RNAStructViz

Main application files:

1. Main.cpp – Starts the application

DO NOT CHANGE; you don’t even have to look at this code.

2. RNAStructViz.cpp – Main application, singleton

This is where you might add code for adding stats windows, if you decide to do a similar thing to how diagram windows are added.

3. MainWindow.cpp – GUI for the application, singleton

This is where you would have to add the button for opening stats windows. Make sure that it correlates to the “selectedFolder”, similar to opening a diagram window.

It is also a good reference for working with the reference structure, such as setting and getting, and working with the round buttons for each structure in the stats window. Look at the “AddFolder” method and the “SelectCallback” method.

4. StructureManager.cpp - Manages the active structures in the application (“m\_structures”)

This is where you will have to add the code for updating stats windows (adding new structures), similar to folder windows and diagram windows. Look at “AddNewStructure” method.

Structure Files:

1. RNAStructure.cpp – This is the code that represents a structure.

DO NOT CHANGE, but know what the “GetBaseAT” method and the “GetFilename” method do. Also understand how the two enums “Base” and “Truth” work.

2. FolderStructure.h – This holds the structure of a folder.

This probably doesn’t need to be changed, but at least take a look at it. You may need to add to it if you are going to do the stats windows similar to the folder windows.

Diagram windows files:

1. DiagramWindow.cpp – This is the main window for showing the circular Feynman diagram of the structures.

DO NOT CHANGE; but this is a good reference for controlling structures for the stats window.

2. GLWindow.cpp – This is the code that does the drawing of the circular Feynman diagram.

DO NOT CHANGE; you don’t even have to look at this code.

Folder window files:

1. FolderWindow.cpp – This is the window that simulates opening a “folder”.

This is where you can view the base pairs of each structure and remove structures.

Here you will have to add code for updating the stats windows when you remove a structure from the folder. Look at the “RemoveCallback” method.

2. InputWindow.cpp – This is the window for renaming a “folder”

DO NOT CHANGE; you don’t even have to look at this code.

StatsWindow.cpp - what needs to be done:

Main: Write and integrate the StatsWindow.cpp code.

* The initial setup is written, but no real content is there, even the content included may not be correct in it’s entirety.

1. Calculating the actual statistics

* The “ClearTruthValues” and the “UpdateTruthValues” methods, I rewrote so that it focuses on a single folder instead of all of the structures in the application. “UpdateTruthValues” calculates the true positives, false positives, true negatives, and false negatives. For calculating the number of base pairs look at the “ComputeNumPairs” in DiagramWindow.cpp.
* Code for computing the number of conflicts and contradictions needs to be added, probably in the “UpdateTruthValues” method. (Included is a paper on how to calculate the statistics as well as perl files that has code which should help in calculating the number of conflicts and contradictions, I’ve also included notes that I’ve made while running the them).
* Figure out a way to hold all the statistics of a structure vs. the reference structure. Perhaps a new type of structure then having a vector of them.

2. Decide how the application manages the stats windows. It can be similar to how the application manages diagram windows or how it manages folder windows; or something totally different.

3. The design of the stats window also has to be decided upon.

My thoughts

* The structures in the folder could be listed in a separate box on the right side along with the round buttons that will be used to choose a reference structure, with a “Calculate” button on top. Look at “AddFolder” in MainWindow.cpp for how to align widgets. To the left would be the tabs for holding the graphs.
* The graphs for showing the number of base pairs, true positives, false positives, true negatives, and false negatives would be in one tab, and the sensitivity and selectivity in another tab (I’ve included separate code of how to work with tabs, along with an idea of how to simulate graphs).

However be creative, this doesn’t have to be design of the stats window, it is only what I had imagined.

Recompiling the Code:

1. While working with the code it is easier just to use a single makefile instead of constantly having to reconfigure the code. In order to do this use the remove the Makefile in the src directory and rename the Old\_Makefile to Makefile and configure the paths in it, so that they point to the correct libraries.

2. However after everything is complete you have to reconfigure the code.

Make sure to keep the following files:

configure.ac – use as a reference for steps 3 and 6

Makefile.am (all of them) – need to update source files

rnastructviz\_config.h.in – add before doing step 7

Then go through the tutorial found at <http://mij.oltrelinux.com/devel/autoconf-automake/>

Last thing, I left the “Test” button in to help with seeing what is going on behind the scenes. Currently it shows the folders and the corresponding files, as well as the “selected folder”.