MassMatrix ProtDB: A Laboratory Information Management System
Adam Champion, Jonathan Ou-Yang, Michael McGrath, Derick Chan

CSE 758 Capstone Project
Instructor: Dr. Rajiv Ramanath, Department of Computer Science and Engineering
Sponsor: Dr. Michael Freitas, Human Cancer Genetics, The Ohio State University

Introduction

- Researchers collaborate to cure cancer (Fig. 1)
  - Collect cells from organisms
  - Perform experiments to find protein “markers” for disease
  - Mine and analyze resulting data files for these patterns
- Problems:
  - Data files are huge (~100 MB) — can’t analyze manually
  - Massively-automated data processing is necessary for statistical analysis
  - Files are not stored at central location
- Solution: Web-based content management system (CMS)
  - Researchers upload data files, perform protein searches
  - Biostatisticians mine files for statistical patterns (Fig. 2)

Requirements

- Process agility diagram is shown in Fig. 3
- Functional Requirements
  - Develop CMS that requires authentication to be use
  - Ensure strict user access controls (UACs) are in place
  - Implement user-navigable create, retrieve, update, delete (CRUD) functionality with client-specified database schemas for data and users (Fig. 4)
  - Retain user-supplied data indefinitely (future)
  - Integrate database with protein search application (future)
- Non-Functional Requirements
  - Implement using Linux, MySQL, Apache, Python (LAMP) stack and Django Web framework on server
  - Provide security for data, user passwords; backup data
  - Provide aesthetically-pleasing user experience

System Architecture

- Django uses a “hybrid” Model-View-Controller (MVC) pattern called Model-View-Template (MTV) (Fig. 5)
  - In MVC, the model is the actual data, the view is the user’s “view” thereof, and the controller mediates between them
  - In Django, the model is a Python-specified database, the view is a Python script that acts as the MVC CONTROLLER, and the template is the user-viewed HTML file (with some controller logic)
- User’s workflow with the Web application is shown in Fig. 6

User Interface

- User visits http://www.massmatrix.net, sees page (Fig. 7(a))
- User can see all his Projects, others’ shared Projects (Fig. 7(b))
- User can “drill down” through Project ➔ Sample ➔ Experiment ➔ Data Set ➔ Result hierarchy (Fig. 7(c))
- Administrator can log in to Django Admin page (Fig. 7(d))

Future Work

- Data retention; security; integration with search engine