**Software Development for Neurophysiological and Behavioral Studies**

BE 177 – Group A Final Proposal

Chen, Edwin

Nelson, Darwin

Nonaka, Anna

Yindeeyoungyeon, Virata

**SPECIFIC AIMS**

As the capabilities for data acquisition have exponentially increased in volume and resolution, cognitive and neurobiological sciences have become increasingly dependent on technology for data analysis. These analyses are critical in determining relationships between tangible, neurobiological mechanisms and cognitive processes. For example, eye position can be tracked, recorded, and analyzed over a series of visual cues and rewards in order to understand how cognition is related to sensory-motor pathways. These studies further extend to understand how these relationships are compromised in the case of disease. The problem in this field of research is the inability to analyze the data collected. Dex, a program used to graphically display the collected data, is written in C++ and is currently the only program of this nature available. Dex frequently crashes or fails to output results when issued certain commands and the user interface can be daunting for a researcher unfamiliar with the program. Many researchers have had to abandon their projects, avoid this field, or try to adjust to the compromised program. Additionally, most researchers in this field do not have the programming background required to amend the Dex code or generate a new program.

This project seeks to provide a widely accessible program written in Python for analyzing eye position and experimental event data that can be used by laboratories with researchers of varied resources and data formats which entails the following: (i) Create a comprehensive program with all the functions and data formats that should be operational in Dex in order to address the analysis needs of the researchers in this field. (ii) Engineer a user-compatible interface that requires little to no instruction in order to operate the program functions. (iii) Provide the program for significantly lower costs than Matlab or other program licenses.

This project will face the internal challenges of collective programming, avoid the previous errors unresolved in the Dex C++ code, and anticipate and satisfy the needs of a variety of research projects. We will use version control software to share and track changes to the program code, increasing the readability of others’ work and thus facilitating the integration of our code. In order to mitigate possible failures in the program code, we will first devise a very detailed schematic of the program architecture. This also orients all group members to write with the same intentions and integration in mind. Using these methods, we propose the following:

**AIM 1: Develop a portable program capable of meeting the analysis needs of the field of cognitive neuroscience.** Although it is not completely functional, Dex provides a comprehensive guideline for the functions needed by the target research community. This project will create a program with all the features that are ideally operational in the current Dex program. These include creating and marking signals in the data strands, building data rasters, viewing the data through different parameters, summarizing the statistical analyses, and outputting the results of these functions to portable files. The successful product will not have any of the errors associated with Dex and functionality will be verified by unit tests (providing known outputs) throughout the development process. (Experimental data is also available, although some of the functions in Dex are not operational so those outputs are unknown). We have also selected to write in Python since this is an open source, simpler, and highly portable language while still retaining the necessary programming power for this project. This simpler language may have the additional advantage of researchers being able to update the program code to their more individual needs independently.

**AIM 2: Develop the capability to accept various file formats in order to increase the program expediency.** Data files can vary in both file format and acquisition resolution which can affect a program’s ability to accept and analyze data. We will implement internal conversion capabilities of files obtained by matlab programs and Rex (most widely used for data acquisition in this field) based on existing libraries in order to allow researchers the freedom to obtain data in any method that is feasible for their resources.

**AIM 3:** **Design an interface that is user-friendly, requiring minimal training.** We will test the interface outlay with prospective users and survey these users for the operational ease of the program. The program will be formatted and tested to be able to perform equivalent functions at a faster rate than Dex. These tests will attempt to prove with statistical significance that using our program is faster and results in less unnecessary navigation. Inexperienced users will be asked to complete a set of tasks in Dex. Then, 1) time to complete each task and 2) number of unnecessary navigation steps will be recorded. The successful program interface will be operable by a completely inexperienced user with little or no training in a relatively short amount of time when compared to the Dex system.

**I. SIGNIFICANCE**

***A. Neurological and Mental Disorders*.** Cognitive neuroscience is a field that encompasses the study of neurological and mental disorders. Research in this field provides useful neurological and psychological information regarding diseases such as addiction, schizophrenia, and Parkinson’s disease. Studies focus on the relationship between eye movement, neuronal pathways, and cognition. The experiments are generally designed to stimulate the subjects electrically, pharmacologically, or visually [1][2][3][4]. An event-specific code will be transcribed for each event throughout the experiment (i.e. beginning of a trial, a reward, ending of a trial), so that the researcher can observe the correlation of these events with the subject’s eye movement in real time. Eye position data are recorded by a magnetic induction technique. The event codes and positional data are saved as a waveform, a binary signal format, for later analysis [5][6][7]. These studies allow researchers to further see into the neurological and behavioral aspects that make up the human mind, leading to greater understanding of the human body and the neurological and mental diseases that may attack our bodies. Mental disorders affect 26.2% of American adults and neurological disorders affect 33.3% of Americans at some point in their lifetime, making it clear that we are dealing with no small amount of the population [8]. Mental disorders are also the leading cause of disability in the US and Canada [9]. Relative to the prevalence of neurological and mental disorders, disease understanding is very limited. In cognitive neuroscience, this limitation is imposed by a bottleneck in the research process at data analysis. Relieving the bottleneck by providing a functional, easy-to-use analysis program, as our project aims to do, will result in field-wide standardization and consistency in research analysis promoting field expansion in terms of researchers and knowledge. This could have transformative effects on the diagnosis and treatment of the neurological and mental disorders causing significant problems like disability.

***B. Limitations and Problems With the Current Analysis Options.*** The major bottleneck or rate limiting step in this field is that the scientists performing this type of research are not able to analyze their data due to little or no experience in programing. Data analysis programming limits the speed and scope of their investigations. This field needs a reliable, universal, and simple program to eliminate the need for programming competency or having to settle for a compromised program such as Dex. Many experts in this field who have been trained by the NIH frequently use this program to analyze, display, measure, and calculate their experimental results. However, Dex has several problems that include crashing, loss of work in progress, and inability to perform a listed function. For example, errors in data allocation coding do not allow researchers to process batch files under certain functions. When researchers attempt to compile their results, the program crashes and fails to deliver any output. Even this error is not consistent and sometimes a researcher is lucky enough to load his data. In this case, many other errors can still occur when operating Dex. For example, when a researcher clicks to zoom in on a graph, the entire graph may suddenly disappear and the researcher has to repeat the process of loading the data and building the graph. Furthermore, Dex does not have flexibility for the input files and only accepts A-files and E-files generated by Rex [10]. The lack of analysis capability in this niche of research requires a novel, well-engineered, and commercially available GUI-based program. As mentioned previously, these hurdles have caused researchers to abandon their work and prevented researchers from entering this field altogether who could otherwise have be capable of transformative work if given the proper tools.

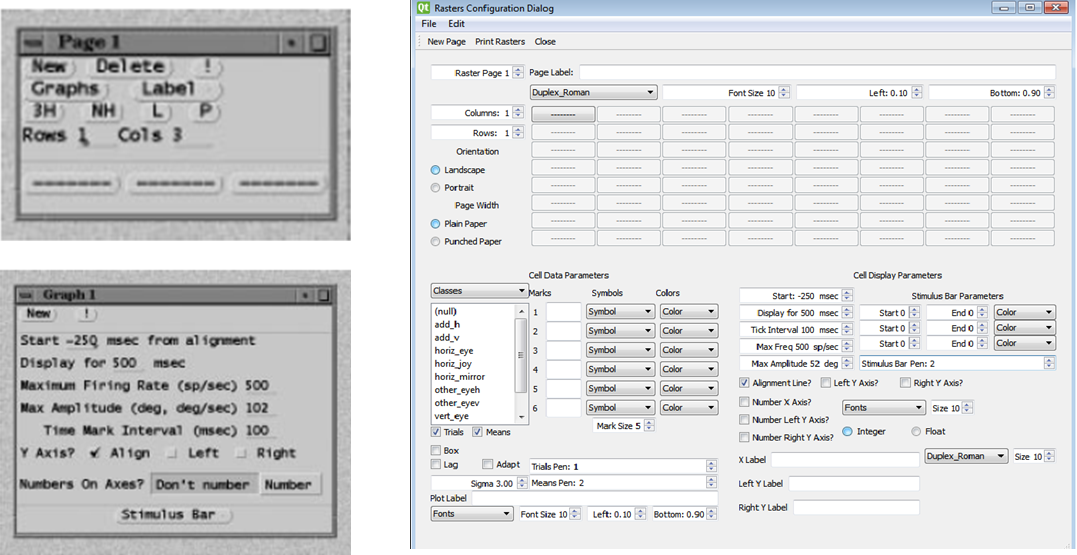
***C. Making the User Interface More Intuitive and Aiming for Standardization.*** We aim to engineer a reliable, user-friendly graphical user interface (GUI) based application using the Python programming language to simplify data analysis for the positional, neuronal, and experimental event data obtained in behavioral and cognitive neuroscience experiments. A GUI is the platform a researcher uses to access the program functions and is like a map that allows the user to view his options. This usually is structured as a series of pop-up windows and menus. A GUI-based application enables the users to manage and communicate with the program in a simplified and intuitive manner such that the scientists who are not familiar with programming could still efficiently examine their data in the form of explicit, graphical icons within the program. They would be able to point, click, and look at their data without going through any complicated coding processes. The Dex GUI is very complicated and even the manual fails to guide researchers adequately. As seen in Figure 1 on the right at the top of the next page, sometimes the manual instructions to not correlate to the actual Dex menu windows [11]. The data analysis program we will produce will allow researchers to analyze the positional information recorded in eye movement studies and display their numerical results in more coherent visual mediums, such as graphs. Although there are several prominent researches in the area of cognitive neuroscience, there is no universal, reliable platform for the integration of data acquisition and analysis. Researchers either suffer through Dex or produce their own algorithms. Consequently, the same experiments may be analyzed using different platforms, producing slightly different results for each researcher or laboratory. A standardized universal platform would allow researchers to more equivalently compare results and streamline the research processes in this field. Researchers would also be able to more easily and accurately compute their results. This GUI-based application will facilitate the merging of scientific data and creativity in this field which will accelerate communication and innovation. Creating a reliable, GUI-based interface program would not only accelerate cognitive neuroscience research across the country, but also broaden this research community by opening up research possibilities in this field to scientists who were not previously capable of analyzing their data or avoided this type of research because of the data analysis complications.

Figure 1. To the left shows the GUI as described by the Dex manual. These two windows are supposed to represent the window in the actual Dex GUI, shown on the right, and instruct the user in building a data raster, one of the core functions of Dex. There is clearly a strong disagreement between the manual instructions and how the program is actually operated [11].

**II. INNOVATION**

**A. Background.**

*Current processes of data acquisition and analysis in cognitive neuroscience.* In recent years, research in the field of cognitive neuroscience has led to increased understanding of normal neurological and physiological functions as well as understanding of these processes in diseased cases, illuminating possibilities for diagnoses and treatment. Researchers in this field are examining the neuroscience behind our cognition and decision-making behaviors through eye movement tracking. They must first acquire behavioral information from animal and human subjects in the form of eye position and neuron firing data, and then correlate these with experimental events[1],[3]. While experimental platforms may vary, the data acquisition software is highly standardized. This program, Rex, generates a binary data file containing two lines of time-stamped information – one for eye movement and neuron firing and one for experimental events. For data analysis of this binary file, the most widely used program is Dex, which is a GUI-based program exclusively created by the National Institute of Health (NIH) and is not commercially available to the public [11]. Ideally, Dex allows researchers to view their data in different formats such as time oriented graphs or density plots called data rasters. The goal of these graphical representations is for researchers to determine data trends in order to lead to greater understanding of neurological and physiological processes.

*a. Dex, and C++.* When working with Dex, one problem is that the program is written in C++. Reworking another person’s code is nearly impossible in many cases and C++ is one of the more difficult programming languages available. To compound this issue, most researchers in this field do not have any C++ background. These problems associated with Dex being written in C++ do not allow current researchers to generate the experimental analysis they need or to be able to fix the problems impeding their analysis [11].

*b. Financial hurdles in acquiring current systems.* Many researchers currently use Dex or MATLAB for their data analysis needs and these programs may require licenses [12]. While many academic labs have a campus associate license or funds for such programs, not all researchers in this field are academically associated and/or have limited resources. These licenses may also be limiting as to the extent to which researchers can use the program. While these may not prevent researchers from making space in their budgets for acquiring the necessary programs, there is no disagreement among those purchasing such licenses that such resources would be more useful if applied to other aspects of improving their research.

*c. Cross-Platform Portability and Field-Wide Standardization*. Dex is only available for Linux and Windows and is incompatible with Mac operating systems (OS) [11]. While Mac OS may not have been very common when Dex was originally written (over 20 years ago), nowadays, almost every single program or application is available for Mac OS. This makes Dex inconvenient for Mac users who must run the computer in “bootcamp” to use Dex. Dex also only loads data that is generated by Rex, and, while most researchers do use Rex, some researchers do not use Rex and instead obtain and analyze their data through other platforms [11]. The limitations in Dex portability not only limit the researchers that use this program to Rex users, but may also be the cause behind the lack of data analysis standardization in the field of cognitive neuroscience. Research in this field needs a universal program for this type of data analysis to increase communicability and reliability of results.

***B. Areas of Innovation***

*Choosing a programming language and coding approach.* This project will engineer an upgraded program for researchers in eye movement and neuron firing studies of cognitive neuroscience to accurately analyze their data and generate results without internal program errors or complications. Although the Dex system is available, a truly portable, fully functional system for this type of analysis is not accessible to the public. This project will take an innovative approach to addressing the analysis needs of these researchers while considering their programming capabilities and the operational ease of the program in several ways. We will use python, a novel language in this field of research, and a modular approach. This will make the program accessible, effectively eliminate analysis costs to the researchers, and offer programming flexibility not found in Dex.

*a. Python.* Python has exemplary GUI toolkits, which allow users to create an appealing interface with high-level functionality. This provides us with the widest range of major GUI libraries, such as Qt, Traits, and Wx, that we can use to design the GUI [13]. Python is also an open source language, so our program will be free to public as opposed to requiring an expensive license [14]. On the other hand, using another language, like MatLab, and requiring an expensive license will impede researchers with limited resources from accessing the program and standardizing research analysis in this field. Using Python allows researchers to freely view, modify, and adapt the source codes to their own research. The relative ease of Python to more complicated platforms like C++ also makes it easier for researchers with a weaker programming background to adapt the program, which is nearly impossible to accomplish with the Dex system and would waste valuable research time. Python also provides extensive file conversion libraries and scientific programing dictionaries to organize data and results to help streamline the program architecture and allow researchers versatility in both the input and output of their file types [13],[14],[15],[16]. Python is also a high-level programming language, meaning that this language is more removed from the elements of a computer system versus a more rudimentary, low-level language [15]. Being a high-level language, Python code completing the same purposes of the C++ Dex code will be 5 – 10 times shorter, making our program easier to write and read [16]. Additionally, Python does not sacrifice the necessary programming power and speed for this project despite being a higher-level programming language.

b. *Modular Architecture*. Programmers usually have a methodical outlay for their programs and try to document their writing. One architecture method is to write the program functions in modules such that each module completes a distinct purpose of the program. These modules work independently of each other and can be integrated to create a unified program [18]. A modular architecture packages and distinctly divides up different chapters of the program. This accomplishes a highly organized program and makes navigation of the program much easier – something greatly lacking in the Dex source code. Using a modular build also reduces the severity of a critical bug in the program. Modules allow for targeted testing to more easily and quickly identify the error location. In addition, if there is a critical bug in a more exterior module or even a core module before exterior modules are integrated, it is isolated to that module and will not affect the functionality of the rest of the program. Preventing critical failures is a key focus in our build plans since this is the main failure of the Dex program. Modular architecture also allows for relatively easy modification and addition in the future since a user can easily work on a single module they wish to update or can simply integrate a new module they have written. Whereas Dex is stagnant, our program will be able to evolve and expand.

**III. APPROACH AND DESIGNS**

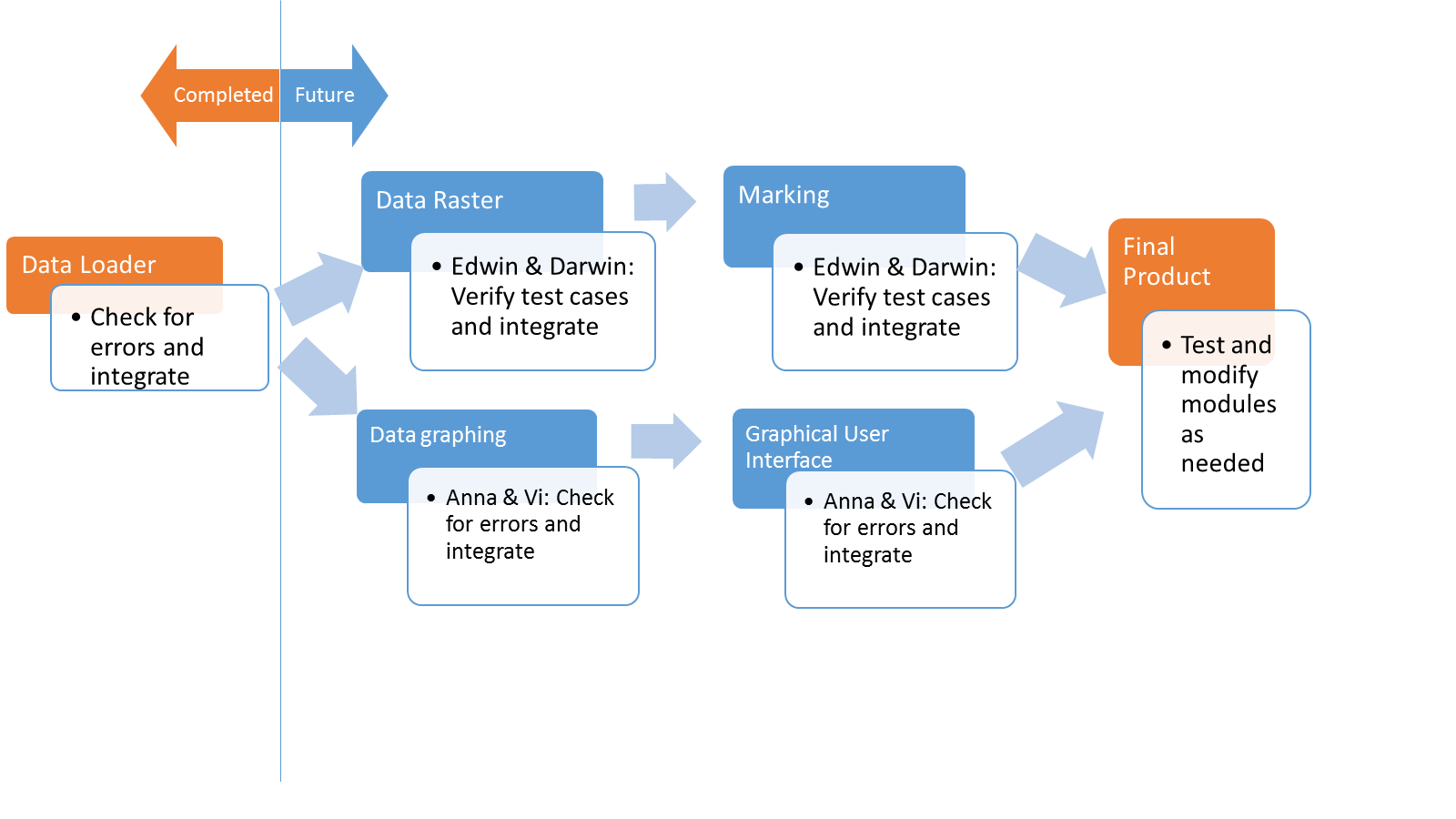
***A. Preparation – Defining the Product and Building the Coding Environment***

*1. Defining a successful, functional product.* In order to create a program that can become widely used in this field of research, we must assess the needs of the clients. We interviewed the UCLA lab that is contracting this project and contacted five other prominent researchers in this field1 in regards to what they desire in the fully functional program and the level of programming and statistical abilities of their researchers. These researchers serve as a representative client for the several campuses across the country performing this type of research and we find more collaborators in the new quarter as well. These researchers specifically desire that a new program have all the operations previously available or ideally available (some functions are faulty) in the Dex software. Our project has also obtained the Dex manual and source code, allowing us access to all the aspects of the Dex program [11]. This program has been described as an accurate model for what researchers are looking for from our project. However, simply observing researchers using this program and hearing testimony from the laboratory directors at UCLA has made it clear that in addition to dysfunctional actions, the program is overly complicated and not intuitive. In order for researchers to accept a new program, it must offer a better version of the old capabilities, be more readily updated, and allow even an inexperienced, undergraduate researcher to perform all functions from data acquisition to producing graphical analyses. This success may also be quantitatively assessed as described below in our Research Design and Methods section.

*2. Programming platform counseling.* Our project has decided to use the Python programming language (as previously stated), Git, and GitHub for our writing process. We chose Python based on the ease, power, and portability of the language as described by our teaching assistant, Keegan Owsley, and Python tutorials . Python is also free to use and will incur no costs for our research project [17]. We have also chosen Git and GitHub for their abilities to record program modifications and provide a server for us to work collectively on the same code [22]. These were also described to us by our teaching assistant. Based on the evidence, this platform should create the easiest writing environment for our project while still being capable of writing the functions we require and will similarly be the best platform for our client for the same reasons. These capabilities are also possible in MATLAB; however, this license will limit our writing to using a remote desktop to SEASNET at UCLA and be more expensive to our clients who do not own the license [12].

*3. Unit testing.* To avoid the coding problems encountered in Dex, our program will be tested for validity and accuracy by a unit-testing method. A unit test uses mock data with a known output to verify the correctness of a unit of functionality and prevents failing from the early stages of programing to the final product. Each unit test is designed to test particular functions of the program and can expose and locate failures at every step of the writing process, allowing the programmer to target the errors.

*c. Git and GitHub – tracking modifications and increasing the readability of the program code*. Git is used to track changes throughout the coding process and GitHub is an online entity that allows programmers to share and collectively work on the same code. This allows our project to keep track of previous versions and view how other members have modified the code. In addition to facilitating our collective efforts in writing this program, Git and GitHub can also allow researchers who will potentially use this program the same readability of the code. This will in turn aid them should they choose to modify the program to meet their needs or create additional functions to facilitate their analytical processes. This feature is not offered by Dex.



***B. Research Design and Methods***

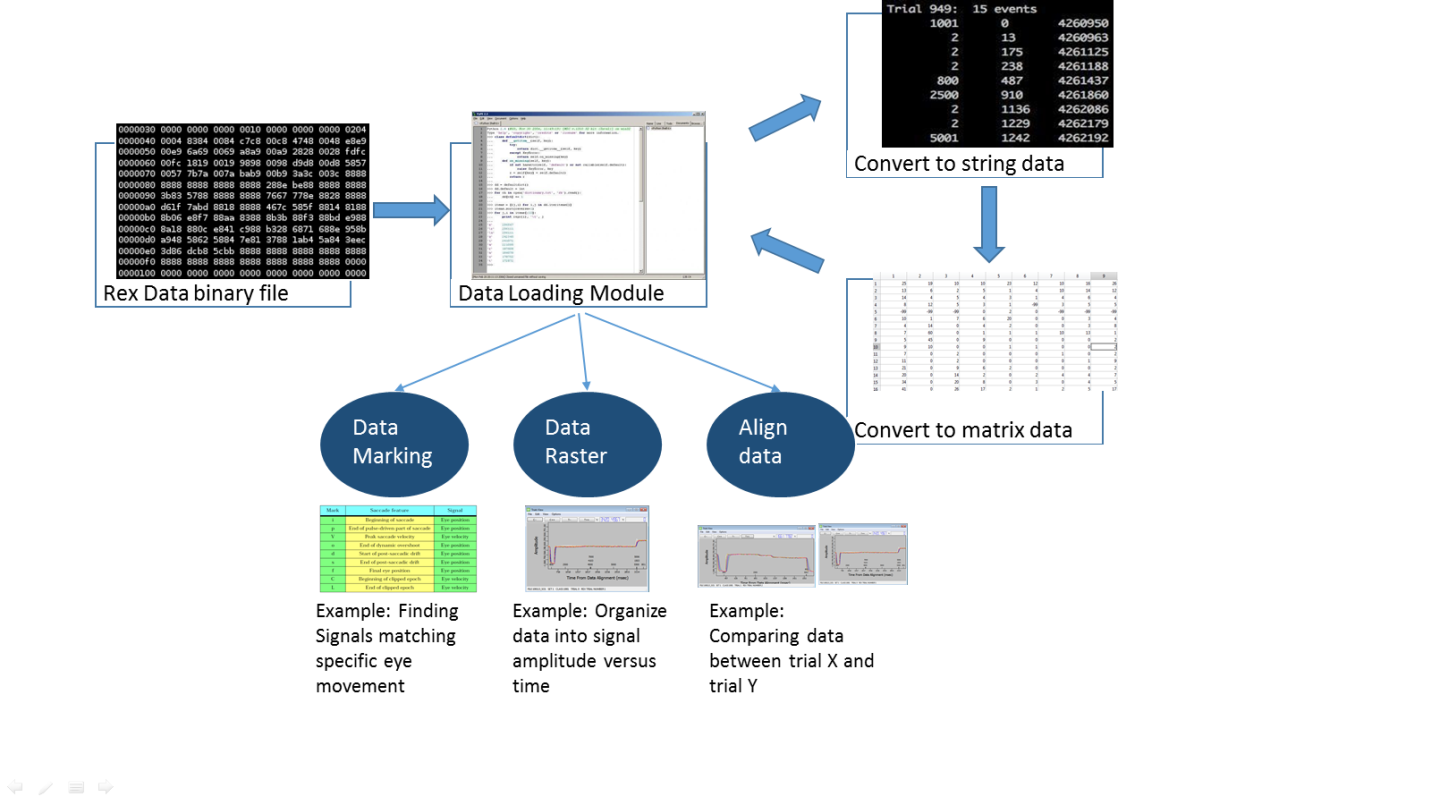
Figure 2, left, describes the overall plan of the development process. Using a modular approach allows our team to divide the program by modules and work on the different program functions in parallel and independent of each other. Next quarter, we plan to write and integrate the blue modules and GUI as illustrated by Figure 2 to produce the final product. We then plan to thoroughly test and modify the program as time permits. These plans are discuss in detail in our specific aims below.

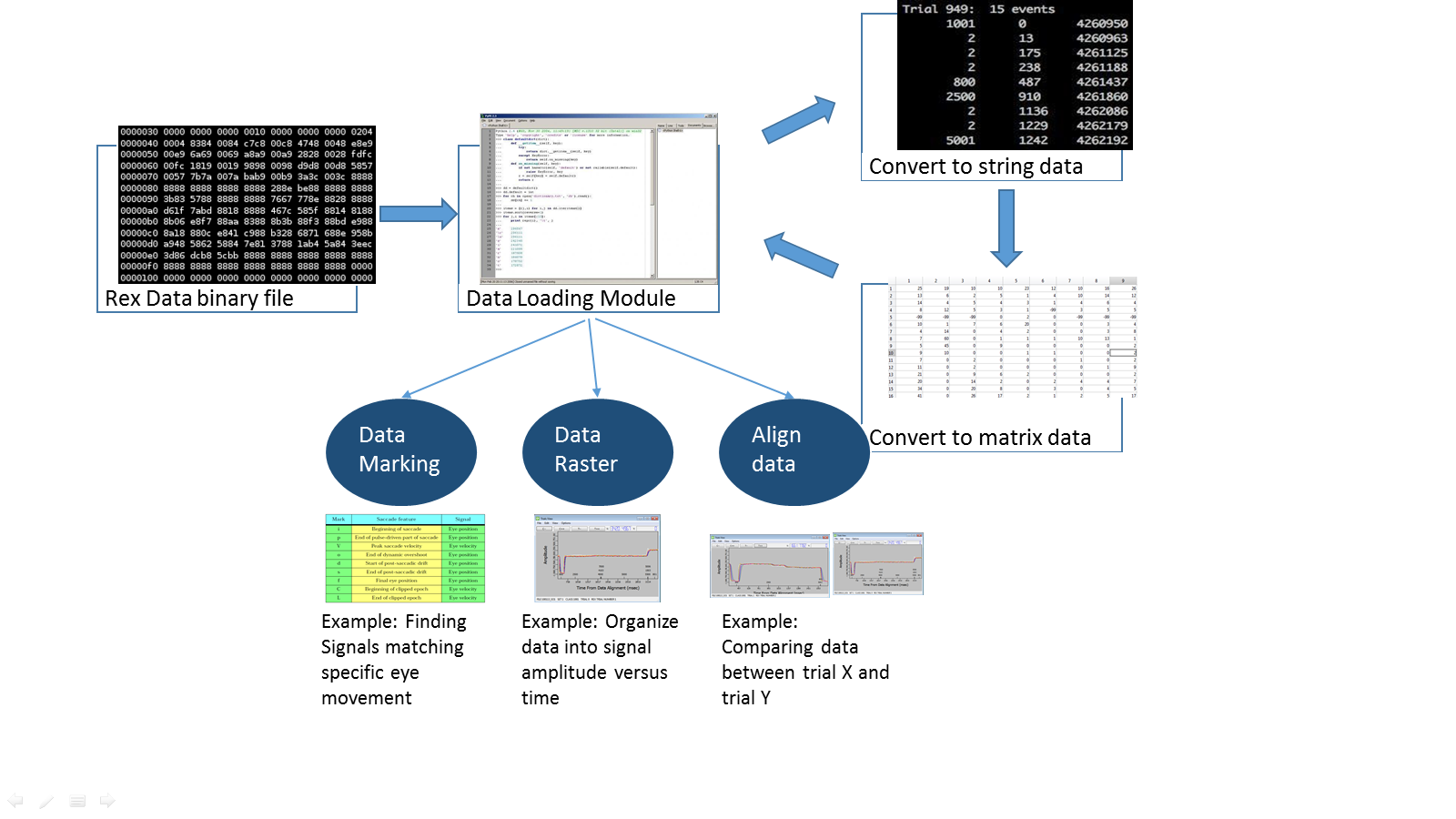
Figure 2. This figure serves to illustrate the work flow of the project and to give developmental context to the aims discussed below.

*Specific Aim 1.* *Provide cognitive neuroscience researchers a cross-platform program with the analysis capabilities necessary to relieve the data analysis bottleneck*. Currently, there is no standardized analysis platform across different labs conducting decision-making neurophysiological and behavioral research. Our aim is to unify the analysis experience in a comprehensive package such that all the analysis needs attempted in Dex are met, the program is compatible with all major OS (i.e. Windows, Mac, Linux), and comparison of standardized methods and results in the field can be realized.

*1a. Meet or exceed the analytical power of the predecessor Dex in a flexible offline package.* Functions will be modeled after the aforementioned Dex analysis software through a ground-up coding approach in Python. Writing in Python will provide offline compatibility with all major OS and allow us to utilize libraries from other languages [16]. For example, we are able to make use of functions that have already been created, such as the trdd function (discussed later) to save on programming time and make our writing more succinct and thus readable. The standard package we hope to provide will include the Dex functions we have previously discussed in this paper: data-point marking, building data rasters, parameter-filtered data viewing, statistical analyses summarization, and portable dump-to-file ability [11]. We will continually consult the writer of Dex as well as end-users with regards to functionality to increase product usefulness as well. These consultations will give insight to the researchers’ preferences for how they can access the aforementioned functions, modifications to existing functions, or additional function modules should time permit expansion upon the baseline Dex capabilities.

*1b. Systematically prevent and mitigate risk of software errors by unit testing and version control software*. A primary reason we endeavor to write software de novo is to replace the error-prone, limitedly-functional Dex software with a bug-free successor. We will introduce progressive unit tests of the software, which will help ensure functionality at each advancement in the code to identify errors early on and avoid costly program re-design. Programmers will generate unit tests as frequently as they feel necessary to access their code and the entire team will extensively test each module upon completion before integrating the module into the core program module (Data Loading Module). This testing and integration intervals are illustrated by the work flow in Figure 2. Additionally, as stated, third-party version-control software will be utilized to also reduce risk and allow software version rollback if needed. GitHub will create multiple versions of the code by each user and maintain a core code that each user can upload and pull from [22]. These multiple versions also track changes made to the code, essentially offering the programmer a “CTRL-Z” to return to a previous version of their code, perhaps before creating a critical bug.

*Specific Aim 2. Use a modular architecture to clearly organize the program, allowing modification and expansion flexibility of the program in the future*. As discussed in Areas of Innovation, a modular build offers many advantages to us as the program writers and to users who may wish to modify the program. A diagram illustrating the methods of interaction between our modules is shown to the right in Figure 3.

(a)a)

*2a. Write individually functioning modules and prepare them for integration*. Before becoming part of the general program architecture, each module will be confirmed to have functional output without error. For the data loading module, this would be the capability to successfully transform the Rex generated binary file into matricies while still retaining the time-stamped information of experimental events, eye position, and neuron activity. For the three more external modules (data marking, data raster, and data alignment), error-free functionality will be determined by the ability to accept the matricies passed from the data loading module and generate an output based on that module’s function that the user can then manipulate and save for further examination.

Figure 3. (a) The binary file generated by Rex is loaded into the first module, data loading. (b) the data loading module outputs various matrices containing the time stamped data to the three modules in blue – data marking, data raster, and align data modules. (c) Sample outputs are shown below.

(c)

(b)

*2b. Ensuring future flexibility by the nature of a modular build and annotating the code*. As most programmers attempt to do, our code will be well documented and major functional units within the modules will be discussed at length within the source code for any user that may wish to alter the program. The two barriers to sharing data analysis platforms in this field are code readabillity and MatLab license expenses and restrictions. By using Python and this well structured modular architecture, we will eliminate these barriers to promote sharing of this data analysis platform and possible user-developed additions / modifications. This sharing will result in standardization which is one of the main goals of this project. Our fluent architecture and thorough annotation will provide a strong foundation for data analysis. Having a strong foundation allows the technology to move forward more readily. Collaborators will also be asked to indicate any areas of confusion in the souce code so we can be assured of our code readability.

*Specific Aim 3. Update the Dex GUI pathways and windows to create a more intuitive successor.* Modernizing the GUI towards intuitiveness and simplicity will increase product appeal and value to our client base. While a the product flexibility and updatability are central concerns of the project, many of our users simply need the baseline product as is and their main concern will be navigating the program’s functions. Making the GUI as straightforward as possible will help enhance the innovative value of the product and thus motivate wide adoption, allowing researchers to devote valuable time directly towards experimental activity.

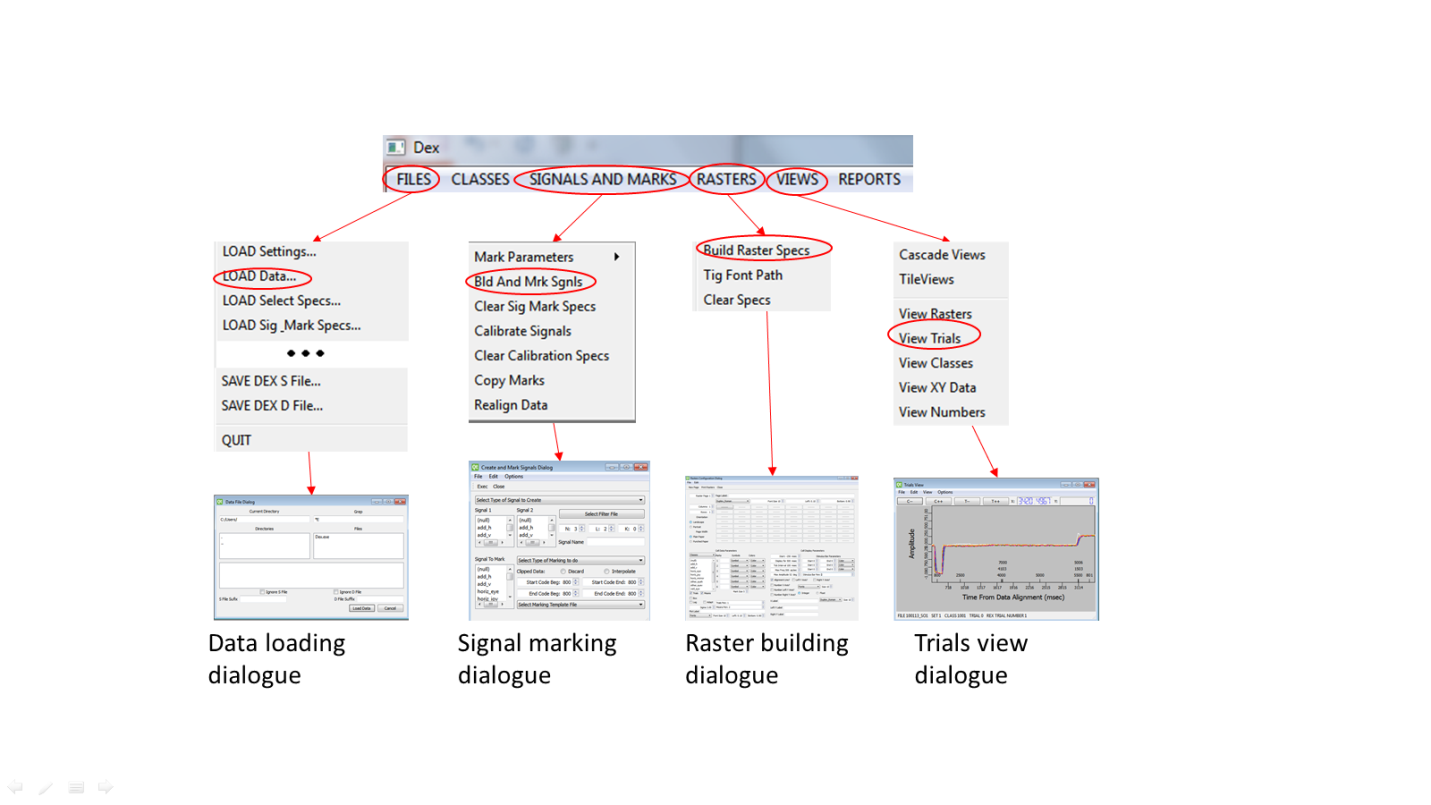
*3a. Implement largely self-explanatory features in an intuitive window layout.* This will include aligning interface design with commonly used layouts and feature presentation used in the Windows, Mac, and Linux environments. Feature captions displayed will be concise and simple, and ease of use will be increased by minimizing the buttons, drop-down menus, and check-boxes a user needs to select to accomplish a function, as well as logically-organized feature presentation based on user feedback. Much of the “intuitive” nature of our GUI layout will be assessed by users completing identical tasks in both Dex and our program and comparing the task-completion times and superfluous selections while navigating through the programs. These data will give us a quantitative assessment of our product and allow us to target areas of weakness in comparison to Dex. Our UCLA collaborators will help us in tailoring the layout toward the minds of cognitive neuroscience researchers for areas that require improvement. We will refer to and improve upon chart displays of data found in the Dex software, such as the ones found in Figure 4 to the right. This shows a small selection of the GUI layout we have mapped out in Dex. These function pathways show the series of windows and parameters the user must select in order to complete the end operation indicated by the last dialogue box. We plan to use this map a guide that we can restructure or cut down to create our more streamlined GUI. This map and tests of our GUI mock-ups will allow us to identify what enhancements are possible by our project timeline and set goals for our desired specifications in quantitatively comparing the two GUIs.

Figure 4. This figure illustrates the GUI layout of some of the function pathways in Dex. The red circles indicate an item selection and the red arrows indicate the pathway and window opened upon selecting the item.

*Specific Aim 4. Determine common file types in addition to Rex generated files and expand the data loading module for interoperabillity.* When writing the data loading module, we will initially target the binary data files generated by the most commonly used real-time data acquisition software, Rex (also used by our immediate client). However, to ensure widespread adoption of our product we aim for input versatility of any file types researchers may use. By collaborating with leading researchers in this field, we hope to determine and address any significant file types that may need to be analyzed.

*4a. Develop and ensure full compatibility with data files of development collaborators*. We will initially aim to meet functionality and compatibility specifications of our immediate collaborators in the field, Dr. Michele A. Basso and Dr. James Bisley. Close collaboration with these leading researchers throughout development will allow file-specific tailoring, immediate feedback to reduce re-design risk of making the same errors as existing programs, and facilitate software adoption by the clients. Data file content integrity analysis, file structure analysis, and controlled file interaction testing will be employed. We will try to obtain a variety of file sizes as well as batch files to assess the capability of our data loading module and confirm that the files are not compromised in any way when being converted in to matrices for inputting to other program modules. To accomplish matrix data allocation, we will also need to be able to read the binary files in an intelligible format to analyze the file for storage patterns. Once we determine the file structure, we can pull out the useful information from the binary data and allocate this information into matrices of our choosing. For Rex generated binary files, this is accomplished by the trdd function (discussed in detail later), however for other files we will need to write our own function equivalent to the trdd function which may take an appreciable amount of time. Thus, while versatility remains an important aim of our project, versifying file acceptance remains a more secondary goal since Rex is, to our knowledge, overwhelmingly favored for data acquisition.

*4b. Ensure program compatability with potential future file types to uphold universal adoption.* Following achievement of initial compatibility goals, we will seek and communicate with other prominent researchers in the field to obtain data samples and end-user feedback for increasing software file intake versatility as described in 4a. At this point we will also attempt implementation of a universal file reading or converting algorithm that can scan and detect file content and structure based upon representative data samples from these labs. This would be a very complicated algorithm and is considered a very low priority at the present.

*4. Risk mitigation by identifying potential pitfalls during the writing process and alternative product solutions.* Potential project problems and alternative strategies will vary accordingly. Perhaps the most severe problem that we may encounter will be the underestimation of coding difficulty such that a viable product is beyond the programming ability of the team. An issue of similar severity would be the discovery of a critical bug, or a poor programming methodology, far along into the development process, such that functionality is significantly reduced and code re-design bears a high cost. Another, less severe problem could be the simple time-restraint of the project timetable, which includes planning and programming skill acquisition prior to the coding process. To reduce the associated risk, we will, as stated, implement a third-party version control know as Git which will allow rollback to the most recent working version and thus will aid error identification. Moreover, the modular development philosophy will facilitate error isolation and functions can be individually added as needed. If the time-restraint proves too severe to deliver a completely comprehensive product on par with all capabilities ideally offered in comparison to Dex, we may consult the immediate clients to collectively decide on the essential functions necessary for the client’s research that are not provided in Dex and address those issues alone. The last resort alternative should a sufficiently severe problem arise (i.e. no chance of a viable novel product) would be to abandon novel software development and instead focus upon enhancement and de-bugging of the original C++ code of the Dex software. This would at least guarantee an end product to the client and have the added benefit of Dex’s name recognition.

**IV. PRELIMINARY RESULTS**

Initial work on our software product has yielded a number of promising results. While the nature of such a programming project inherently limits applicability of computational or experimental methods, in this first phase we have gained definitive preliminary results towards an end product. Among these includes the final selection of the versatile Python language following evaluation of different programming options for project adoption, such as the more complex, costly Matlab or lower-level C++ language. Project team members’ experience with the relative ease of learning Python helped confirm it as the suitable choice for our project. In the area of end-client collaboration, we were able to obtain initial feedback for our proposed project in terms of functionality and software adoption. Consultation with our immediate faculty clients, Dr. Basso and Dr. Bisley, yielded an initial plan for execution, in which basic functionalities to first aim for were determined. As noted by the researchers, this will help ensure that the product will at minimum possess core functionality in the case of time constraints, and useful and expandable upon further work. Beyond the UCLA campus, we also initiated contact with the labs of five other prominent researchers in our target research niche within the United States, and will continue to seek additional end-users domestically and internationally. Of those contacted, three labs have responded with interest in our work, expressing willingness to offer help in our project. Such help included providing direct product functionality and design feedback, as well as providing sample data files for testing. Although none of these labs have expressed immediate intention of necessarily adopting our software for their research, as can be expected at the current early phase, our team nevertheless aims to capitalize on all initiated collaboration. This will help us ensure the suitability of our data analysis software to fill the standard software void for this niche of cognitive neuroscience research, aiding widespread software adoption by labs.

These preparatory results surrounded and supported the initial work done in the actual coding process itself. In this regard, our team first obtained some insightful information regarding the structure and functionality of the original Dex software code. We did this by examining and generally mapping the C++ code of Dex, finding and noting its functions and features. This step not only gave our team a certain degree of intuition regarding which software features to code and how to go about such coding, but most importantly, afforded us an opportunity to note shortcomings of the program interface and functionality upon which we aim to improve. While less, working code was achieved in actuality than was projected in the first few weeks of this project stage, studying and characterizing the old product which is due for improvement effectively equipped us with the knowledge and familiarity necessary for heading in the right direction. This examination process was furthermore applied to the original Dex GUI. For example, interface disparities between Dex software versions and the outdated user manual were noted as significant areas for improvement, as well as were the low level of interface fluidity and logic.

Initial code-writing work following the above resulted in the creation of our first module, which achieves file and data loading into our program. Although seemingly trivial, this feature is the one upon which the entire remainder of our code and interface depends on for success. Creating this module involved examining the file format into which the standard online data recording software, Rex, records experimental data. Testing with sample data files provided by our faculty collaborators, we were able to achieve raw data extraction by writing a wrapper function in Python that reads Rex-generated files. This function in our module is essentially patterned after a similar section of code in Rex known as the Trials Rex Data Dumper (trdd). By creating a Python equivalent of this data-dumping function, we enabled first module to extract raw data into understandable, organized text files of strings [24]. The module then stores the accessed information in a known matrix arrangement for simple referencing by future functionality modules. We are able to confirm the accuracy of data interpretation by checking alignment of our data extraction results with what is known to be the content of the sample data files. Our research end-clients have done previous work with these files for analysis and even for publications.

**REFERENCES**

[1] M.A. Basso, X. Li, “Cues to move increase information in superior colliculus tuning curves,” *Journal of Neurophysiology*, vol. 106(2), May 2011, pp. 690-703.

[2] B. Kim, M.A. Basso, “A probabilistic strategy for understanding action selection,” *Journal of Neuroscience*, vol. 30(6), Feb. 2010, pp. 2340-2355.

[3] R. Kiani, A.K. Churchland, M.N. Shadlen, “Integration of direction cues is invariant to the temporal gap between them,” *Journal of Neuroscience*, vol. 33(42), Oct. 2013, pp. 16483-16489.

[4] D.A. Bulkin, J.M. Groh, “Distribution of eye position information in the monkey inferior colliculus,” *Journal of Neurophysiology*, vol. 107(3), Oct. 2011, pp. 785-795.

[5] X. Li, M.A. Basso, “Competitive stimulus interactions within single response fields of superior colliculus neurons,” Journal of Neuroscience, vol. 25(40), Dec. 7, pp. 11357-113573.

[6] D. Robinson, A. Fughs, “A method for measuring horizontal and vertical eye movement chronically in the monkey,” *Journal of Applied Physiology*, vol. 21(3), May 1996, pp. 1068-1070.

[7] “Neuroscience and eye tracking”. Tobii Eye Tracking Research.

<<http://www.tobii.com/en/eye-tracking-research/global/research/neuropsychology/>>, 2013.

[8] The Numbers Count: Mental Disorders in America [Internet]. National Institute of Mental Health. 2013 – [Cited 2013 Dec]. Available from: [www.nimh.nih.gov/health/publications/the-numbers-count-mental-disorders-in-america/index.shtml](http://www.nimh.nih.gov/health/publications/the-numbers-count-mental-disorders-in-america/index.shtml)

[9] Warning Signs of Mental Illness [Internet]. Arlington, Va. American Psychiatric Association. 2013 – [Cited 2013 Dec]. Available from: [www.psych.org/mental-health/more-topics/warning-signs-of-mental-illness](http://www.psych.org/mental-health/more-topics/warning-signs-of-mental-illness)

[10] National Eye Institute Public Data Share [Internet]. Bethesda, MD. National Institutes of Health. 2013 – [cited 2013 Nov 21]. Available from: <http://datashare.nei.nih.gov/LsrDirectoryServlet>.

[11] J.W. McClurkin, *Dex User’s Guide*, Laboratory of Sensorimotor Research, NIH.

[12] Matlab [Internet]. Mathworks. 2013 [cited 2013 Dec]. Available from:

<http://www.mathworks.com/products/matlab/>

[13] Application Domains [Internet]. Python Software Foundation. 2013 [cited 2013 Dec]. Available from: http://www.python.org/about/apps/

[13] M.F. Sanner, “Python: a programming language for software integration and development,” Journal of Molecular Graphics and Modeling, vol. 17, Feb. 1999, pp. 57-61.

[14] M. Andrews, J. Diaz, “And now for something completely different: Python in cognitive science,” Division of Psychology, Nottingham Trent University, CogSci 2012.

[15] About Python [Internet]. Python Software Foundation. 2013 [cited 2013 Dec]. Available from: [www.python.org/about/](http://www.python.org/about/)

[16] Ousterhout John K. Scripting: Higher level programming language for the 21st century. IEEE Computer Magazine March 1998.

[17] Comparing Python to other Languages [Internet]. Python Software Foundation. 1997 [cited 2013 Dec]. Available from: <http://www.python.org/doc/essays/comparisons.html>

[18] Modules [Interest]. Python Software Foundation. 2013 [cited 2013 Dec]. Available from: http://docs.python.org/2/tutorial/modules.html

[19] Ousterhout John K. Scripting: Higher level programming language for the 21st century. IEEE Computer Magazine March 1998.

[20] Comparing Python to other Languages [Internet]. Python Software Foundation. 1997 [cited 2013 Dec]. Available from: <http://www.python.org/doc/essays/comparisons.html>

[21] T.E. Oliphant, “Python for Scientific Computing,” *Computing in Science and Engineering*, vol. 9(3), May 2010, pp. 10-20.

[22] Zhao M, Lee W-P, Garrison EP, Marth GT (2013) SSW Library: An SIMD Smith-Waterman C/C++ Library for Use in Genomic Applications. PLoS ONE 8(12): e82138. doi:10.1371/journal.pone.0082138

[23] Zhu H, Luo M (2013) Chemical structure informing statistical hypothesis

testing in metabolomics. Bioinformatics doi: 10.1093/bioinformatics/btt708

[24] Subprocess Management [Internet]. Python Software Foundation. 2013 [cited 2013 Dec]. Available from: <http://docs.python.org/2/library/subprocess.html>