Commanalities in Network Structure of Viral Respiratory Diseases and COVID19

Shruti Kaushal
Data Science Institute
Columbia University
New York, NY 10027
sk4963@columbia.edu

Abstract

With death toll in millions worldwide, Coronavirus is considered one of the most deadliest viruses to exist. The symptoms are similar to other viral respiratory disorders but what makes COVID19 deadly is still being extensively researched. In this paper, I try to answer this question by contrasting causal gene regulatory networks for three viral respiratory disorders - common cold (caused by Rhinovirus), pneumonia and influenza with those of critical and non-critical COVID19 patients. I also propose possible gene targets for designing drug therapies for COVID19 by calculating the Markov blanket of targets of drugs currently in clinical trials or being prescribed.

1 Introduction

Early on in the pandemic, doctors were prescribing drugs used to treat auto-immune diseases for COVID19. This was because research had shown that coronavirus remained undetected in the human body for long. Upon detection, the immune system produced a cytokine storm to in an attempt to overcompensate for the delayed response. This storm is actually counterproductive and often leads to an increase in disease severity. The severity is characterized by symptoms and symptoms of COVID19 are similar to the symptoms of any other viral respiratory infection, like cold, cough, headache to name a few.

In this paper, I tried to characterize the differences and commonalities between COVID19 and other respiratory disorders caused by viruses. I did so in the space of gene regulatory networks with the focus on network structure. I used publicly available datasets to learn causal networks for each condition and compared their structures. In doing so, I was able to identify common sub-networks that are conserved across all respiratory condition and their corresponding phenotype.

2 Data

I used two publicly available NCBI datasets GSE157240 (viral respiratory diseases dataset) and GSE172114 (COVID19 dataset) that contain RNAseq data of peripheral blood samples from healthy and healthy controls and respiratory virus-infected patients. The COVID19 dataset contains 69 samples in total corresponding to 46 critical and 23 non-critical patients at hospitalization. This dataset doesn't contain any controls.

GSE157240 contains 162 samples of patients/individuals from US an Sri Lanka with 5 different types of infections including common cold, pneumonia, influenza, dengue, adenovirus and cytomegalovirus. Unlike the COVID19 dataset, this contained 20 healthy controls.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Both datasets are normalized using a different algorithm and so could not be combined for the purposes of learning a network. The raw counts were normalized using edgeR and DESeq2 and log transformed. Since the data was in normal form no transformations were done before performing any kind of analysis.

3 Methods

The datasets contain count values for more than 15,000 genes. Learning a gene regulatory network with more than 15,000 nodes, given limited computational capacity can take exponential time. Hence, I extracted statistically significant genes for each condition by comparing with healthy controls and performing differential gene expression analysis. Student t test was used to for hypothesis tests and p-values were adjusted using FDR correction with $\alpha = 0.01$. For each condition or respiratory disease, a list of statistically significant genes was generated. Finally, genes that belonged to the lists of all conditions were selected for further analysis. Since there are no controls in the COVID19 study, differential gene expression analysis was not performed and the intersection of all genes with non-zero count values was taken with all the other lists of statistically significant genes. The number of genes greatly reduced from 15,000 to 99 genes. The normalized count values of these genes were then used to learn directed graphs. Essentially, the networks differed only in terms of edges and had the same nodes.

For each condition, a network was learnt using the samples/individuals belonging to that condition group. Three different structure learning algorithms were used, two constraint based and one greedy score based - PC, Grow-Shrink algorithm and GES algorithm with BIC score as the metric. The critical value for PC was set to 0.05 to avoid generating extremely sparse graphs given the small number of nodes. Fischer's Z test was used for hypothesis testing. Other non-parametric options for hypothesis testing could be explored, but due to lack of time I decided to go with the default settings for each algorithm.

Grow Shrink algorithm was implemented using the causal-learn package that wrapped over the functions in bnlearn R package. It is a constraint-based algorithm for learning Bayesian networks by estimating Markov blankets of each node that is added and deleted. Similar to forward and backward equivalence search in GES, Grow-Shrink algorithm has two phases - "growing" and "shrinking". In the growing phase, it added nodes to the graph or markov blanket based on conditional independence test followed by the shrinking phase in which it removes nodes based on similar criteria.

The purpose of learning networks using different approaches was to try and quantify the sensitivity the data to these approaches. In fact, PC and Grow-Shrink algorithm learnt much sparser graphs compared to GES with an average number of edges approximately equal to 100. While most networks learned using GES had more than a thousand edges.

All the analysis was performed in python using the causal-learn package.

4 Results

This section is divided into three sections. The first section talks about quantifying the differences between networks of different conditions. It is focused on generating and visualizing metrics. The second section, on the other hand, focuses on translating the differences observed in the first section to what is observed in biology, for example, analyzing the GO terms of genes that form an important subnetwork or a connected component.

4.1 Structural Differences

All the analysis performed below was done using the networks learned by the PC algorithm. The decision of using these graphs was made after careful consideration of what the data represents and what would be the most meaningful. Grow-Shrink algorithm came out in the year 2003 but due to my limited knowledge about the same I decided to not pursue it for this project.

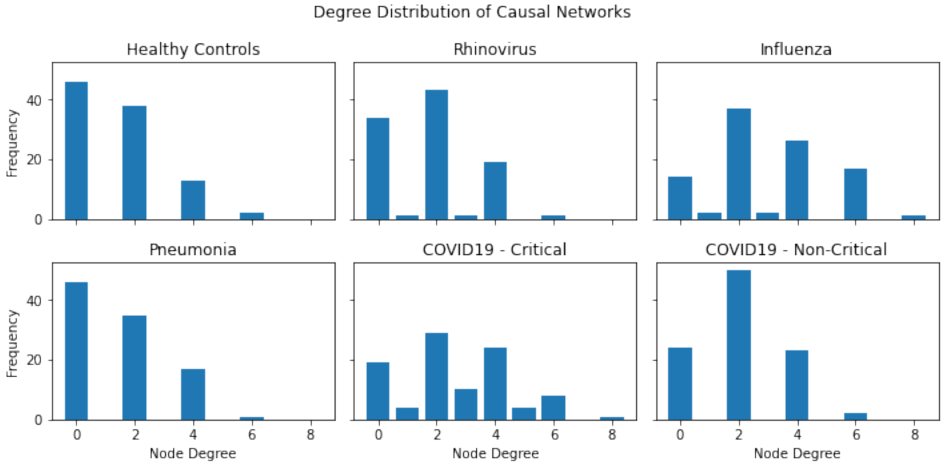

Figure 1: Degree distribution of all networks

4.1.1 Degree Distribution & Structural Distance

One of the first things that I looked at was the degree distribution of each network (see figure 1). Moving forward the reader must keep in mind that each network corresponds to a condition and so there are 6 networks for controls, pneumonia, influenza, common cold (rhinovirus), critical and non-critical COVID19 to be compared.

It was surprising to see that the causal network of pneumonia was as sparse as that of healthy controls. A close second was the network for common-cold. Networks corresponding to influenza and critical COVID19 were relatively the most dense. This is clearly not enough to quantify differences between disease conditions and controls as it does not take into account the actual edges and what the nodes represent.

I then looked at the structural hamming distance (see figure 2) and its variation called the structural intervention distance. Explanation of structural intervention distance is beyond the scope of this project but it can be thought of as a "better" version of structural hamming distance. The distance between healthy controls and each condition was calculated and plotted as a scatter plot. A value closer to zero would imply that the two networks are structurally similar. Every edge that is missing or not present in the network of healthy controls is counted as a mistake by this metric. For directed graphs, opposite direction is also counted as a mistake. It can be seen from the figure that networks for dengue, common cold and pneumonia are structurally the closest to healthy controls. The results for pneumonia seem counter-intuitive and could be due to the small size of the network and subsequent loss of information. It was interesting to see that influenza ranks higher than critical COVID19. The trend between critical and non-critical COVID19 was expected.

Unlike structural hamming distance, structural intervention distance put all the networks from the healthy controls at the same distance, equal to 136. It is probably because it averages certain quantities out and so is not informative enough for the purposes of this analysis.

4.1.2 Conserved Sub-networks

Having looked at the structural distances and sparsity of graphs, I moved on to checking if there was a conserved sub-network or substructure that was present in all the causal networks learned. It is hard to give a definite answer before performing the analysis as one would expect some similarities gives that all conditions fall under the same umbrella of diseases but also each network was learnt using a very small number of genes. It could be that the loss of information led to filtering out genes that are in fact important in same way for characterizing one disease but not another. Extracting sub-networks common to the healthy controls and a given disease could shed light on how severe the disease is.

Due to lack of time I was unable to generate visualizations but all conditions, on average, had less than a 6-node sub-network common with that of healthy controls. Hence, this analysis did not prove to be informative

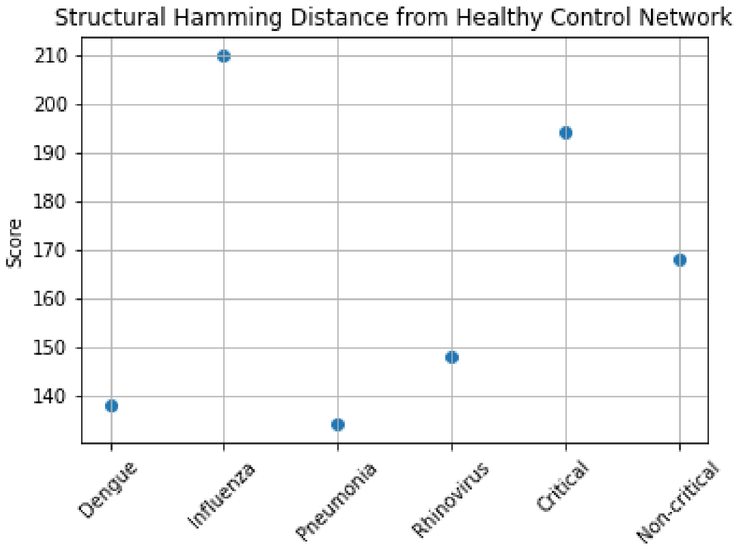

Figure 2: Scatter plot of hamming distance of all disease conditions to healthy controls

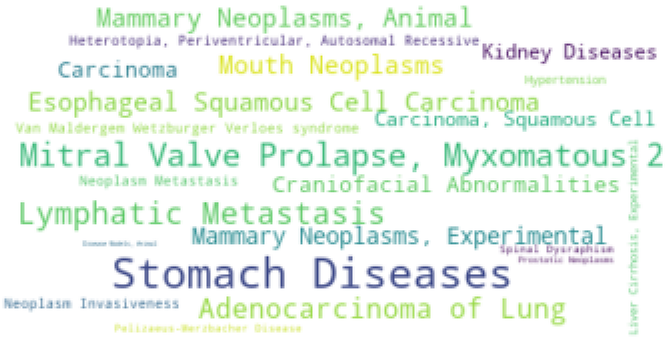

Figure 3: Degree distribution of all networks

4.2 Phenotoype Analysis

In an attempt to actually see if the causal networks learnt actually were representative of the interactions given a certain condition, strongly connected components were calculated for each graph and subsequently enriched with Gene Ontology or GO terms. Information regarding GO terms was downloaded from the publicly available database CTD. As it turns out, MAPK signaling pathway is present for connected components of both levels of severity of COVID19 and for common sub-networks in influenza, pneumonia and COVID19. Research has shown that coronavirus upregulates MAPK and that its inhibition can be a promising therapeutic for curing COVID19. Moreover, GO terms related to the immune system including 'Interleukin-4 and 13 signaling', both cytokines and known for creating a cytokine storm, repeatedly appear for all permutations and combinations involving COVID19 networks.

The word-cloud of the diseases associated or inferred from the genes in the strongly connected component of the network corresponding to critical COVID19 is further evidence validating the approach and that these networks can be leveraged to design drug therapies

4.3 Possible Drug Targets for COVID19

This analysis does not align with the premise of this project but it is equally important. Having established that the causal networks are in fact informative and are able to capture difference between conditions, I wanted to see if they could be leveraged to design possible drug therapies for COVID19 or any other disease if the same process is followed.

I exploited graph theory when performing this analysis. I used the concept of Markov blankets - set of all nodes that make the target node conditionally independent of the rest. When looking at a huge network, it is easy to get lost in the global nature/behavior which is bound to complicate things given the size. Markov blanket not only takes care of the size but also the dependency relationships. To identify/predict additional drug targets, I extracted the gene targets of drugs currently in clinical trials or being prescribed for COVID19 from publicly available database DrugBank. The list of the drugs I investigated includes - remdesivir, favipiravir, adalimumab, dihydro-artemisinin piperaquine, leflunomide, dipyridamole, chloroquine, hydroxychloroquine, suramin sodium, lopinavir, ritonavir, arbidol, umifenovir, IFN-alpha 2b, and dexamethasone. Of all the gene targets that these drugs have, only one was present in the network for critical and non-critical COVID19, a gene named 'ANXA1'. This gene encodes a protein that has anti-inflammatory activities. Moreover, the loss of function of this gene has been detected in multiple tumors. Clearly, it seems to be involved in the immune system in some way. The Markov blanket of this gene includes three other genes - 'MMADHC', 'PCMT1', 'VPS29'. Enriching these for GO terms, I found two interesting results -

1 These genes are involved in GPCR signaling pathways that SARS-CoV-2 is known to hijack to dysregulate lung ion and fluid transport leading to severe damage and fluid buildup in the lungs

2 They are also involved in the cytokine signaling pathway and could potentially solve the problem of a cytokine storm that the immune system creates in order to overcompensate for the delayed response

The motivation behind this process is that given that these drugs works and have a definite target, the genes in the markov blankets of these targets could theoretically also have the same effect. Hence the genes mentioned above could be investigated as potential drug targets. The enrichment analysis proves their involvement in pathways that are directly affected by SARS-CoV-2.

5 Discussion

There is a lot of scope of improvement. Some of the areas include -

1 More work can be done in FDR correction of differential gene expression analysis

2 More work on figuring out which algorithm will work best for genetics dataset – score based seem like a better option since they do not perform any hypothesis tests but do have a sparsity parameter for which a grid search can be done

3 Could also incorporate associations in terms of correlation when extracting genes for structure learning

4 Could perform princpal component analysis on the data for each condition and the learn the graph using the genes/features in each principal component