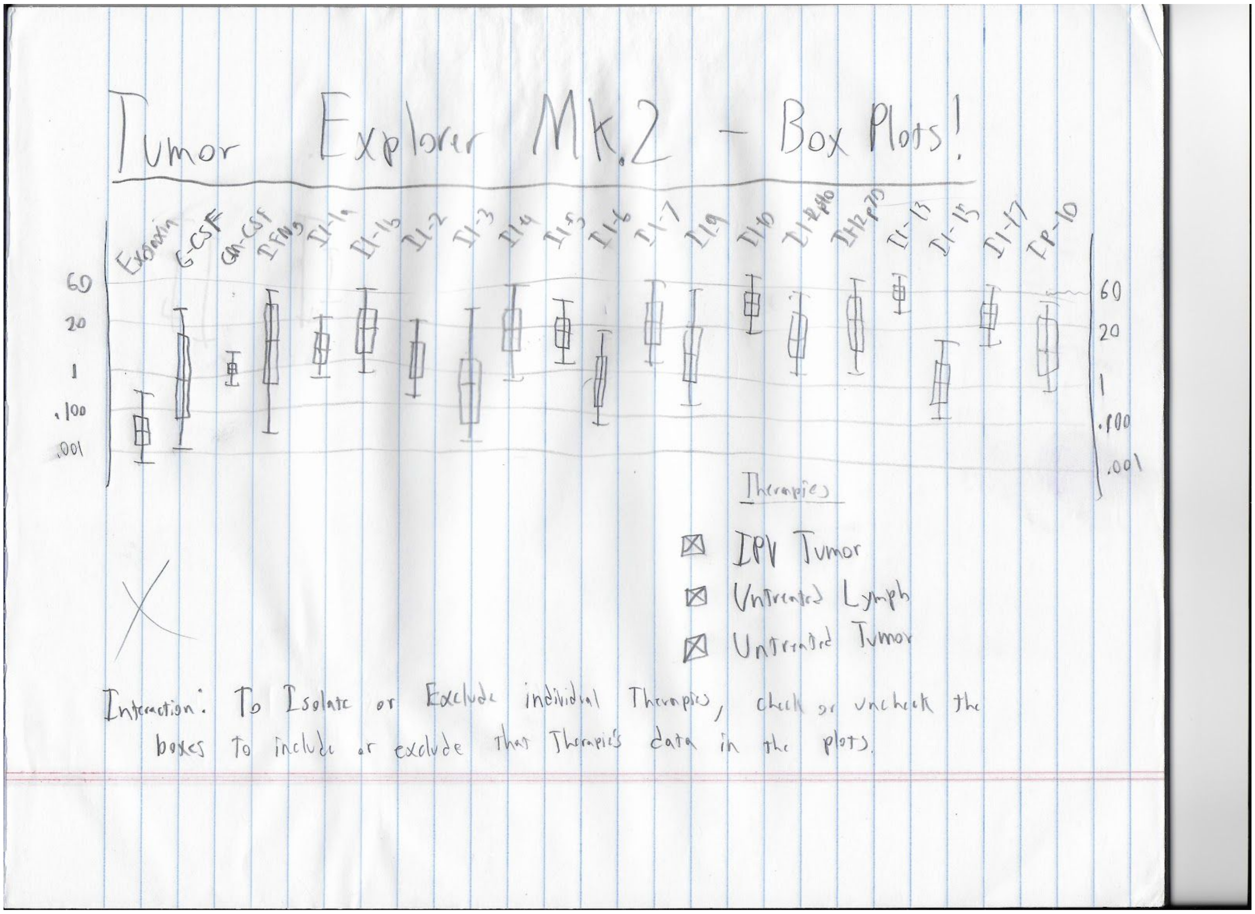
Assignment 3: MultiD with Parallel Coordinates: Overview Technique

Part 1:

Of the three issues our groups were to consider, we found the clutter problem to be the most straightforward to address. We considered a number of ways to consolidate data, or spread it out, to make more details visible. Or, alternately, cut down on what is being displayed to focus on what info is directly needed. After a number of mostly non-functional possibilities, we settled on using box & whisker plots to summarize the distribution of data at each vertical therapy. We also suggested allowing the plot to expand horizontally and spread out using a scroll bar, since in most cases seeing all therapies at once wouldn't be as important as having clear sight to details.

Below is a sketch we made to summarize the idea. One fault of our initial design is that is did away with the parallel aspect of the plot entirely, because we did not fully understand what useful information was being gleaned from those elements.



Interaction: To Isolate or Exclude individual Therapies, check or uncheck the boxes to include or exclude that Therapie's data in the plots.

In the class discussion, most people appreciated the box and whisker plot concept, particularly as another group came up with a similar plan. However, it was explained that the parallel lines, when combined with the movability of the individual protein expression locations, helped in identifying cases of co-expression, when different proteins often/always are expressed in correlated amounts. The final decision of the class, to preserve this, was to overlay the box and whisker plots on TOP of the existing parallel plot, in an attempt to preserve the original intent while also providing the statistical summary.

Part 2:   Design

When approaching this project, I first attempted to find pre-existing box and whisker plot examples in D3 to either integrate semi-directly or else get some insight into implementation possibilities. I found a good number of examples, but nearly all of them were variants or modifications of Mike Bostock's implementation, as seen here: https://bl.ocks.org/mbostock/4061502 . I spent some time attempting to integrate it, and in the end had to abandon that as there were just too many conflicts in the ways that the original Tumor visualization and Bostock's code were approached. Additionally, I wasn't sure I could adapt the Bostock code to display in a logarithmic scale.

Starting from scratch, I began by trying to find out where on earth in the tumor vis code I could introduce my addition. It took a long time and an unfortunate amount of fruitless testing until I could get even a basic square introduced and actually displayed. I think that while there was a good amount of commenting in the code, I was lacking in some basic background to make it make sense. The difference in scale from our first project to this was quite large.

Having managed to display a square, I built some functions to aggregate the data from the selection of trials being displayed, perform the statistical analysis, and provide them to the boxplot drawing function. I also appropriated a quartile calculating function to keep somewhat efficient.

I made a new canvas layer to make the boxplots in, to provide some isolation and reduce conflicts. And after a good bit of trial and error, I determined how to use the visualizations scaling functions to fit my plots to the scale of the parallel plot, and how to position them on the scale lines. By placing the call to the box-drawing function in the right place, it gets re-drawn whenever the visualization is rearranged or therapies are turned on or off, though I needed to find the right place to put a full clear of the canvas inbetween re-drawings.

I made a lot of small tweaks to make the boxplots more visible and clear. I tried to get the boxplots placed OVER the scale lines for better visibility, but because that obscured the svg layer, it stopped responding to interaction, and the canvas implementation I had made wasn't easily moved to the svg layer.

I do have a concern as far as the assignment requirements as described on the Data Vis webpage however. It sounds like a separate box plot is requested for each therapy/protein combo, which would mean up to 12 x 33, or 396 separate boxplots crowded into the space we had already determined was too cluttered to easily comprehend… This seemed completely counter to the point of the exercise. In addition, either the plots could be all stacked 12 in one spot so none are clear, or else they could be spread out vertically, where they would no longer correspond to the axis provided. As I implemented it (prior to re-reading the assignment and noticing this aspect of the requirements and how it conflicted with how I had initially understood it), there is one box plot for each protein, displaying the aggregated distribution of all therapies currently being displayed. Any single therapy can be inspected by isolating it the same way the original visualization could turn on and off each therapy. This does mean that one has to do some clicking to get down to an individual therapy, but it is at least not a huge jumble of unintelligible bits.

I like the idea of the grad student implementation where the position of the boxplots are horizontally spread and color coded, but you run into massive problems displaying any more than two or three at a time, again running into clutter and overlap. My original idea of allowing the plot to expand horizontally would have helped that, but I have no inkling how to even begin to adjust the entire visualizations' implementation to accommodate that, if in-fact it is even possible.

I'm also curious about the vertical histogram-like implementation suggested for the second extra credit option. Specifically, what kind of math could be used to derive that sort of sound-wave like shape. If it were simply binning, you could only feasibly break it down into 3 or 4 bins, since each therapy only has 9 trials in the dataset.