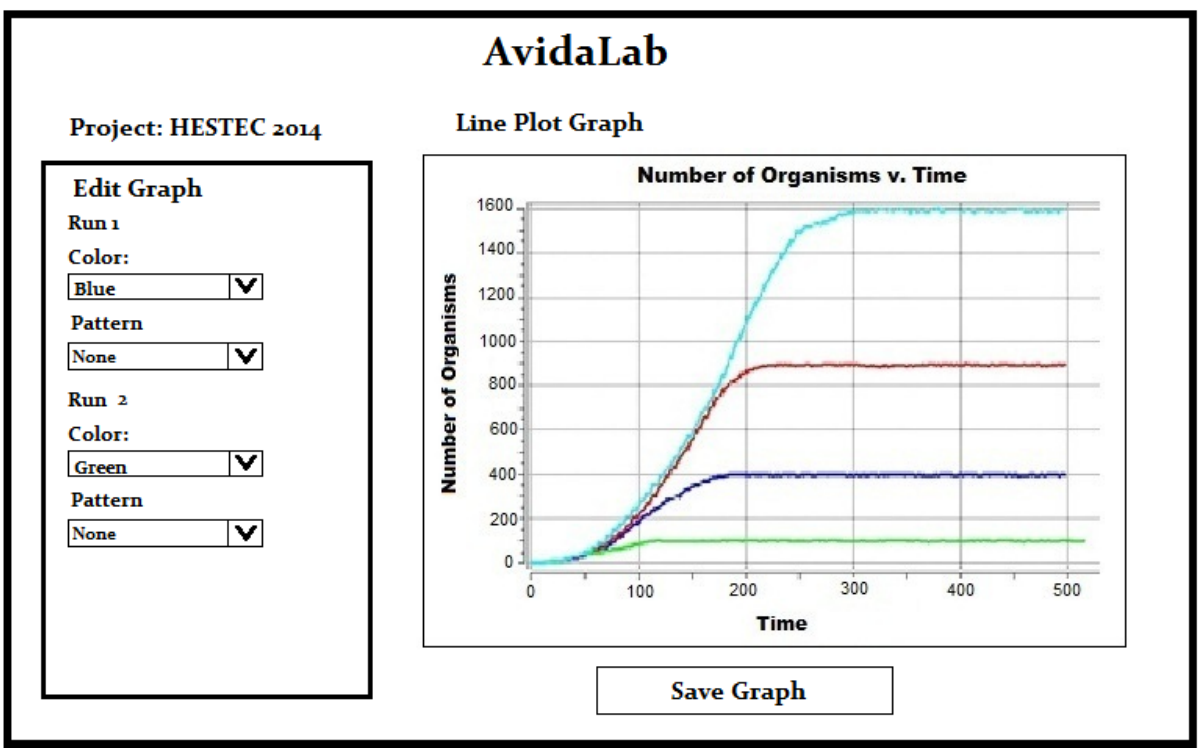
The Graphing tab displays a number of options used for the graph design, such as choosing columns for a certain access, which runs of the experiment will be part of the graph, and which operations to perform on these values. The design of this tab may change according to the type of graph being built, which can be selected with a dropdown menu. Note that on the following screen, the graphing tab is selected.



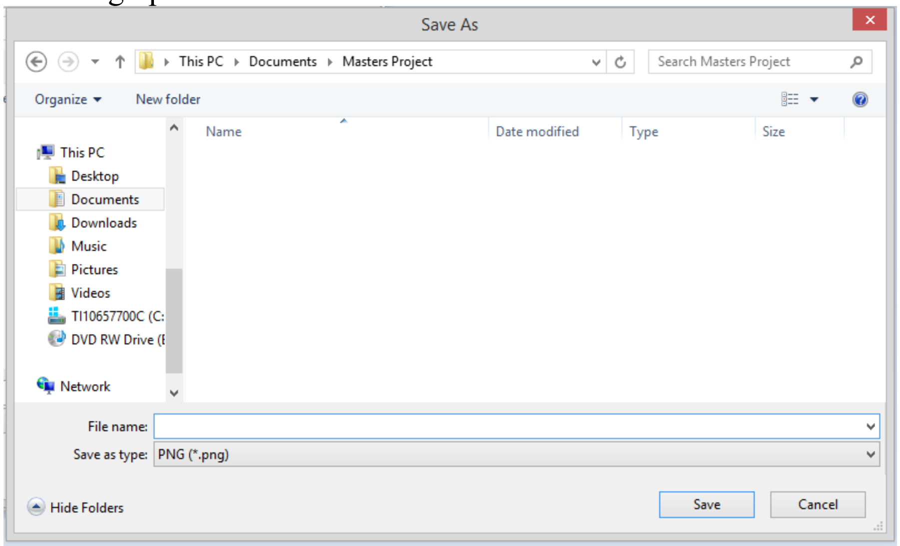
The Statistical Analysis tab allows the user to select a number of statistical preprocessing and functions to be performed on a column of data, from a user selected number of runs. The result is displayed on the same screen.



If the user chooses to create a graph on the Graphing tab, the program checks for the validity of the selected fields, and generates the graph. The user is taken to the following screens, which displays the graph and has the option to make a couple of aesthetic adjustments.



The previous screen gives the user the option to save the graph to a file. If this button is clicked, the program will display a save dialog like the previous ones, where the user can select which image file to save the graph as.



In addition to these screens, AvidaLab will display error messages whenever the user enters erroneous information, and will take the user back to the previous screen. The system will also provide navigation tools, facilitating the user to go back and forth between screens.

4. Glossary