GUI for PluMA: Plugin-Based Microbiome Analysis

Senior Project, 2019, Spring

Student: Bhavyta Chauhan[1], Mentor: Trevor Cickovski[1], Instructor: Masoud Sadjadi[1]
[1]Florida International University

ABSTRACT

PluMA is a software application developed under the Bioinformatics Research Group to construct flexible and lightweight analysis pipelines. These pipelines can be used by a developer to implement a new algorithm in a programming language of their choice, and easily test and debug within a larger pipeline.

PluMA accomplishes this through plugins, and has a large collection available in its plugin pool, implemented in various programming languages for both the CPU and GPU.

PluMA GUI is a desktop application that provides an interface for the user to assemble and run pipelines through such operations as searching the online plugin pool, downloading plugins, assembling pipelines by dragging and dropping installed plugins, uploading input/ output files associated with a plugin, and saving the configuration file of an assembled pipeline.

MY ROLE

I was responsible for the following features:

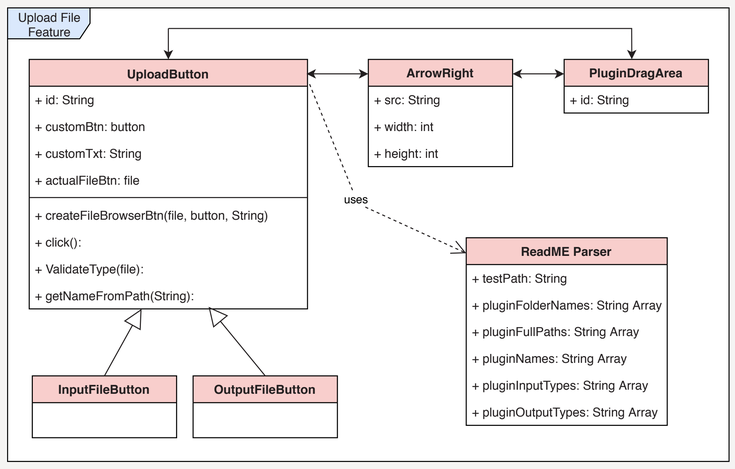
- Buttons for uploading input and output files for the first and last plugins in the pipeline respectively
- Check if uploaded file type matches allowed file type with respect to the specific plugin
- Display static arrows between each plugin in the pipeline assembly area to better convey the structure of the pipeline software visually

TECHNOLOGIES USED

The following technologies were employed to implement the features:

- HTML/ Javascript/ CSS
- Node.js library
- jQuery library
- Electron Desktop App

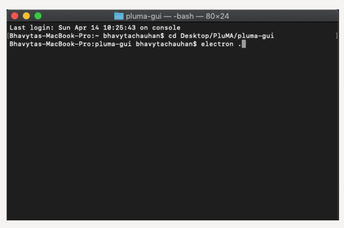
CLASS DESIGN



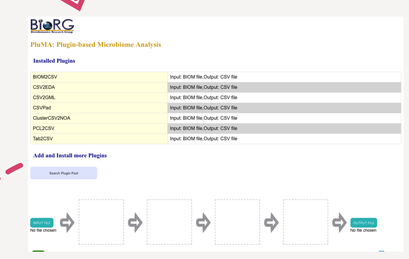
Minimal class design depicting the different classes in the subsystem of uploading a file and their interactions with each other.

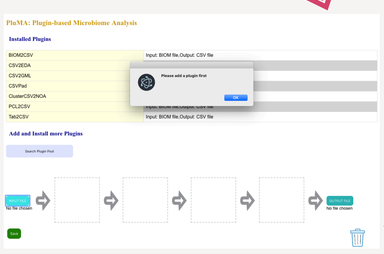
SCREENSHOTS

Launch the App



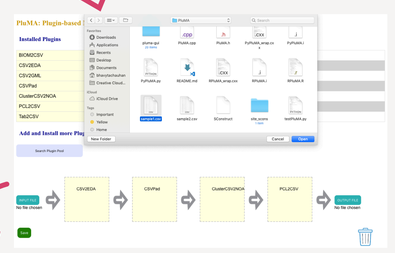
Start Page

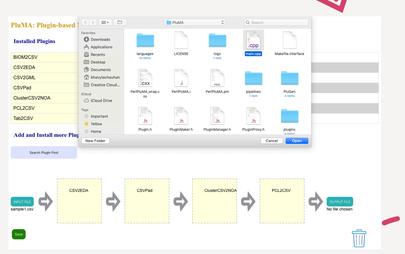




Alert when user tries to upload a file without adding a plugin first

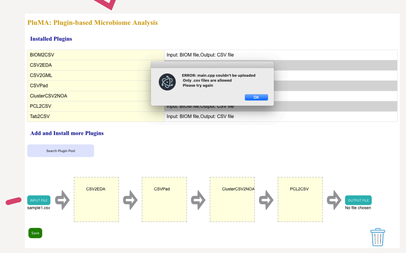
Open file browser window and choose correct input file for upload





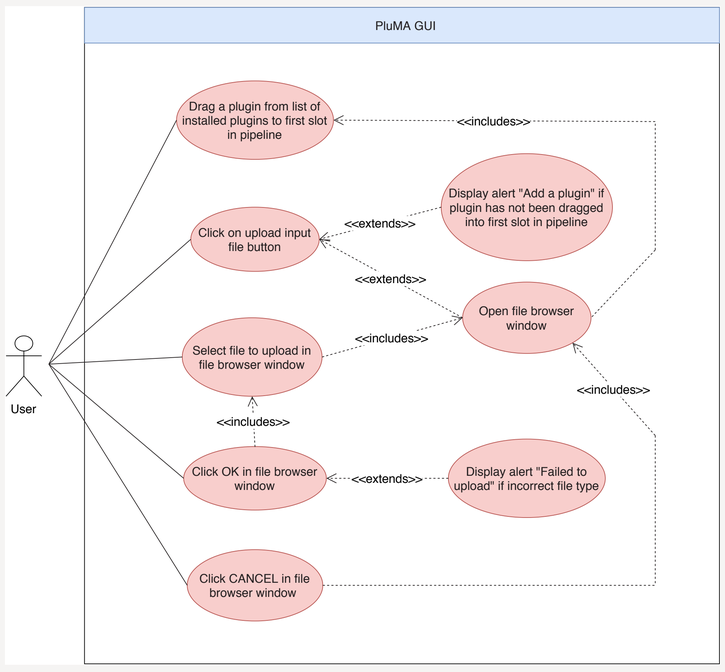
Choose output file for upload

Alert when user chooses incorrect file type for upload and failure to upload



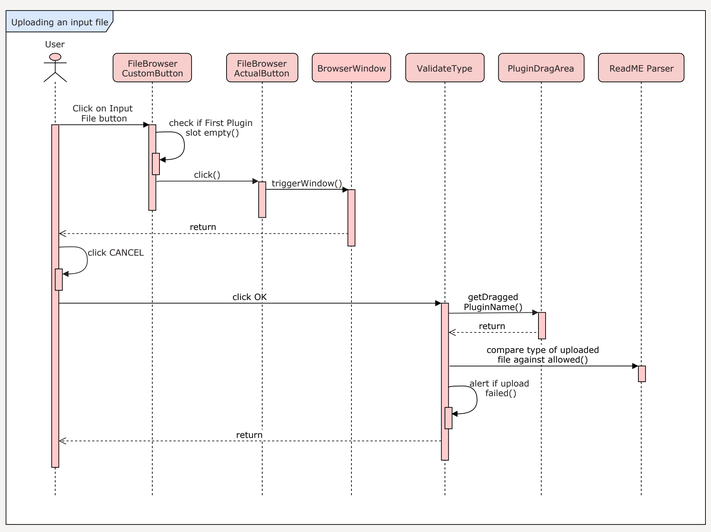
Final assembled pipeline

REQUIREMENTS



The requirements for uploading an input file shown in terms of use cases. The system has one actor: the user.

IMPLEMENTATION



Sequence diagram for uploading an input file is shown above.

The FileBrowserCustom Button is used to display a customized upload button and interacts with the ActualButton to trigger the Browser Window.

The ReadME Parser is used to validate the uploaded file type and interacts with the PluginDragArea through ValidateType.

FUTURE WORK

This is Version 1.0 of the PluMA GUI App and there are many areas that can be built upon:

- Allowing the user to upload input/ output files for intermediate plugins in the pipeline if needed
- Making the arrows between plugins dynamic to allow the user to choose which plugins supply input/ output files to each other

ACKNOWLEDGMENTS

This poster is based upon the efforts of Dr. Trevor Cickovski and the Bioinformatics Research Group (http://biorg.cs.fiu.edu/). I am also grateful to my fellow team members, Cesia Bulnes and Rishabh Vaidya.