A COMPUTATIONAL TOOL FOR FDG KINETIC PARAMETER ESTIMATION USING DYNAMIC POSITRON EMISSION TOMOGRAPHY (PET) DATA TO AID IN ALZHEIMER'S RESEARCH

by
Guadalupe Ayala

An Internship Report Presented in Partial Fulfillment
of the Requirements for the Degree
Professional Science Master in Computational Biosciences

Report 04-17

ARIZONA STATE UNIVERSITY
May 2005

A COMPUTATIONAL TOOL FOR FDG KINETIC PARAMETER ESTIMATION USING DYNAMIC POSITRON EMISSION TOMOGRAPHY (PET) DATA TO AID IN ALZHEIMER'S RESEARCH

by

Guadalupe Ayala

has been approved

May 2005

APPROVED:

______________________________, Chair

____

____

Supervisory Committee

ABSTRACT

This report describes the goals for the Internship Project, the software development process, methods utilized, the significance of work, and conclusions. The major goals during the internship were the following:

- The design and implementation of a ®MATLAB's[1] based application for kinetic parameter estimation of Fluoro-Deoxy-Glucose (FDG) metabolism.
- The completion of a user manual provided as Appendix B of this report.
- The development of a Web site to publish the code and user manual.

This report is an overview of how all of these milestones were accomplished and serves as a wrapper report for the accomplished user manual.

keywords: PET, FDG Metabolism, Parameter Estimation, Alzheimer's, bigK, PATLAK, GLLS

[1] ®MATLAB is a registered trade mark of The Mathworks, Inc.

ACKNOWLEDGMENTS

This work was supported in part by the Arizona Alzheimer's Disease Research Center which is funded by the Arizona Department of Health Services, NIH grant EB 2553301, and the Underrepresented Graduate Enrichment Match (UGEM) Fellowship award to Guadalupe Ayala.

Advisors: Dr. Rosemary Renaut, Dr. Bradford Kirkman-Liff, Dr. Hongbin Guo

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Goals of Internship Project

As mentioned before the major goals of the Internship project were the following:

- The design and implementation of an application for kinetic parameter estimation of Fluoro-Deoxy-Glucose (FDG) metabolism. We aimed to build an application that would allow the user to estimate the parameters with a lot of flexibility on choice of inputs and options. The application may receive several types of inputs and would estimate kinetic parameters representing FDG metabolism. Inputs for the application are TTAC, PTAC and options which include constraints, filtering of input data, or estimation model. All of these inputs along with outputs are described in detail in the application user manual also provided in Appendix B.
- The completion of a user manual. The user manual was written as a collaborative effort. The manual was written using Latex, and several revisions were made before completion. The manual includes information on the practical use of the application. It gives the user background information on the estimation kinetic model, input and output formats, minimum hardware and software requirements, as well as installation and execution procedures. The manual gives a "How-To" guide for all the required tasks in order to start a run. It tells the users how to read inputs, specify constraints and model to start a run. One of the chapters walks the user through an slice run, so the user can familiarize with the application. It also explains in detail all the different options available.
- The development of a web site to publish the code and user manual. I developed the web site in order to make the code and user manual available for public use. The major pages for the web site are shown in Appendix A.

CHAPTER 2

Software Development

The process of software development consisted of the following procedures:

1. Understanding Compartmental Kinetic Model for Parameter Estimation. Refer to Appendix B, for more information on this model.
2. Write Requirements Document.
3. Identify Inputs/Outputs.
4. Identify User options.
5. Build Shared Code Repository.
6. Error Handling: We had to plan for different cases in which application should alert the users of faulty behaviors.
7. Building Prototype.
8. Testing Prototype.

   This section will give an overview of how these procedures were accomplished, as well as which tools were used and what challenges were encountered.

1. Requirements Document

The software development process started by writing the requirements document, identifying inputs/outputs and user options. The requirements document was a great starting point. In order to write the requirements document, I met with my advisors to brainstorm about the different uses of the application, and answer several critical questions for the application design. These are some of the questions that arose in our discussions:

1. What were the Inputs and Outputs for the Application?
2. What should the format for the Inputs and Outputs be?
3. How should we save the results?

4. What naming conventions should use?

5. What user options and features should we include?

6. What should we display on interface?

7. What information should we save for each run?

8. What run information is required in order to replicate a run?

9. How were we going to use existing code?

Careful examination to the answers of these questions allowed me to design the application. Discussions with my advisors helped me to look at the problem from different perspectives. We analyzed the problem from the Computer science, Mathematics, and Bio-Medical points of view, and came up with answers to these questions. The answers to these questions were critical to the user interface design.

2. Code Repository

Since we were working as a team, it was critical that we put the code in a shared code repository. Therefore, I built a CVS based code repository. The CVS code repository was very useful for the following:

1. Document code modifications.

2. Maintain a copy of old file versions, in case we needed to revert back

3. Be able to notify each other if we were working on a file, so we did not overwrite each other changes.

4. To be able to checkout the latest code, if a mistake was made.

For more information on CVS go to online reference [1].

3. Error Handling

It was essential to always keep in mind how to approach users possible faulty behaviors. I had to come up with different cases in which we had to alert the users of processing errors and direct them on how to resolve them. These are some of the cases where error handling was required:

1. When users try to start a run and have not yet entered required inputs. The application warns the users and directs them to enter the missing inputs by telling them which inputs are missing.

2. When the users try to enter inputs inconsistent with the required inputs for a given type of run. This is prevented by disabling menus and buttons that are not required for a given type of run. For example, the pop-up menu for constraint options is disabled for cluster runs.

3. When the users try to view results before starting and completing the run, the application tells that user that no data is available.

These error handling approaches allow the application to run more smoothly by preventing the users from causing errors or re-directing them to the correct procedures.

4. Building Prototype

Building the prototype was a cyclic approach. First, I implemented the initial prototype based solely on the requirements document. Then, it was continuously modified until we had what we desired. The process required several meetings to demo the prototype and discuss any pertinent changes. After each meeting, I made the modifications and met again to discuss any further changes. This process allow us to add some unforseen options and features that had not been included in the initial design.

5. Testing Prototype

Finally, I had just a short amount of time to test the prototype. Due to time constraints, I performed tests of only the most common cases and fixed any problems during testing. There is a large number of test cases due to the diversity of user options and models, and the application is very computational intensive. It takes approximately a 1 min/slice run. Therefore, it would take a long time to go through all of the test cases. Perhaps, that is something that can be done in future work.

6. Development and Technical Writing Tools

There were several tools critical for the software development. I had to train myself on them, but it was a great opportunity to acquire new skills. The following are the tools that I had master:

1. GUIDE was very helpful in designing the interface prototype. GUIDE is a ®MATLAB user interface development tool which allows developers to graphically design a user interfaces by dragging and dropping objects, such as labels, listboxes, and popupmenus.

2. CVS is a Linux based version control system, I had to do my own research on how to use it, but I learned that it was a great tool when working in a team environment.

3. Latex is a tool for the development of written materials. It was specially useful when writing the user manual, as it automatically organizes documents into sections and has a math environment that facilitates the use of mathematical notation.

7. Challenges and Problems Encountered

There were several challenges and problems that I had to overcome during the internship. These are some of them:

1. Dynamic PET data was difficult to deal with, because there was a high amount of noise at the early time frames. We decided the best solution to this problem was to mark pixels that did not give accurate results.

2. ®MATLAB does not support parallel processing, so while the application is running the user cannot interact with it. Therefore, the user has to wait until the application finishes the run. Future work might be necessary to make the application able to run independently.

3. We had to integrate new code with existing code. I had to revise the existing code to fit it to the application design.

CHAPTER 3

Significance of Internship and Future Work

1. Significance of Internship

This internship was a great opportunity to acquire excellent Technical and Professional skills. I gained hands-on experience on the whole development cycle. Also, I gained experience in working in an interdisciplinary team. I mastered several technical and development tools, such as GUIDE, Latex, CVS.

Also, the products of this internship which are the FDG kinetic parameter estimation application, the user manual, and the web site will potentially be beneficial to Alzheimer's Disease Researchers. The web site could be an important portal for Alzheimer's researchers to download and make use of this application. Furthermore, they will also have access to the user manual which will guide them on how to use the application. Also, they might be able to reuse the code to customize it for their testing purposes. Perhaps, new features may be added in the near future.

2. Conclusion and Future Work

In conclusion, even though this application has already been useful in evaluating different estimation models, there is still some room for improvement. Future work may include the following:

1. More filtering options may be added to try to reduce the initial noise in dynamic PET scans.
2. Other estimation methods may be added to compare and contrast current results.
3. Develop a set of new test cases, and perform a thorough testing of the application.
4. Add a feature, so the users can automatically save results as *.eps files.
5. Convert the application to a independent executable from ®MATLAB [2].

   These are just a few of the improvements that can be made, but the possibilities are endless.

Bibliography

[1] *Cvs online manual*, CVS Web Site, 2005, Concurrent Versions System.

[2] *Mathlab online reference*, MathWorks Web Site, 2005, MATLAB is a registered mark of MathWorks, Inc.

APPENDIX A
WEB SITE MAJOR PAGES

This is the web page where you can download code and user manual for the application. Also, you may find more information on related publications, and contact information.

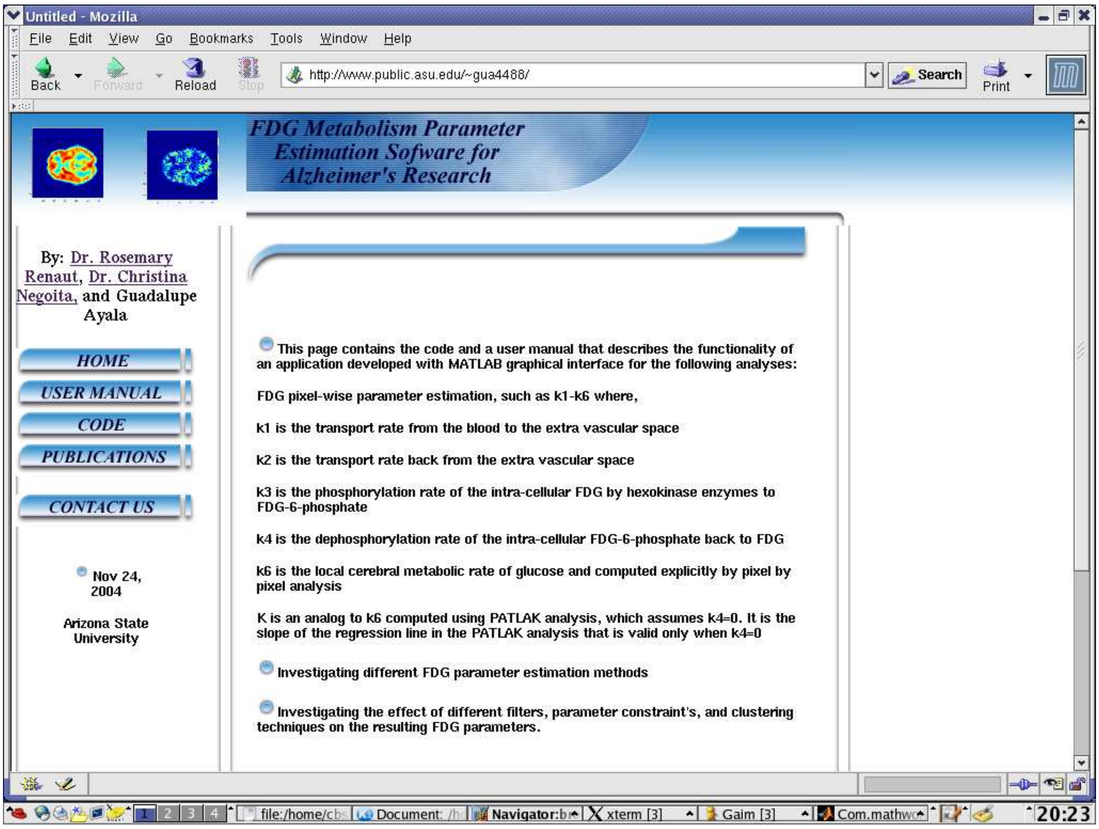

Figure 1. Home Page of Web Site to Download Code and Manual

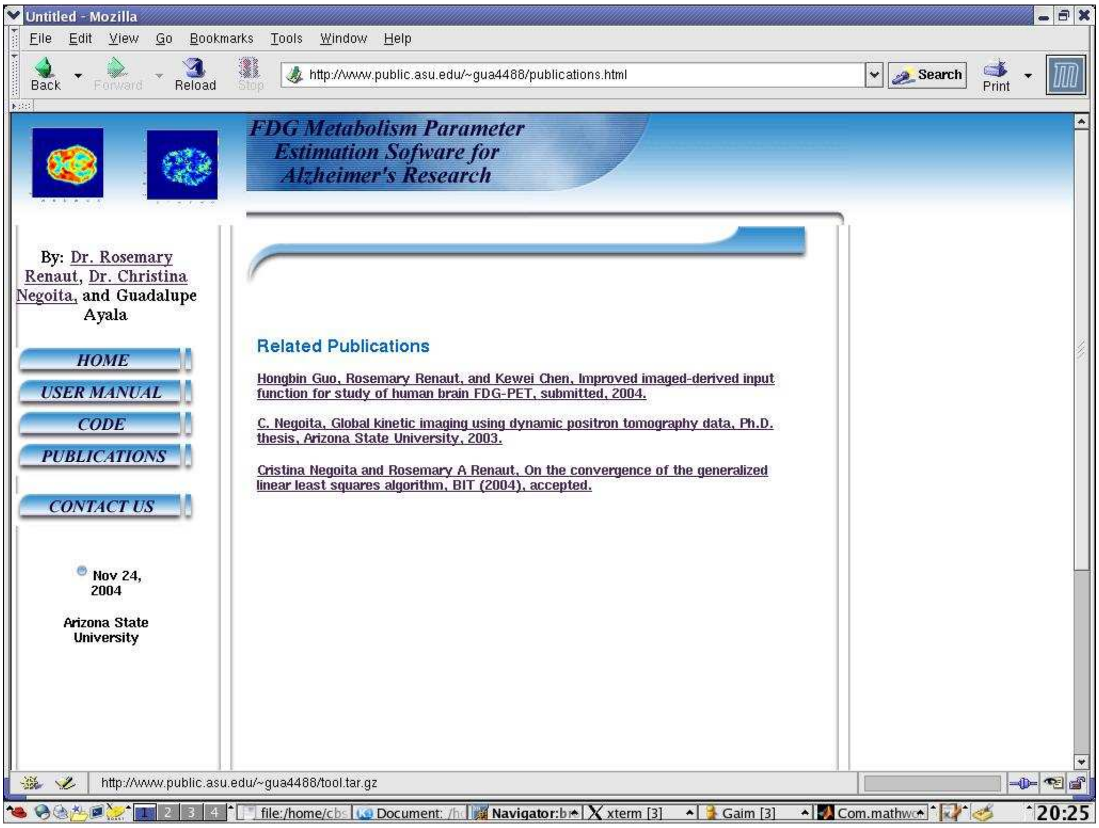

Figure 2. Web Page for Related Publications

APPENDIX B
USER MANUAL: PARAMETER ESTIMATION FOR A COMPARTMENTAL TRACER KINETIC MODEL APPLIED TO PET DATA