Subject: Similarity heatmap
Posted by cfuttrup on Wed, 02 Dec 2020 12:54:23 GMT
View Forum Message <> Reply to Message

We are looking at the possibility of using openmolecules as a way of determining with proteins we need to analyse closer based on similarity and heatmap. I can see that Openmolecules doesn't have a heapmap currently and I am wondering, were to look to implement this functionality.

Subject: Re: Similarity heatmap
Posted by thomas on Wed, 02 Dec 2020 19:33:17 GMT
View Forum Message <> Reply to Message

Currently, DataWarrior cannot show heatmaps also it is able to perform a hierarchical clustering. Probably, heatmaps will come some time in the future. Protein similarity is another thing that is not likely to be added soon, because DataWarrior focuses on small molecules. Proteins will play a role soon, but as target for docking only.

Subject: Re: Similarity heatmap
Posted by cfuttrup on Thu, 03 Dec 2020 09:47:56 GMT
View Forum Message <> Reply to Message

If I wanted to contribute and implement a heatmap where so I look and begin?

Subject: Re: Similarity heatmap
Posted by thomas on Fri, 04 Dec 2020 15:29:14 GMT
View Forum Message <> Reply to Message

It would need two parts: First the columns from a hierarchical clustering result would have to contain more information: at least a reference to the parent column and the similarity value to its partner in the similarity graph. The clustering in DataWarrior is hierarchical, but doesn't store these values. One might reconsider, whether there are alternatives to the algorithm, which is brute force and needs the full similarity matrix.

Second and much more effort: Currently all 2D-views are calculated and drawn by one class JVisualization2D (with support of its parent class JVisualization). It is overdue to break up the various 2-dimensional views (scatter, bars, boxes, whiskers, pies) into separate classes, which still share a lot of properties (marker size, color, labels, font size, ...), but I didn't ever have the time, because a lot would be involved including reorganising the way view properties are saved in templates and how macro properties interact with views.

The fastest, but messy approach would be to add another preferred view (heat map) to the JVisualization2D class, but working on the foundation first would be the better approach. I suggest you look into the source code and get a picture. If you then really like to contribute substancial time, then we should closer discuss options. Of course, I am grateful for any support.