

Waitlist Correspondence

This PowerPoint is a conclusion to a long running project since my junior year. It recaps what me and my lab's principal investigator set as goals since our last meeting, tests I did and improvements that I made to the program. The improvements highlight how different proteomic regions our found with higher accuracy and a checking method that uses Jaccard Similarity Index to combine incorrectly found unique regions within the proteome.

This program can be found at the github link:

<https://github.com/Srujyama/Redcarpet>

Carpet Change Catcher Formulized

Srujan Yamali

Recap

Moving away from Number of break points in favor of a adjusting penalty value based on the number of proteins

Previously did Average Protein processing time so I used that a bench mark (from PipelineCCC presentation)

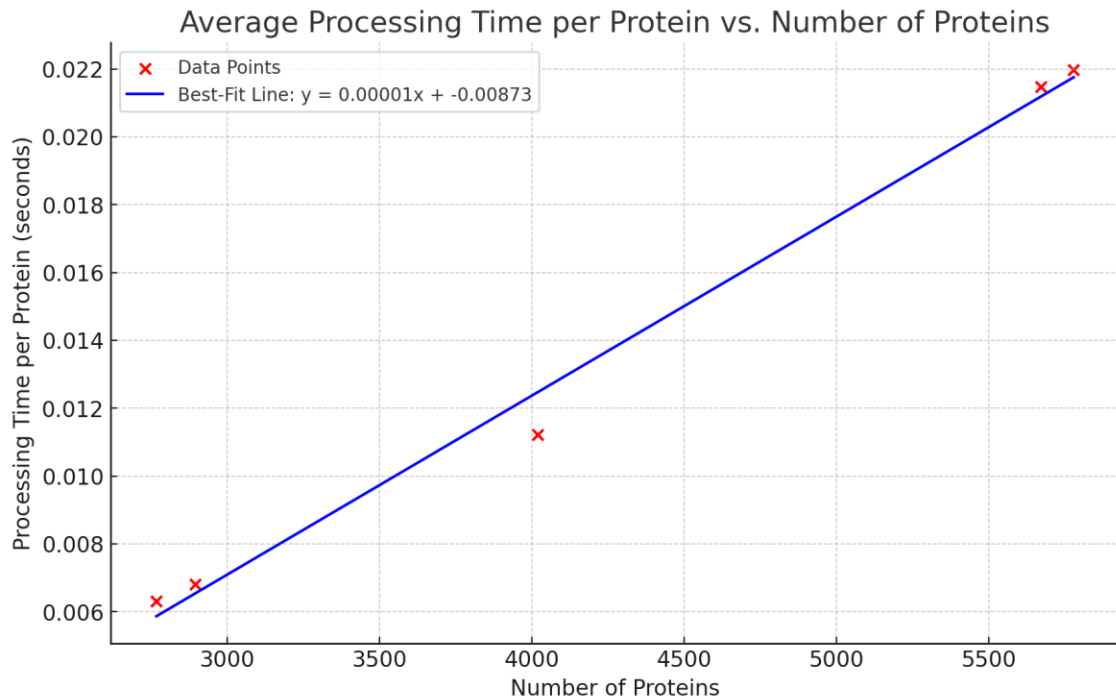
Tried to do a naturally adjust number of break points but that would always end up with it maxing out on the amount.

Pre-testing and General Info

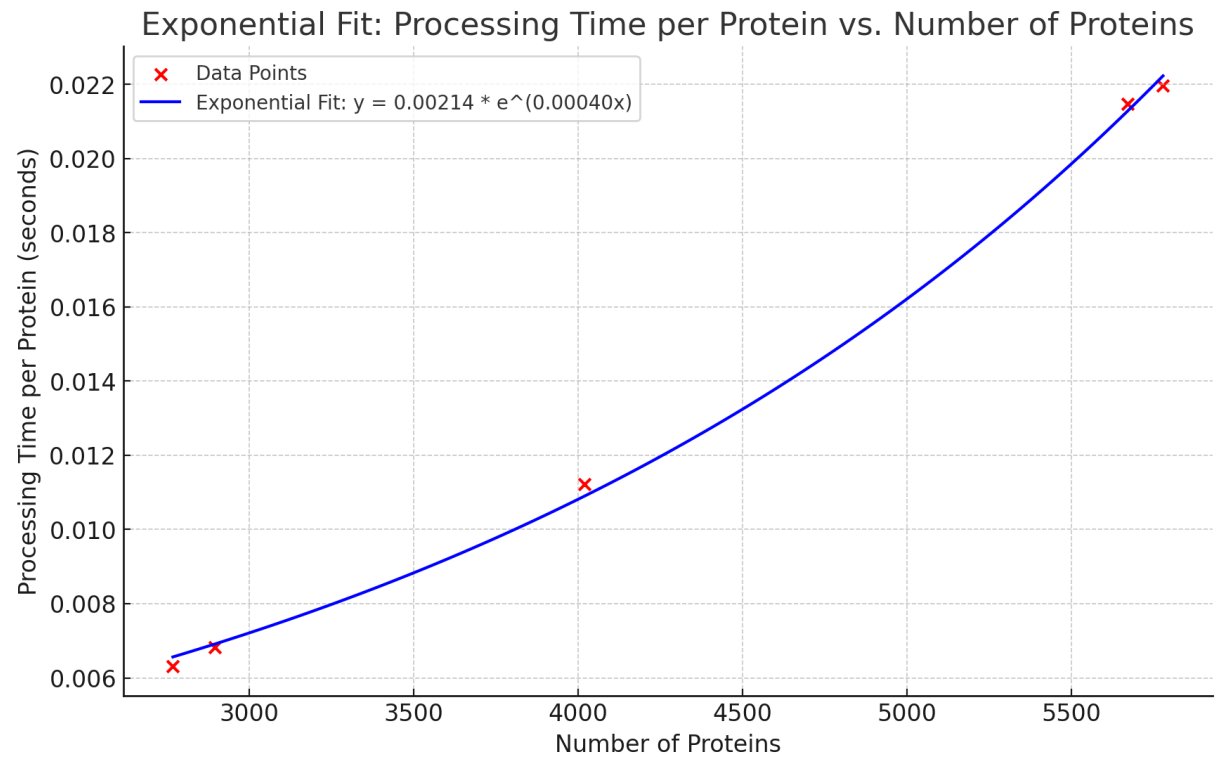
- penalty value is logarithmic in relation to amount of breakpoints
 - As Penalty value increases, decrease in amount of break points
- One standard penalty value does not work.
- This is due to the fact that there is not a linear relationship between penalty value and number of proteins produced in isolation
 - Penalty of 2000 gives 7 breakpoints given 2800 proteins
 - Penalty of 2000 gives 36 breakpoints given 4000 proteins
 - Penalty of 2000 gives 78 breakpoints given 5600 proteins
- This mimic the processing time which increases exponential.
- Dynamic penalty value needed

Average Protein processing time:

Linear regression line:
 $y = 0.00001x - 0.00873$



Exponential Function:
 $y = 0.00214 \times e^{(0.00040x)}$



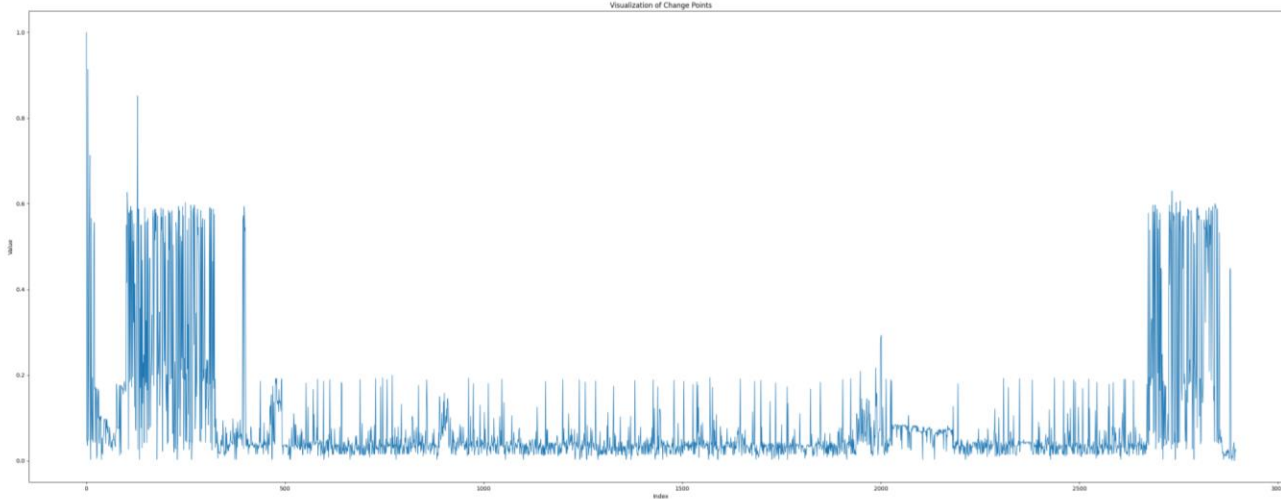
Synthetic Dataset experiment

- Tried to use synthetic dataset
- This would then provide data for a model that we could then refine
- There is one major issue
 - Making the synthetic datasets is fundamentally impossible
- This is due to the fact that these synthetic dataset is trying to emulate real world data, and the parameters that I would set to train this model would be arbitrary, which would mean that I would be training a model which would not simulate accurate data
- A linear reg model made out of faulty data

More Background

- Since we are trying to find Phages and Plasmids
- Mobile Genetic Elements were used as general benchmarks
- Previous research
 - Using 2,199 *E. coli* genomes from various sources found that nearly all isolates harbored multiple MGEs. While exact counts per genome weren't specified, the presence of multiple MGEs per isolate was common. [1]
 - Research on *P. aeruginosa* isolates showed that were common MGEs, with 98% of extensively drug-resistant strains harboring at least one gene associated with MGEs. Additionally, 47% of these isolates co-harbored three or four MGE-associated genes. [2]

Background to data



This shows how our code reveals this genomic islands and Mobile genetic elements. When the phage area is zoomed in on, we can see that is similar to the previous research within the field of how a time series would look like when a genetic change is detected.

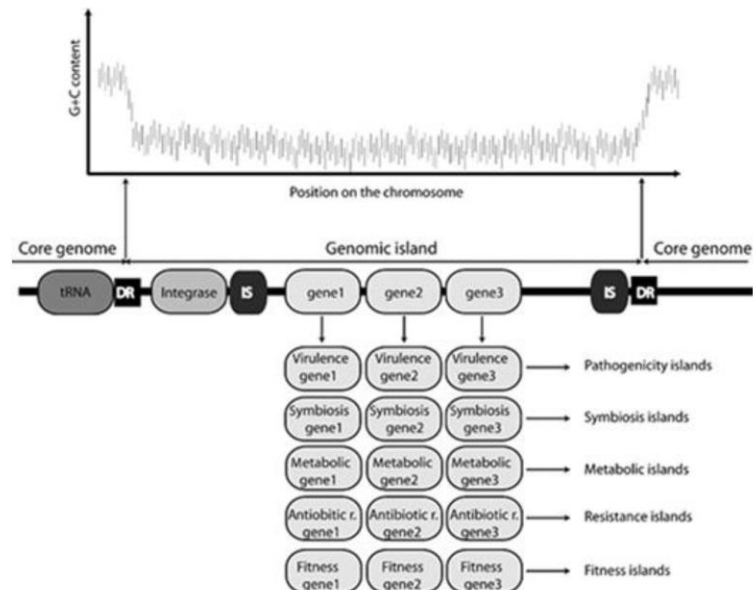
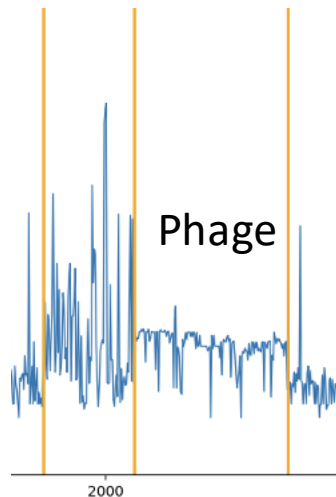


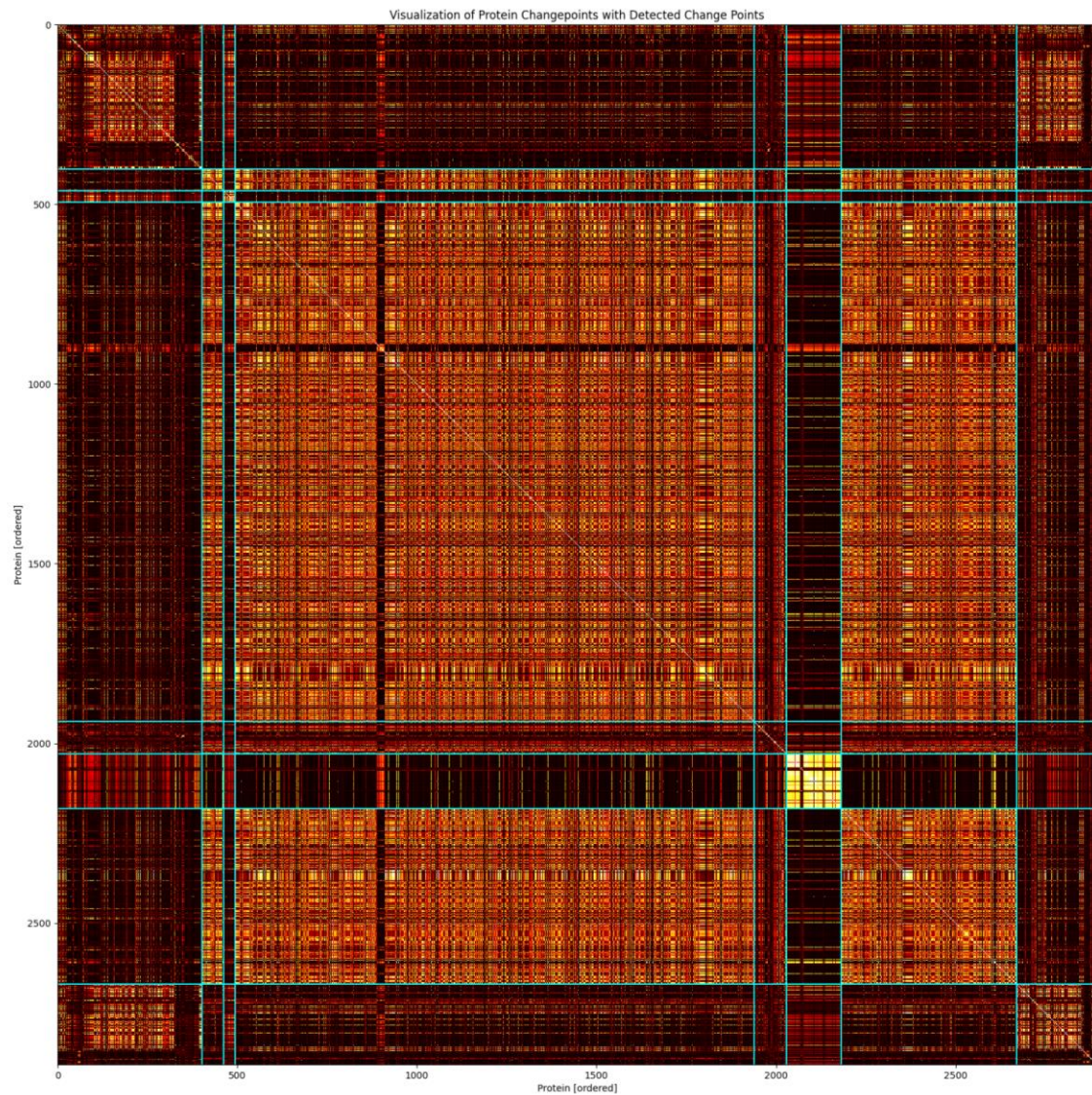
Figure
from
Source 1

Change points Detection Changes

```
def detect_change_points(data_chunk):  
    num_proteins = data_chunk.shape[0]  
    # Scale penalty: base value + a coefficient times protein count  
    penalty = (num_proteins * 1) # Supposed to be  $Y = M * X + B$  but  
ended up as  $Y = X (M * X)$   
    algo = rpt.KernelCPD(kernel="linear").fit(data_chunk)  
    return algo.predict(pen=penalty)
```

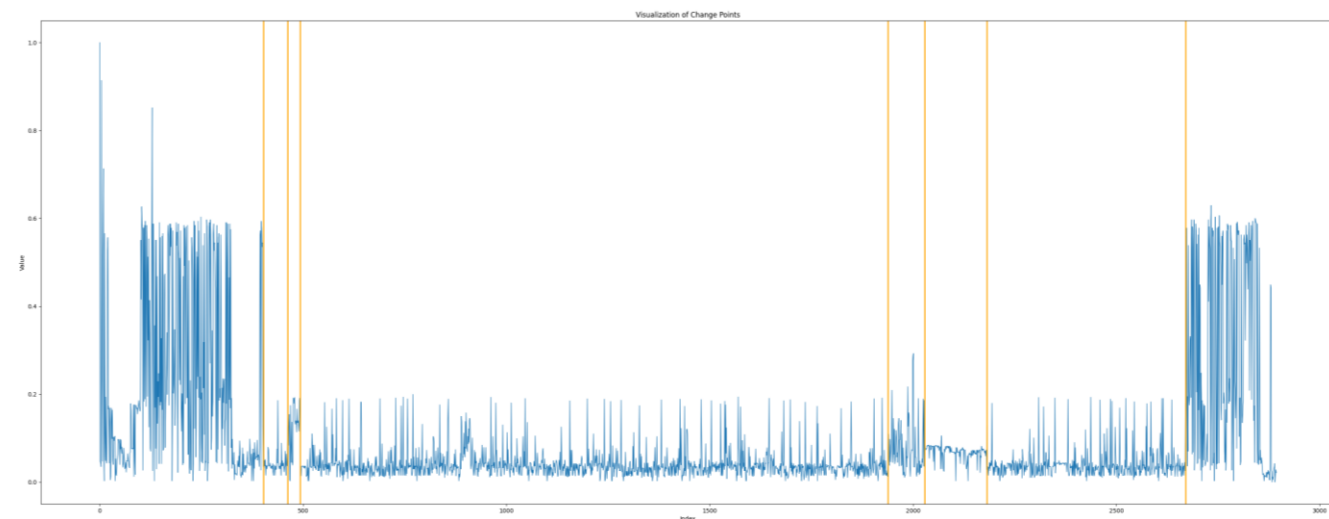
- Currently set for the amount of proteins is the penalty value
- If given (annotated?) genomes with the precise amount of break points found using wet lab a linear reg model can be created

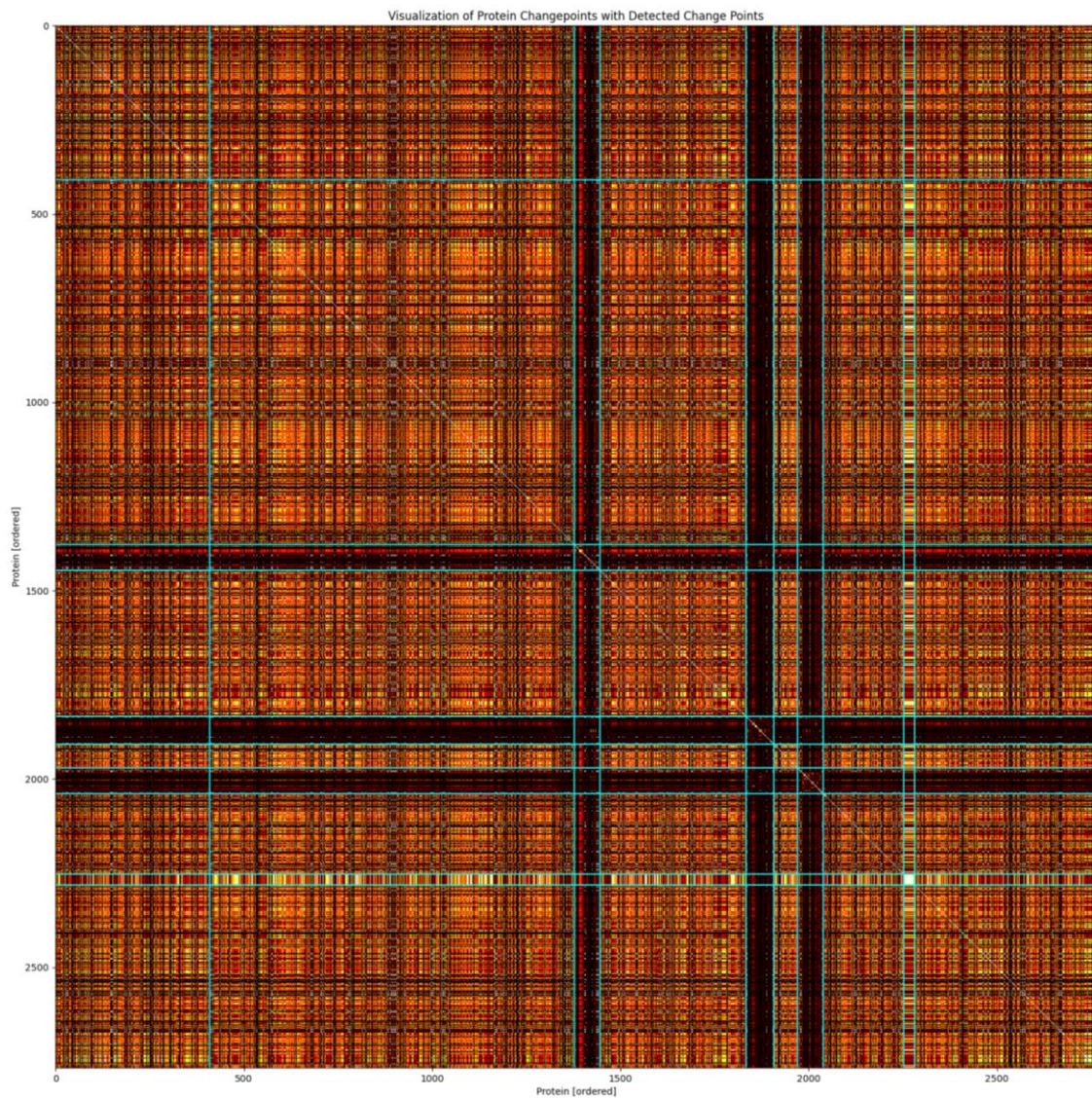
Results



S. Aureus – GCA – ST 239

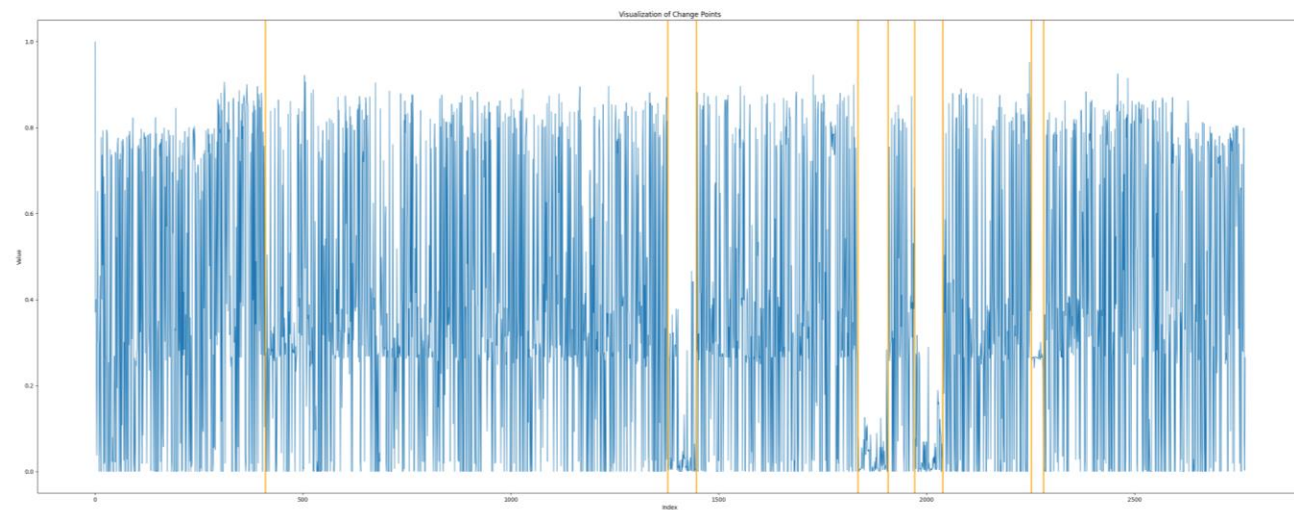
Proteins: 2894





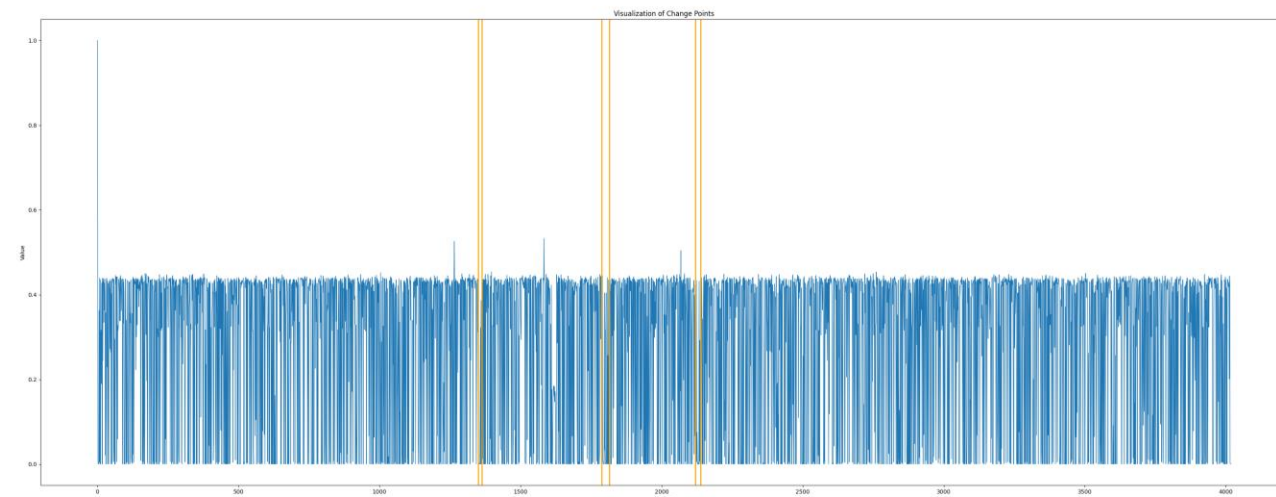
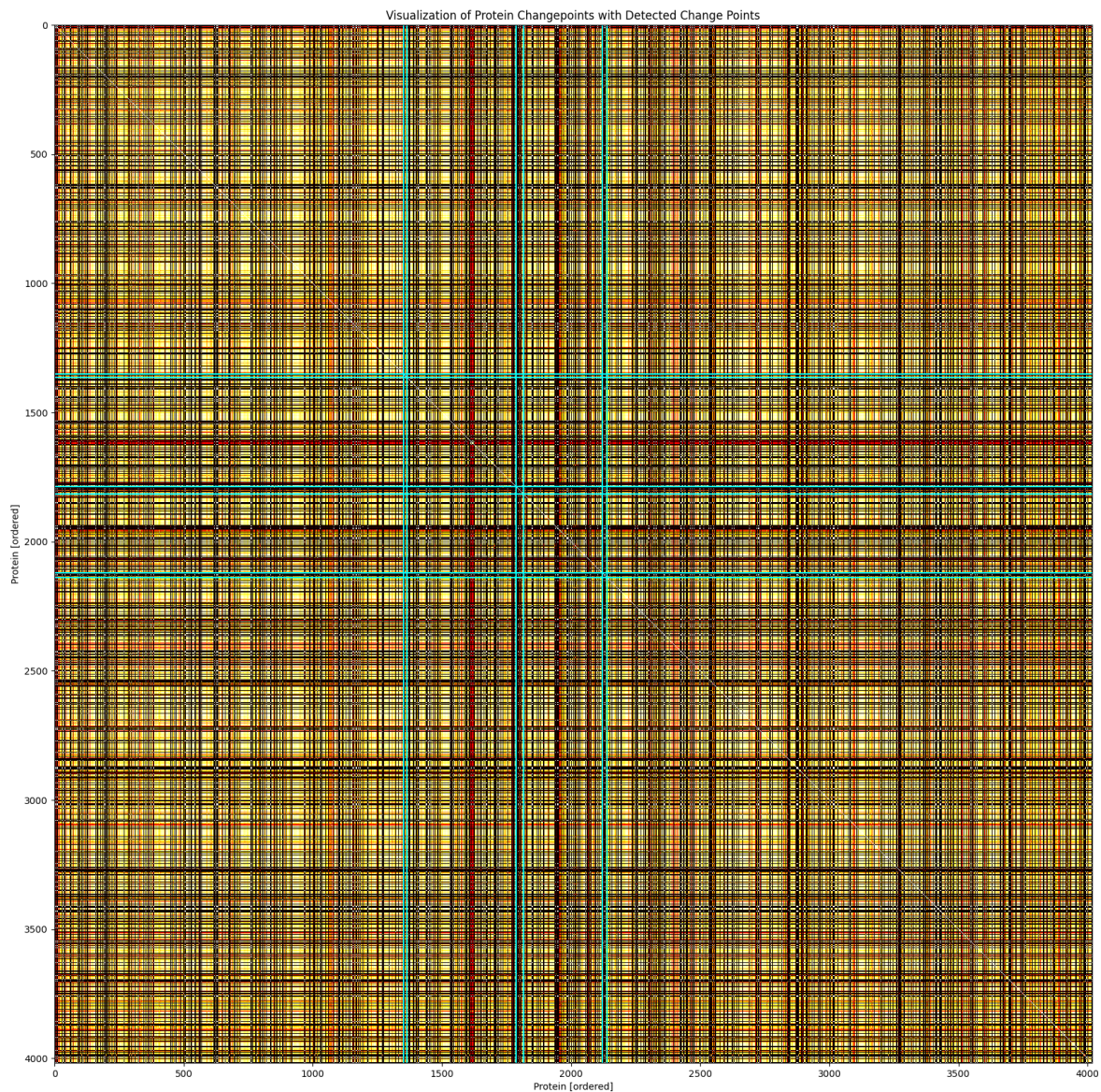
S. Aureus – GCA – ST 8

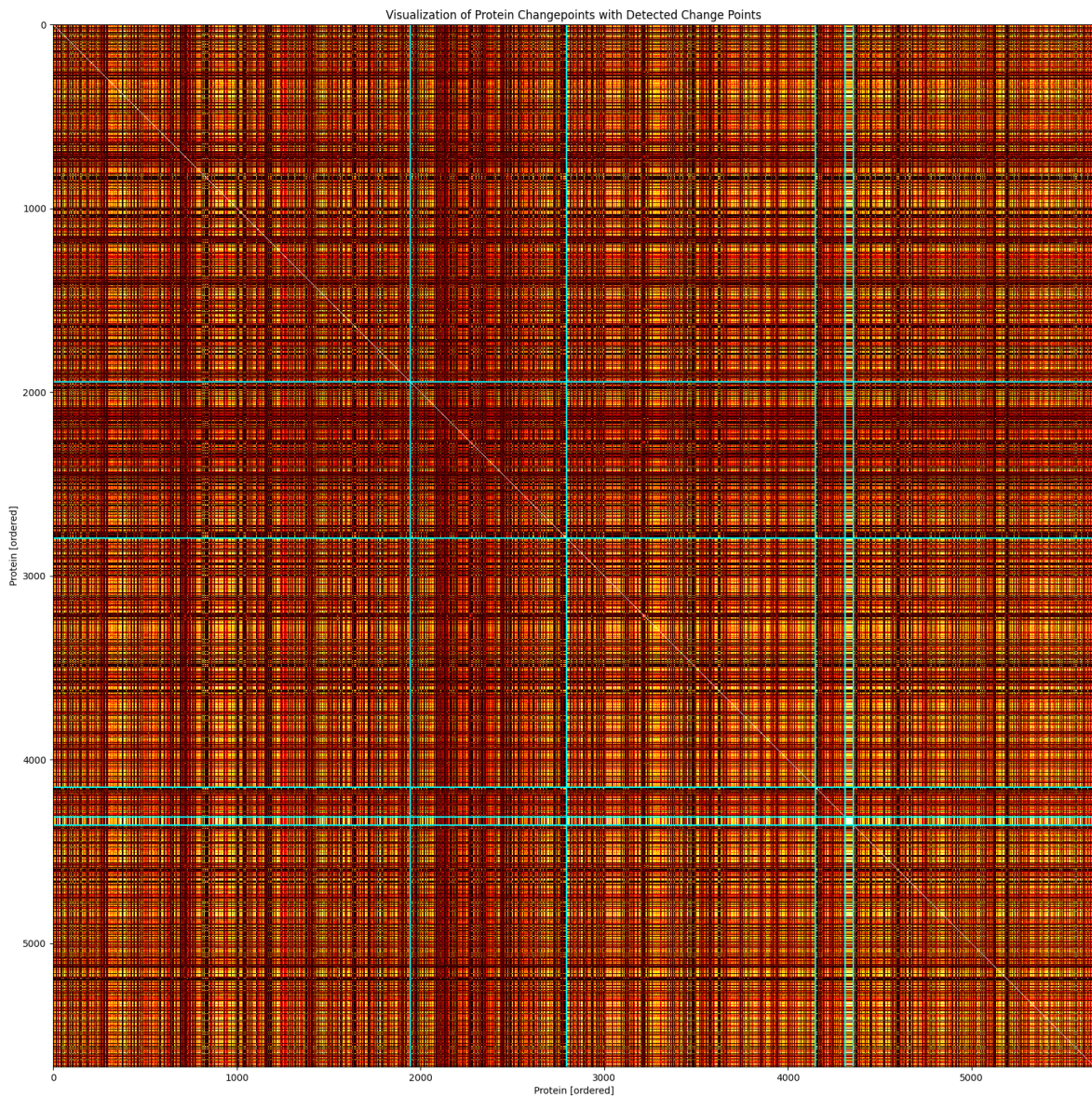
Proteins: 2767



Mycobacterium tuberculosis

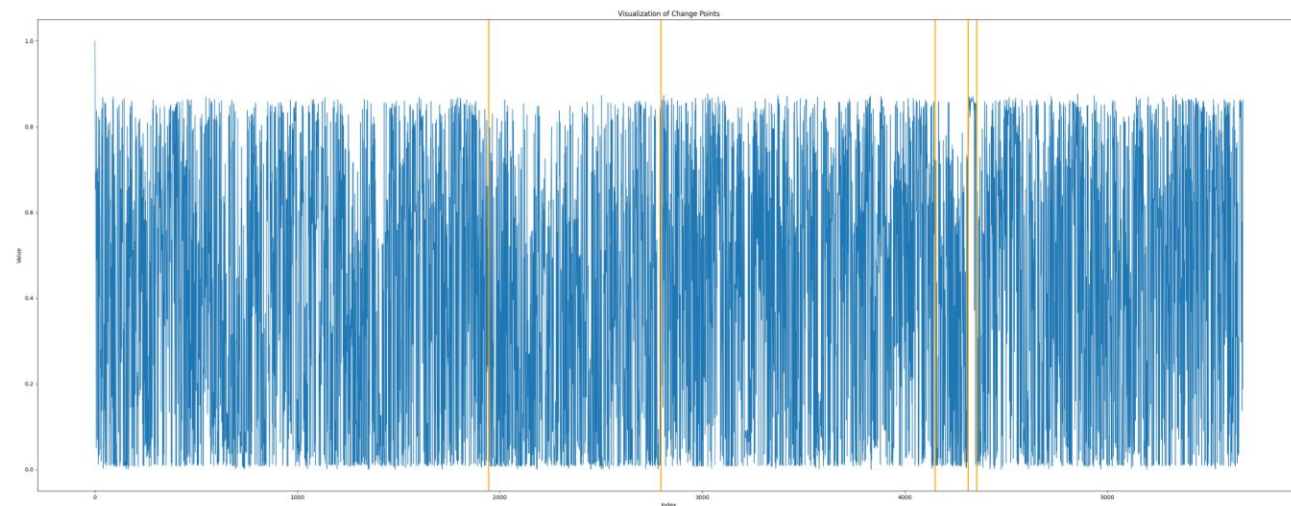
Proteins: 4018

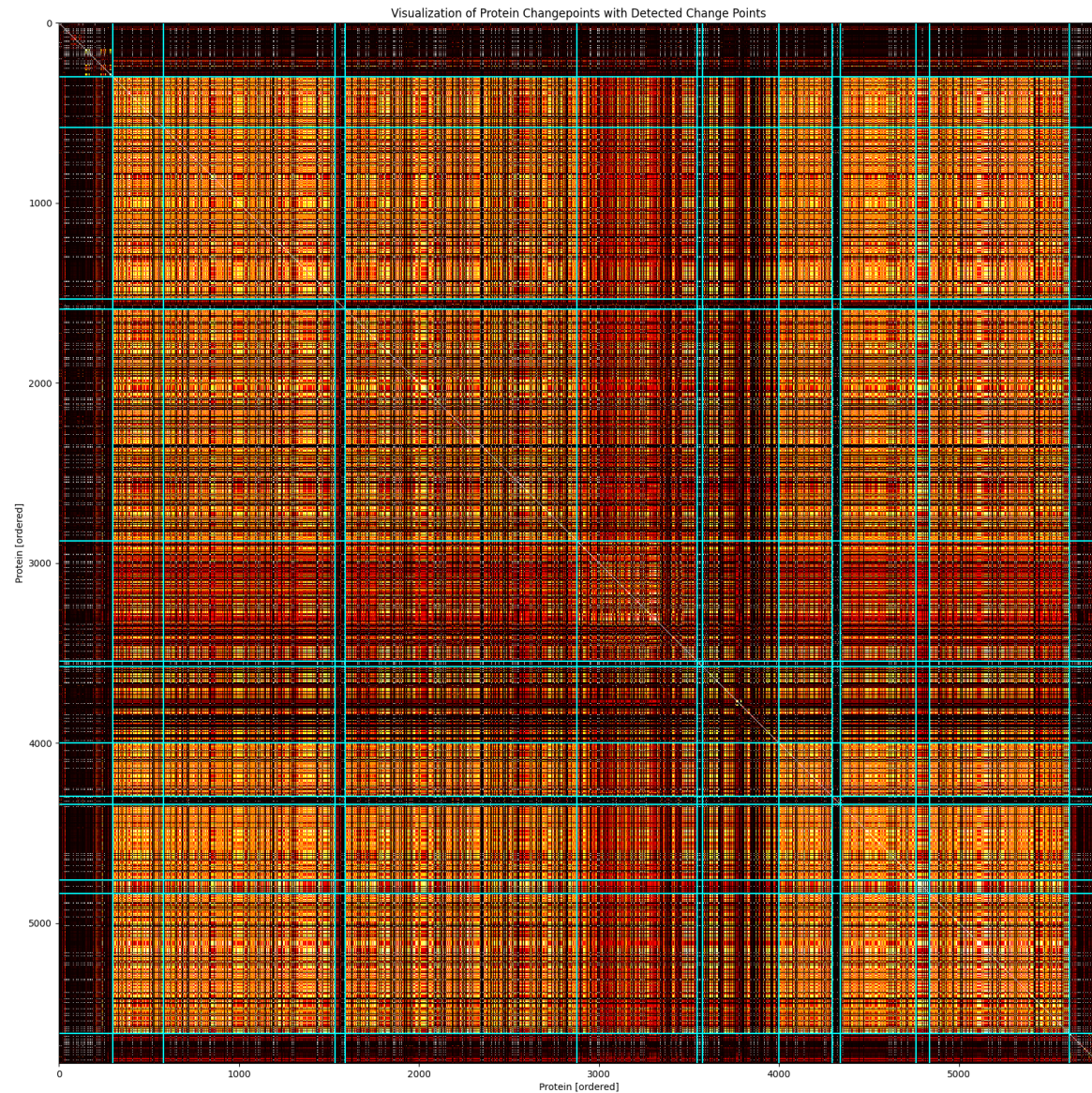




Pseudomonas aeruginosa

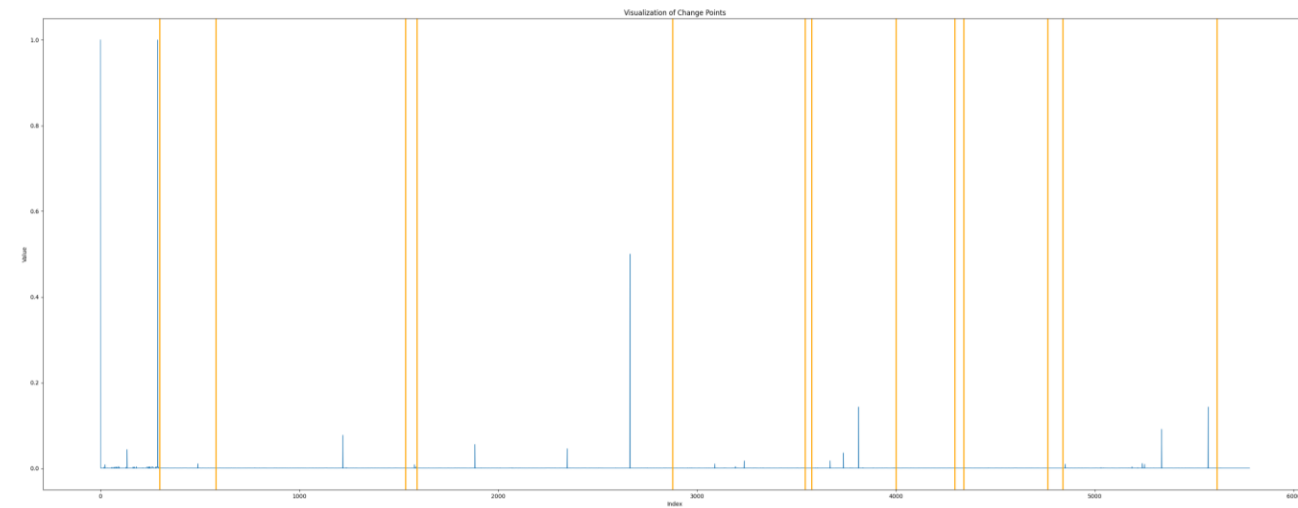
Proteins: 5671





Klebsiella pneumoniae

Proteins: 5779



Addition

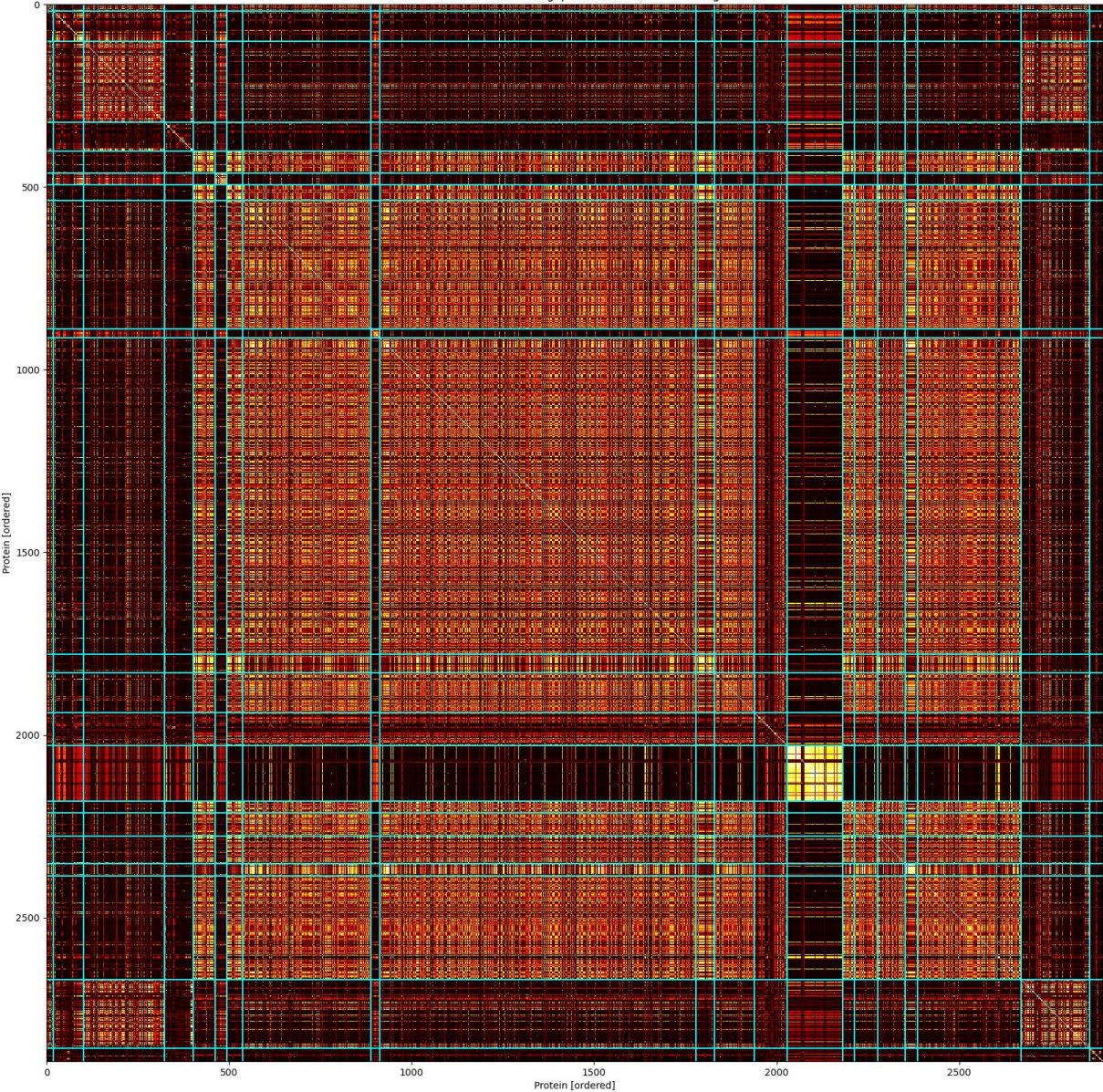
Similarity Index Checks

- Using the Jaccard correlational index information statistically significant regions are found
- If two regions next to each other are not statistically different from each other it is detected
- Those two regions are merged together
- This is significantly reduce the amount of regions found
- For example given a static penalty value we can see the effects of this safety check in action
 - From 700 to 450 (This was penalty of 100)
 - From 21 to 13 (This was penalty of 1000)

Pre Similarity Index Checks

Penalty of 100

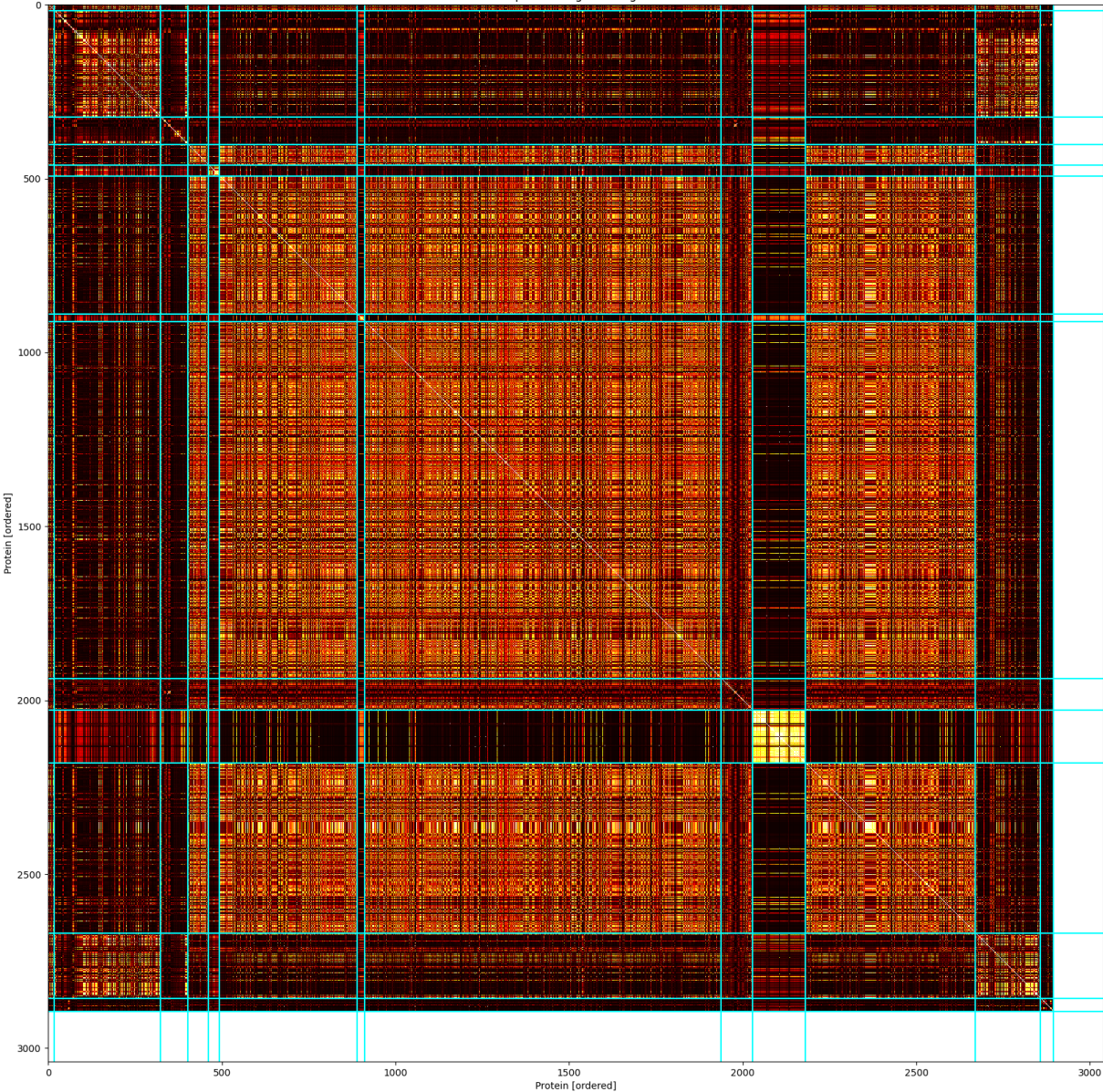
Visualization of Protein Changepoints with Detected Change Points



Post Similarity Index Checks

Penalty of 1000

Heatmap with Merged Change Points



Similarity Index Checks Merging

```
def merge_adjacent_similar_regions(data, regions, similarity_threshold=0.05):
    merged_regions = []
    i = 0
    while i < len(regions):
        start1, end1, data1 = regions[i]

        while i + 1 < len(regions):
            start2, end2, data2 = regions[i + 1] # Get the next region

            if end1 + 1 == start2: # Check if regions are adjacent
                region1_data, region2_data = pad_to_same_length(data1, data2)
                t_stats, p_values = ttest_ind(region1_data.T, region2_data.T, axis=1, nan_policy='omit')
                avg_p_value = np.nanmean(p_values) if p_values.size > 0 else float('inf')

                if avg_p_value > similarity_threshold:
                    print(f"Merging regions ({start1}, {end1}) and ({start2}, {end2}), p-value: {avg_p_value:.4f}")
                    end1 = end2
                    i += 1
                    continue
            break

        merged_regions.append((start1, end1))
        i += 1

    return merged_regions
```

How it looks like in terminal

- Merging regions (19, 101) and (102, 323), p-value: 0.0619
- Merging regions (494, 537) and (538, 889), p-value: 0.1713
- Merging regions (913, 1780) and (1781, 1829), p-value: 0.1262
- Merging regions (913, 1829) and (1830, 1938), p-value: 0.2136
- Merging regions (2182, 2214) and (2215, 2277), p-value: 0.3182
- Merging regions (2182, 2277) and (2278, 2352), p-value: 0.1979
- Merging regions (2182, 2352) and (2353, 2386), p-value: 0.3737
- Merging regions (2182, 2386) and (2387, 2670), p-value: 0.2220

Acknowledgments

1. Juhas M, van der Meer JR, Gaillard M, Harding RM, Hood DW, Crook DW. Genomic islands: tools of bacterial horizontal gene transfer and evolution. FEMS Microbiology Reviews. 2009;33:376–93.
2. Rana C, Vikas V, Awasthi S, Gautam D, Vats A, Rajput S, et al. Antimicrobial resistance genes and associated mobile genetic elements in Escherichia coli from human, animal and environment. Chemosphere [Internet]. 2024;369:143808. Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0045653524027097>

Moustafa Lab

Ahmed Moustafa
Erin Theiller

Lab Email

Yamalis@chop.edu

Planet Lab

Arnav Lal

